



Computer Science and Information Systems

Published by ComSIS Consortium

Volume 7, Number 4
December 2010

ComSIS Consortium:

University of Belgrade:

Faculty of Organizational Science, Belgrade, Serbia
Faculty of Mathematics, Belgrade, Serbia
School of Electrical Engineering, Belgrade, Serbia

Serbian Academy of Science and Art:

Mathematical Institute, Belgrade, Serbia

Union University:

School of Computing, Belgrade, Serbia

University of Novi Sad:

Faculty of Sciences, Novi Sad, Serbia
Faculty of Technical Sciences, Novi Sad, Serbia
Faculty of Economics, Subotica, Serbia
Technical Faculty "M. Pupin", Zrenjanin, Serbia

University of Montenegro:

Faculty of Economics, Podgorica, Montenegro
Faculty of Mathematics and Science, Podgorica, Montenegro

EDITORIAL BOARD:

Editor-in-Chief: Mirjana Ivanović, University of Novi Sad

Vice Editor-in-Chief: Ivan Luković, University of Novi Sad

Managing editor: Zoran Putnik, University of Novi Sad

Managing editor: Miloš Radovanović, University of Novi Sad

Editorial Assistant: Vladimir Kurbalija, University of Novi Sad

Editorial Assistant: Jovana Vidaković, University of Novi Sad

Editorial Assistant: Ivan Pribela, University of Novi Sad

Editorial Assistant: Slavica Aleksić, University of Novi Sad

Editorial Assistant: Srđan Škrbić, University of Novi Sad

Associate editors:

P. Andrae, Victoria University, New Zealand

D. Bojić, University of Belgrade, Serbia

D. Bečejski-Vujaklija, University of Belgrade, Serbia

I. Berković, University of Novi Sad, Serbia

G. Bhutkar, Vishwakarma Institute of Technology, India

L. Böszörményi, University of Clagenfurt, Austria

K. Bothe, Humboldt University of Berlin, Germany

Z. Budimac, University of Novi Sad, Serbia

G. Devedžić, University of Kragujevac, Serbia

V. Devedžić, University of Belgrade, Serbia

J. Đurković, University of Novi Sad, Serbia

S. Guttormsen Schar, ETH Zentrum, Switzerland

P. Hansen, University of Montreal, Canada

L. C. Jain, University of South Australia, Australia

V. Jovanović, Georgia Southern University, USA

Lj. Kaščelan, University of Montenegro, Montenegro

P. Mahanti, University of New Brunswick, Canada

M. Maleković, University of Zagreb, Croatia

N. Mitić, University of Belgrade, Serbia

A. Mitrović, University of Canterbury, New Zealand

N. Mladenović, Serbian Academy of Science, Serbia

P. Mogin, Victoria University, New Zealand

S. Mrdalj, Eastern Michigan University, USA

G. Nenadić, University of Manchester, UK

Z. Ognjanović, Serbian Academy of Science, Serbia

A. Pakstas, London Metropolitan University, England

P. Pardalos, University of Florida, USA

M. Racković, University of Novi Sad, Serbia

B. Radulović, University of Novi Sad, Serbia

M. Stanković, University of Niš, Serbia

D. Tošić, University of Belgrade, Serbia

J. Trninić, University of Novi Sad, Serbia

J. Woodcock, University of Kent, England

EDITORIAL COUNCIL:

S. Ambroszkiewicz, Polish Academy of Science, Poland

Z. Arsovski, University of Kragujevac, Serbia

D. Banković, University of Kragujevac, Serbia

T. Bell, University of Canterbury, New Zealand

S. Bošnjak, University of Novi Sad, Serbia

Ž. Branović, University of Novi Sad, Serbia

H. D. Burkhard, Humboldt University of Berlin, Germany

B. Chandrasekaran, Ohio State University, USA

V. Ćirić, University of Belgrade, Serbia

D. Domazet, Faculty of Information Technology,
Belgrade, Serbia

S. Đrođević-Kajan, University of Niš, Serbia

P. Hotomski, University of Novi Sad, Serbia

P. Janičić, University of Belgrade, Serbia

Z. Jovanović, University of Belgrade, Serbia

L. Kalinichenko, Russian Academy of Science, Russia

Z. Konjović, University of Novi Sad, Serbia

I. Koskosas, University of Western Macedonia, Greece

W. Lamersdorf, University of Hamburg, Germany

T. C. Lethbridge, University of Ottawa, Canada

A. Lojpur, University of Montenegro, Montenegro

Y. Manolopoulos, Aristotle University, Greece

A. Mishra, Atılım University, Turkey

S. Mishra, Atılım University, Turkey

J. Protić, University of Belgrade, Serbia

D. Simpson, University of Brighton, England

D. Starčević, University of Belgrade, Serbia

S. Stojanov, Faculty for Informatics, Plovdiv, Bulgaria

D. Surla, University of Novi Sad, Serbia

M. Tuba, University of Belgrade, Serbia

P. Tumbas, University of Novi Sad, Serbia

P. Zarate, IRIT-INPT, Toulouse, France

K. Zdravkova, University of Ss. Cyril and Methodius,

Skopje, FYROM

ComSIS Editorial office:

**University of Novi Sad, Faculty of Sciences,
Department of Mathematics and Informatics**

Trg Dositeja Obradovica 4, 21000 Novi Sad, Serbia

Phone: +381 21 458 888; **Fax:** +381 21 6350 458

www.comsis.org; Email: comsis@uns.ac.rs

**Volume 7, Number 4, 2010
Novi Sad**

Computer Science and Information Systems

ISSN: 1820-0214

ComSIS Journal is sponsored by:

Ministry of Science and Technological Development of Republic of Serbia -
<http://www.nauka.gov.rs/lat>



Computer Science and Information Systems

AIMS AND SCOPE

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

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

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

Indexing Information

ComSIS is covered or selected for coverage in the following:

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

Information for Contributors

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

Criteria for Acceptance

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

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

Copyright and Use Agreement

All authors are requested to sign the "Transfer of Copyright" agreement before the paper may be published. The copyright transfer covers the exclusive rights to reproduce and distribute the paper, including reprints, photographic reproductions, microform, electronic form, or any other reproductions of similar nature and translations. Authors are responsible for obtaining from the copyright holder permission to reproduce the paper or any part of it, for which copyright exists.

Computer Science and Information Systems

Volume 7, Number 4, December 2010

CONTENTS

Editorial

Disclosure of Plagiarism

Invited Paper

- 679** **Advanced Indexing Technique for Temporal Data**
Bela Stantic, Rodney Topor, Justin Terry, Abdul Sattar

Papers

- 705** **COLIBROS: Educational Operating System**
Žarko Živanov, Predrag Rakić, Miroslav Hajduković
- 721** **An Approach to Project Planning Employing Software and Systems Engineering Meta-Model Represented by an Ontology**
Miroslav Liška, Pavol Návrat
- 737** **Facilitating Information System Development with Panoramic View on Data**
Dejan Lavbič, Iztok Lajovic, Marjan Krisper
- 769** **Metrics for Evaluation of Metaprogram Complexity**
Robertas Damaševičius, Vytautas Stukys
- 789** **An Evolutionary Solution for Multimodal Shortest Path Problem in Metropolises**
Rahim A. Abbaspour, Farhad Samadzadegan
- 813** **Trojan horses in mobile devices**
Daniel Fuentes, Juan A. Álvarez, Juan A. Ortega, Luis Gonzalez-Abril, Francisco Velasco
- 823** **A Tabular Steganography Scheme for Graphical Password Authentication**
Tsung-Hung Lin, Cheng-Chi Lee, Chwei-Shyong Tsai, Shin-Dong Guo
- 843** **A Multi-attribute Auction Model by Dominance-based Rough Sets Approach**
Rong Zhang, Bin Liu, Sifeng Liu
- 859** **Design of Median-type Filters with an Impulse Noise Detector Using Decision Tree and Particle Swarm Optimization for Image Restoration**
Bae-Muu Chang, Hung-Hsu Tsai, Xuan-Ping Lin, Pao-Ta Yu

Papers selected from "The 4th International Conference on Information Technology", June 3 – 5, Amman, Jordan 2009

- 883 Personalising the Dynamic Information Resources in a Self Organized Web System Using CAS and MAS**
Rattrout Amjad, Assaf Rasha, Al-Dahoud Ali
- 907 A Hybrid Variable Neighborhood Search Algorithm for Solving Multi-Objective Flexible Job Shop Problems**
Jun-qing Li, Quan-ke Pan, Sheng-xian Xie
- 931 Automation of the Moving Objects Movement Prediction Process Independent of the Application Area**
Ivana Nižetić, Krešimir Fertalj

EDITORIAL

As the year draws to a close, reflecting on the past several years and the road our journal has taken brings into focus a very positive outlook for the future. The academic interest in ComSIS has steadily increased, an observation supported by the large number of higher-quality articles being submitted, widening coverage of the journal by leading abstracting and indexing databases, and the rising number of citations to articles published in ComSIS. Based on the last observation, we are optimistic that ComSIS will receive a solid two-year impact factor for 2010 from Thomson Reuters, placing the journal firmly into the Science Citation Index.

This issue of ComSIS contains one invited paper, nine regular papers, and three papers selected from ICIT 2009, 4th International Conference on Information Technology, held during June 3–5 in Amman, Jordan. We would like to use this opportunity to thank the organizers of ICIT, especially the General Chair of the conference, professor Yahya Abdelfatah A., Al-Zaytoonah University, Jordan, Dean of the Faculty of Science and IT, and Program Chair of the conference, professor Ali Al-Dahoud, Al-Zaytoonah University, for assisting in the contribution of high-quality articles to this issue of ComSIS.

The invited paper in this issue, “Advanced Indexing Technique for Temporal Data” by Bela Stantic, Rodney Topor, Justin Terry, and Abdul Sattar, is an extended version of the paper presented at the ADBIS 2010 conference (14th East European Conference on Advances in Databases and Information Systems, September 20–24, 2010, Novi Sad, Serbia). The paper describes a detailed investigation of the “Triangular Decomposition Tree” (TD-Tree): a new approach to indexing temporal data that facilitates open-ended intervals, with the core idea of managing temporal intervals by virtual index structures that rely on geometric interpretations of intervals, and using a space partitioning method that produces an unbalanced binary tree. The authors demonstrate that the single-query algorithm is suitable for different query types, and that the TD-Tree exhibits good space and time complexity compared to the current state-of-the-art.

The first of the regular papers, “COLIBROS: Educational Operating System” by Predrag Rakić, Žarko Živanov, and Miroslav Hajduković, gives an overview of a small, object-oriented, library operating system whose primary purpose is to facilitate teaching a basic operating system undergraduate course. Despite its simplicity, COLIBROS supports high level concurrency and synchronization primitives. The system is also equipped with a hardware emulation layer that emulates keyboard, monitor, disk, and interrupt mechanism. COLIBROS joins

the concepts of classical and object oriented operating systems with multithreading concepts of high-level programming languages through simple and clean interface, helping undergraduate students establish first contact with the problems of concurrency, parallelism and operating systems.

The next regular paper by Miroslav Líška and Pavol Návrát, entitled “An Approach to Project Planning Employing Software and Systems Engineering Meta-Model Represented by an Ontology,” explores the interaction between the model driven architecture (MDA) approach to system development and the semantic Web. The article presents an approach aimed at facilitating the use of software and systems engineering meta-model (SPEM) for improvements in the applicability of MDA standards in the technical space of the Semantic Web that are rooted in knowledge engineering approaches.

In “Facilitating Information System Development with Panoramic View on Data”, Dejan Lavbič, Iztok Lajovic, and Marjan Krisper present Panorama – an associative thinking paradigm for the implementation of a software solution. The foundation of Panorama is an object recognition process, based on context and focus information visualization techniques. The approach focuses on using and updating data vocabulary by users without extensive programming skills.

The article “Metrics for Evaluation of Metaprogram Complexity” by Robertas Damaševičius and Vytautas Štuikys analyzes software complexity management and measurement techniques, and proposes five complexity metrics (Relative Kolmogorov Complexity, Metalanguage Richness, Cyclomatic Complexity, Normalized Difficulty, Cognitive Difficulty) for measuring complexity of metaprograms at information, metalanguage, graph, algorithm, and cognitive dimensions.

Rahim A. Abbaspour and Farhad Samadzadegan, in “An Evolutionary Solution for Multimodal Shortest Path Problem in Metropolises” tackle the problem of finding shortest paths between two locations in large urban areas. The proposed approach presents an adapted evolutionary algorithm, which uses chromosomes with variable lengths and particularly defined evolutionary stages. Empirical evaluation on data representing the city of Tehran, which takes into account three different modes of transportation, demonstrates the validity of the approach.

“Trojan Horses in Mobile Devices” by Daniel Fuentes, Juan A. Álvarez, Juan A. Ortega, Luis Gonzalez-Abril, and Francisco Velasco, examines the behavior of malicious software that attempts to steal information from a mobile device while the user is unaware. The article describes the communication links “Trojan horses” use, the damage that an infection can inflict on a PDA device, and proposes different solution to tackle the problem.

Tsung-Hung Lin, Cheng-Chi Lee, Chwei-Shyong Tsai, and Shin-Dong Guo, in “A Tabular Steganography Scheme for Graphical Password Authentication” propose a novel graphical password authentication protocol that tackles the conflicting requirements of passwords to be secure and easy to remember. The proposed graphical password system can help the user easily memorize their password without loss of security of authentication. The user’s chosen input is hidden into an image using steganography technology, while the authentication server needs only to store a secret key for decryption instead of a large password database.

“A Multi-attribute Auction Model by Dominance-based Rough Sets Approach” by Rong Zhang, Bin Liu, and Sifeng Liu introduces a novel multi-attribute online auction model founded on the dominance-based rough set approach (DRSA). The model resembles a natural reasoning learning method, is able to directly derive the preference relations between the attributes of alternatives, and can reuse the agent’s preferences to increase bidding efficiency.

The article “Design of Median-type Filters with an Impulse Noise Detector Using Decision Tree and Particle Swarm Optimization for Image Restoration” by Bae-Muu Chang, Hung-Hsu Tsai, Xuan-Ping Lin, and Pao-Ta Yu, proposes a method for the recovery of the corrupted gray-level images that first uses an impulse noise detector to determine whether a pixel is corrupted, and then applies the filtering component which consists of three different filters. Noise detection is treated as a classification problem, approached using decision trees coupled with particle swarm optimization (PSO) for optimizing the thresholds to find out the approximate optimized decision tree and the suboptimal solutions for a set of parameters.

The following three articles represent selected and revised versions of papers published in the proceedings of the 4th International Conference on Information Technology (ICIT), June 3–5, 2009, Amman, Jordan.

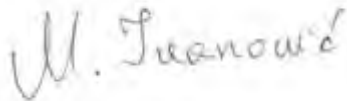
Amjad Rattout, Rasha Assaf, Ali Al-Dahoud, in “Personalising the Dynamic Information Resources in a Self Organized Web System Using CAS and MAS” present a model that uses new Web usage information to explore the effects of usage on semantic values, and how such information can help achieve a robust well personalized and organized Web. The article simulates the usage space environment in the multi-agent system (MAS) and complex adaptive system (CAS) paradigms.

In “A Hybrid Variable Neighborhood Search Algorithm for Solving Multi-Objective Flexible Job Shop Problems” Jun-qing Li, Quan-ke Pan, and Sheng-xian Xie propose a novel hybrid variable neighborhood search algorithm combined with the genetic algorithm (VNS+GA) for solving the multi-objective flexible job shop scheduling problems (FJSPs) to minimize the makespan, the total workload of all machines, and the workload of the busiest machine, demonstrating the success of the approach on well-known benchmarks.

Finally, “Automation of the Moving Objects Movement Prediction Process Independent of the Application Area” by Ivana Nižetić and Krešimir Fertalj contribute to research on moving objects by presenting a conceptual model for movement prediction independent of application area and data model. The authors propose a generic model that addresses the disadvantages of related approaches, and evaluate their method on three case studies.

On behalf of the Editorial Board and the ComSIS Consortium, we would like to thank the article authors for submitting their high-quality contributions, and also the reviewers for the considerable efforts invested into the preparation of this issue of Computer Science and Information Systems.

Editor-in-Chief
Mirjana Ivanović



Managing Editor
Miloš Radovanović



DISCLOSURE OF PLAGIARISM

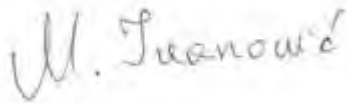
We regretfully announce that the article “Blind Separation Using Second Order Statistics for Non-stationary Signals” by Jun Du, Ju Liu, Shengju Sang and Jun Wang, published in Volume 7, Number 1 of ComSIS (pp. 163–176, 2010), is a confirmed plagiarism of the original work by Chia-Chi Lin, “Blind Source Separation For Non-stationary Signals Using Second Order Statistics,” Master Thesis, National Tsing Hua University (NTHU), Hsinchu, Taiwan, July 2007.

The Editorial Board of Computer Science and Information Systems sincerely apologizes to the author, Chia-Chi Lin, and his advisors, Professor Wing-Kin Ma¹ and Professor Chong-Yung Chi², for allowing the publication of this article. The article has been removed from the ComSIS Web site (www.comsis.org/Archive/Vol7Num1.htm), and replaced by instructions to download the original work.

¹Department of Electronic Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong.

²Institute of Communications Engineering & Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan, R.O.C.

Editor-in-Chief
Mirjana Ivanović



Advanced Indexing Technique for Temporal Data

Bela Stantic, Rodney Topor, Justin Terry, and Abdul Sattar

Institute for Integrated and Intelligent Systems
Griffith University,
Queensland, Australia
{b.stantic, r.topor, j.terry, a.sattar}@griffith.edu.au

Abstract. The need for efficient access and management of time dependent data in modern database applications is well recognised and researched. Existing access methods are mostly derived from the family of spatial R-tree indexing techniques. These techniques are particularly not suitable to handle data involving open ended intervals, which are common in temporal databases. This is due to overlapping between nodes and huge dead space found in the database. In this study, we describe a detailed investigation of a new approach called “Triangular Decomposition Tree” (TD-Tree). The underlying idea for the TD-Tree is to manage temporal intervals by virtual index structures relying on geometric interpretations of intervals, and a space partition method that results in an unbalanced binary tree. We demonstrate that the unbalanced binary tree can be efficiently manipulated using a virtual index. We also show that the single query algorithm can be applied uniformly to different query types without the need of dedicated query transformations. In addition to the advantages related to the usage of a single query algorithm for different query types and better space complexity, the empirical performance of the TD-tree has been found to be superior to its best known competitors.

Keywords: Temporal Databases, Access Methods, Performance Evaluation.

1. Introduction

While being ever changing, time is an important aspect of all real world phenomena. Each event bears a time attached to it, sometimes in more than one form. Time marks the starting and ending of an event and establishes the validity of data. Facts, i.e. data, valid today may have had no meaning in the past and may hold no identity in the future. Some data, on the other hand may hold historical significance or may continue to be valid up to a predefined point in time. This relationship with time adds a temporal identity to most data and, in this light, it would be hard to identify applications that do not require or would not benefit from database support for time-varying data. Most current database systems represent a single state of data and this is most commonly assumed to be its current state. Any modifications normally result in the overwriting of the data with the old data being discarded. Although commercial databases offer some capabilities to keep track of old and discarded data (e.g. the Oracle

TimeSeries cartridge, Oracle “Flash-Back”, and the Informix TimeSeries Data-Blade), this is solely for the purpose of database recovery and not to retain the previous state of the data.

In the last two decades, research on temporal databases has advanced in various aspects and reported many important results, however, many challenges still remain [4], [18]. In most of the previous studies, core concepts have been established, but it is yet to be shown how they can be applied for efficiently managing time dependent data. Because temporal databases are in general append-only and usually very large in size, an efficient access method is even more important than for conventional databases. Numerous access structures were proposed, however, they have usually lacked practical credibility. Moreover most of these structures cannot handle *now-relative* data adequately. This means that these structures assume that the starting and ending points of intervals are known explicitly when recorded into the database. Obviously, this is unrealistic and this particular constraint makes these structures unsuitable for practical temporal applications.

Many multidimensional access structures have been proposed and some of them have been recommended for handling temporal data [12]. The effectiveness of the majority of these index structures has been theoretically evaluated [17]. We classify existing access methods for temporal data into four groups. The first group contains methods which represents extensions of data partitioning spatial indexing structures such as the Segment R-tree [9], 4R-tree [3], or a number of partially persistent methods [12]. In the second group, we have identified modifications of regular B⁺-tree access structures, such as the Fully Persistent B⁺-tree [13] and the Snapshot index [23]. The third group includes techniques based on incremental structures, such as, the Time Index [5], Time Index⁺ [24], and the Monotonic B⁺-tree [16]. Finally, in fourth group are methods which are employing the existing B⁺-tree access structure by mapping of one dimensional ranges to one dimensional points, such as, MAP21 [14], mapping strategy that linearize the data like Interval Space Transformation method (IST) [7] or managing the intervals by two relational indexes the RI-tree [11], [6].

Data partitioning access methods, such as spatial indexes, use a spatial containment hierarchy that clusters data into bounding regions at the leaf level. The nearby internal nodes are then clustered into bounding regions of the parent node forming a hierarchical directory structure. These regions may not represent the entire data space and could overlap. Overlapping is a problem for data partitioning access methods because even for a simple point query it may need to examine multiple paths. When open ended *now-relative* intervals (where the ending point of the temporal interval follows the current time) are represented with widely used maximum timestamp approach a significant overlapping between nodes and dead space causes very poor performance of the index [19], [21].

We intend to propose an access method for temporal data that relies on the exploitation of the relational database systems built-in functionalities; to utilises the native Data Definition Language (DDL), Data Manipulation Lan-

guage (DML); and to use PL/SQL procedure environment within the SQL standard.

A number of access methods for temporal data that utilise the relational database systems built-in functionalities have been proposed, including: MAP21 [14], Time Index [5], Interval B-tree [2], those based on interval space transformation [7] and RI-tree [11]. We observe that the proposed access methods that rely on relational database systems built-in functionalities, such as Time Index, Interval B-tree and those based on interval space transformation IST, have either space complexity problem or are generally tailored to be efficient only for specific query types. In [11], it has been shown that the RI-tree is superior to the Window-List [15], Oracle Tile Index (T-Index) and IST-technique. In the work [10], it was extended and an algorithm for general interval relationships has been presented, but there is still a need to tailor query transformation to the specific query types. It is our intention to propose an efficient access method for temporal data with logarithmic access time and guaranteed minimum space complexity that can answer a wide range of query types with the same query algorithm.

In this paper we present and investigate the “Triangular Decomposition Tree” (TD-tree) access method to index and query temporal data. In contrast to previously proposed access methods for temporal data, this method can efficiently answer a wide range of query types, including point queries, intersection queries, and all nontrivial interval relationships queries, using a single algorithm, without dedicated query transformations.

The TD-tree is a space partitioning access method. The basic idea is to manage the temporal intervals by a virtual index structure that relies on a two-dimensional representation of intervals [20], and a triangular decomposition method. The resulting binary tree stores a bounded number of intervals at each leaf and, hence, may be unbalanced. As data is only stored in leaves, traversing the tree avoids disk accesses, and the tree depth does not affect the performance. Using the interval representation, any query type can be reduced to a spatial problem of finding those (triangular) leaves that intersect with the spatial query region. TD-trees can be implemented on top of a standard relational DBMS.

The efficiency of the TD-tree is due to the virtual internal structure, so there is no need for physical disk I/O's, query algorithms that ensures pruning, and efficient clustering of interval data. On top of the advantages related to the usage of a single query algorithm for different query types and better space complexity, the empirical performance of the TD-tree is demonstrated to be superior to its best known competitors.

The remainder of this paper is organised as follows: In the next Section, we briefly describe several temporal access methods of interest for this discussion and highlight their advantages and disadvantages. In Section 3, we define the mapping strategy and determine regions of interest. Section 4 describes the structure of TD-trees, and the key insertion and query algorithms. Section 5 contains the results and analysis of the empirical study conducted to demon-

strate the practical relevance and efficiency of the TD-tree. Finally, in Section 6, we present our conclusions.

2. Related work

In the literature, two time lines of interest have been mentioned, transaction time and valid time. The valid time line represents when a fact is valid in the modelled world and the transaction time line represents when a transaction was performed. A bitemporal database is a combination of valid and transaction time databases [4]. Because temporal databases are in general append only, they are usually very large in size, thus efficient access method is even more important in temporal databases than in conventional databases. A number of index structures for temporal data are described in the literature [17]. The existing temporal access structures, as highlighted in section 1, fit in one of four groups. We will focus on indexing structures from group four, which can be utilised by exploiting the structures and functionality of commercial RDBMSs and rely on the relational paradigm. We briefly discuss typical representatives from group three and four and highlight their advantages and disadvantages.

The Time Index [5] is an index structure for valid time intervals. It is a set of linearly ordered indexing points that is maintained by a B^+ -tree. To overcome the deficiencies of the Time Index, related to the space requirement which is $O(n^2)$ for storing n intervals, the Time Index⁺ has been proposed [24]. Time Index⁺ relies on efficient storage model for partitioning logical buckets and on employing a new method to handle object versions with long and very long time intervals. The Time Index⁺ requires less storage than the Time Index but still requires significantly more space, even more than the spatial index methods. The disadvantage of this approach is the space required for the index, as for each point in time a bucket of pointers refers to the associated set of valid intervals. Since an interval may be registered with several points in time, This is a problem, particularly for data with many long living tuples.

The Interval B-tree (IB-tree) [2] overcomes the problems related to the extensive space usage of the Time index. It represents an implementation of the Edelsbrunner's interval tree using an augmented B^+ -tree rather than a binary tree. The main memory model of the interval tree is transformed into an efficient secondary storage structure that preserves the optimal space and time complexity. The disadvantage of this approach is the complex three-fold model, which requires a dedicated structure for each level. This makes the IB-tree less attractive from the view point of time complexity.

The access method (ISP) [7] is based on interval space transformation. Since the data space may grow dynamically at the upper bound, this method is well suited for appending intervals. It indexes lists on different orders, start time, end time or duration. This access method is highly specialized with respect to the suggested mapping and can not efficiently answer more complex queries such as intersection query or point query.

The Hierarchical Triangular Mesh (HTM) is method [22] suited for indexing the sphere and especially for astronomy data. It subdivides the half surface of a sphere into four spherical triangles of similar, but not identical, shapes and sizes. Every triangle is further subdivided into four smaller triangles. Division forms a balanced tree, which is then indexed with the Quad-tree. The HTM is highly dedicated for the data that have an inherent location on the celestial sphere. The HTM has been mentioned as it has triangles as a region as our method and to highlight the differences.

The Relational Interval Tree (RI-tree) is an access method for general closed interval data, it can be created for any relational or object-relational table containing intervals [11]. Analytical and experimental evaluation of the RI-tree shows that the performance of this method is superior to the other approaches. This is achieved by introducing a virtual primary structure. Although the structure is space-oriented, the storage of intervals is object-driven so no storage space is wasted for empty regions in the data space. In [10] work was extended and an algorithm for general interval relationships has been presented but still there is a need for tailored query transformation to the specific query types. It is our intention to propose an efficient access method for temporal data that can answer wide range of query types with the same algorithm and that does not require tailored query transformation for different query types.

3. Representation of intervals and interval relationships

We assume a discrete, totally ordered time model with epochs in the range $[0..λ]$, for some (large) $λ > 0$. It is straightforward to map absolute timestamps into such a range of natural numbers, as every Unix system, for example, does. We consider only semi-open intervals $[i_s, i_e)$, where $0 ≤ i_s < i_e ≤ λ$. Each such interval can then be represented as a point (i_s, i_e) in two-dimensional space as shown in Fig. 1. Here, the first coordinate represents the start, S , of the interval and the second coordinate represents the end, E , of the interval. Fig. 1 shows a set of intervals A, B, C and D and their representation in two-dimensional space .

For point data there are only a few distinct query types, *e.g.*, point queries and range queries, but for interval data there are many different query types, *e.g.*. In particular, Allen described 13 distinct interval algebra (IA) relationships that may hold between pairs of intervals [1].

Each of the 13 IA relationships may now be represented as a region, line or point in our two-dimensional space, as shown in Fig. 2. When we study Allen's relationships with indexing and query evaluation in mind, we observe that they fall into two distinct groups.

Relationships between two intervals such as 'start', 'start-by', 'finish', 'finish-by', 'before', 'after', 'meet' and 'meet-by': m_i can be queried efficiently by one dimensional index structures such as B^+ -tree. This is because the problem is reduced to a simple comparison of two points, start or end. However to efficiently answer queries with relationships between two intervals that require the

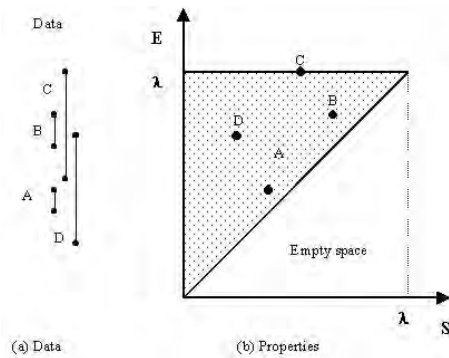


Fig. 1. Interval representation in two-dimensional space

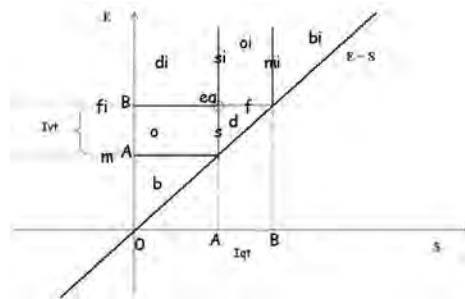


Fig. 2. Allen's 13 IA relationships between two intervals

comparison of both starting and ending points of both intervals such as: 'overlap', 'overlap-by', 'during', 'contain' and 'equal' require special access method

From now on, we focus on the problem of efficiently answering queries about relationships in the second group. We also study queries about the more general 'intersects' relationship and its special case the 'membership' relationship (or 'point' query). The basic query types we consider are queries from group two plus intersection and point query.

If the universe of intervals is $U = \{ [u_s, u_e] \mid 0 \leq u_s < u_e \leq \lambda \}$. Due to the definition of intervals that they are semi-open u_s must be only less than u_e and can not be equal. Then Fig. 3 (a) shows a set of intervals A, B, C, D, a query point at T_0 , and a query interval $I_{qt} = (T_1, T_2)$. The result of each query type above is then a two-dimensional rectangle, or point, as defined below.

- Equality Query EMQ - checks if the database contains an interval which equals the query interval:

$$EMQ([i_s, i_e]) = \{ [r_s, r_e] \mid r_s = i_s \wedge r_e = i_e \}$$

is a point in two-dimensional space;

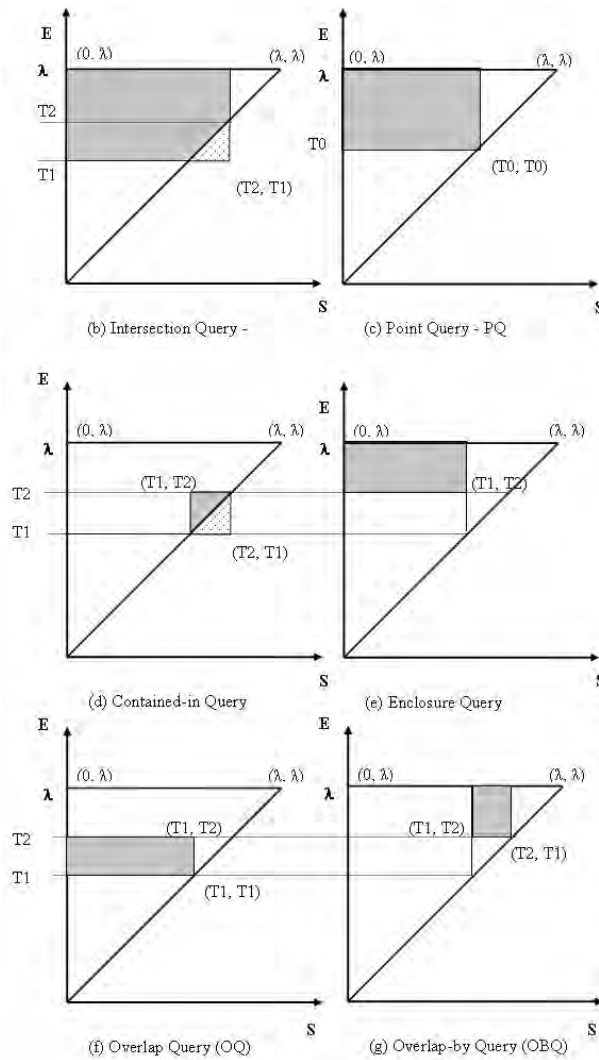


Fig. 3. Query regions

- Intersection Query (IQ), Fig. 3 (b) - finds all intervals that intersect the query interval:

$$IQ([i_s, i_e]) = \{ [r_s, r_e] \mid (r_s < i_e) \wedge (i_s < r_e) \}$$
is a rectangle $[(0, \lambda), (i_e, i_s)]$, in our example $i_s = T_1$ and $i_e = T_2$;
- Point Query (PQ), Fig. 3 (c) - also called timeslice query is a special case of intersection query it finds all intervals that contain the query point: $PQ(p) = \{ [r_s, r_e] \mid r_s < T_0 < r_e \}$ is a special case of IQ and results in the rectangle $[(0, \lambda), (p, p)]$, in our example $p = T_0$;

- Contained-in Query (CQ), Fig. 3 (d) - finds all intervals that are contained in the query interval:
 $CQ([i_s, i_e]) = \{ [r_s, r_e] \mid (i_s < r_s < i_e) \wedge (i_s < r_e < i_e) \}$ can be simplified to $\{ [r_s, r_s] \mid (i_s < r_s < r_e < i_e) \}$, maps to the rectangle $[(i_s, i_e), (i_e, i_s)]$;
- Enclosure Query (EQ), Fig. 3 (e) - finds all intervals that contain the query interval.: $EQ([i_s, i_e]) = \{ [r_s, r_e] \mid (r_s < i_s < r_e) \wedge (r_s < i_e < r_e) \}$ can be simplified to $\{ [r_s, r_e] \mid (r_s < i_s < i_e < r_e) \}$, as $r_s < r_e$ and $i_s < i_e$, and results in the query box $[(0, \lambda), (i_s, i_e)]$,
- Overlap Query (OQ), Fig. 3 (f): $OQ([i_s, i_e]) = \{ [r_s, r_e] \mid (i_s < r_s) \wedge (r_s < i_e < r_e) \}$, maps to the rectangle $[(0, i_e), (i_s, i_s)]$;
- Overlap by Query (OBQ), Fig. 3 (g): $OBQ([i_s, i_e]) = \{ [r_s, r_e] \mid (r_s < i_s < r_e) \wedge (r_s < i_e) \}$, is rectangle $[(i_s, \lambda), (i_e, i_e)]$;

The point of this analysis is that the evaluation of every query type can now be reduced to the spatial problem of finding all data intervals that belong to the rectangle associated with that query. In particular, this means that every query type can be evaluated by a common algorithm, which is what we now study. Note, to form a rectangular query region for particular query type, the query region can extend under the line $E = S$, as for example for intersection query Figure 3 (b) and containment query Figure 3 (d). Because $0 \leq i_s < i_e \leq \lambda$ no intervals will be registered under the line $E = S$ so extending query region under the line $E = S$ to form rectangular query region will not affect the answer.

4. The Triangular Decomposition Tree (TD-tree)

The structure of our indexing method is based on the observation, that all data and query intervals of interest represented in two dimensional space lie in the isosceles, right-angle triangle with vertices at $(0,0)$, $(0, \lambda)$ and (λ, λ) , which lies above the line $E = S$. We call this triangle the *basic triangle* Figure 1. This is due to nature of interval space transformation and fact that $i_s < i_e$.

Given that our region of interest is a triangle, our main proposal is to recursively decompose the basic triangle into two smaller triangles. This triangular decomposition of the basic triangle forms a tree which we call a *TD-tree*. This tree is not balanced in general. Data intervals (points in two-dimensional space) are stored in the database in blocks associated with the leaves of the TD-tree. Figure 4 shows the second and third level of a unbalanced triangular decomposition. Arrows point to the “apex”, the right-angled vertex of the triangle, of each triangle,

In such a triangular decomposition, each triangle is uniquely identified by its *apex* position (s, e) , and its *direction* d , the direction of the arrow from the midpoint of the triangle’s hypotenuse to the apex. Note that there are eight possible directions, corresponding to the eight points of the compass, all of which are shown in Fig. 5.

Given a *Parent* (P) triangle in this decomposition, its apex and direction uniquely determine the apex and direction of each of its two subtriangles. We

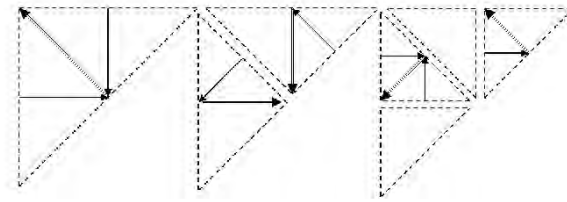


Fig. 4. Unbalanced triangular decompositions of the basic triangle.

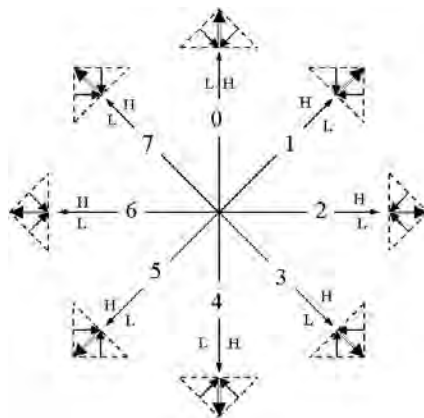


Fig. 5. Positions of low and high subtriangles

call these subtriangles *low* and *high Children* (C). Figure 5 shows, for each possible direction, which are the low and high subtriangles, and where the apexes of these two subtriangles are. Note that we number the possible directions 0 to 7 clockwise starting from direction “north”.

Input: (*P.d*: Parent direction)
Output: *L.d*: Left child direction, *R.d*: Right child direction
begin
 if ($1 \leq P.d \leq 4$) **then**
 $L.d = (P.d + 5) \bmod 8$;
 $H.d = (P.d + 3)$;
 else
 $L.d = (P.d + 3) \bmod 8$;
 $H.d = (P.d + 5) \bmod 8$;
 end
end

Algorithm 1: Children apex directions

By observation of Fig. 5, we see that it is possible to define the apex position and direction of the subtriangles of a given triangle using the following two algorithms. Algorithm 1 computes the direction d of the lower (L) and higher (H) Children (C) subtriangles of a Parent triangle P with direction d .

Input: ($P.s$: Parent start, $P.e$: Parent end, d : direction), $length$ length

Output: $C.s$: Child start, $C.e$: Child end

```

begin
  if ( $d = 0$ ) then
    |  $C.s := P.s$ ;  $C.e := P.e - length$ 
  else if ( $d = 1$ ) then
    |  $C.s := P.s - length/\sqrt{2}$ ;  $C.e := P.e - length/\sqrt{2}$ 
  else if ( $d = 2$ ) then
    |  $C.s := P.s - length$ ;  $C.e := P.e$ 
  else if ( $d = 3$ ) then
    |  $C.s := P.s - length/\sqrt{2}$ ;  $C.e := P.e + length/\sqrt{2}$ 
  else if ( $d = 4$ ) then
    |  $C.s := P.s$ ;  $C.e := P.e + length$ 
  else if ( $d = 5$ ) then
    |  $C.s := P.s + length/\sqrt{2}$ ;  $C.e := P.e + length/\sqrt{2}$ 
  else if ( $d = 6$ ) then
    |  $C.s := P.s + length$ ;  $C.e := P.e$ 
  else
    |  $C.s := P.s + length/\sqrt{2}$ ;  $C.e := P.e - length/\sqrt{2}$ 
  end
end

```

Algorithm 2: Children apex position calculation

Algorithm 2 computes the position (s, e) of the apex of each subtriangle C of a parent triangle P at any level l . This is possible only knowing the position of the parent apex and its level.

From Fig 1 it is straight forward that the basic triangle apex is $(0, \lambda)$ and we accepted that the basic triangle has level 0. Without loss of generality, we may assume that $\lambda = 2^k$, for some $k > 0$. To find the children's apex position the adjustment length that has to be applied to the parent apex position as presented in Algorithm 2. Adjustment length depends only on level of partition l and k . It can be calculated as:

$$length = 2^k * (2^{1/2}/2)^{(l-1)} \tag{1}$$

Note that both child subtriangles of the parent triangle have the same apex position. Position of the child C apex (s, e) will be calculated depending to the direction d of the parent P apex using the Algorithm 2.

Note that the level of the subtriangles of a triangle are one more than the level of the triangle. Note also that the resulting tree need not be balanced. In an unbalanced tree, different leaves may be at different levels. The shape of a tree depends on the distribution and density of data intervals.

Because we can identify the apex and direction of every node of a TD-tree, starting from the basic triangle, using the two algorithms, *we do not need to store the internal tree nodes*. Thus, a TD-tree is a virtual tree. All we need to store is the value λ and a reference to the root node.

The actual data intervals, together with information about the intervals, are stored in a table indexed by a leaf identifier. The tree is organised so that at most b data intervals are stored with each leaf, for some integer blocking factor $b > 0$. A node identifier is a binary string, stored as a (binary) integer, constructed as follows. The identifier of the base triangle or tree root is 1. If a node has identifier ϕ , the lower and upper children of the node have identifiers $\phi 0$ and $\phi 1$ respectively. The length of the identifier is thus one greater than the depth of the node.

Information about leaf nodes themselves are stored in a separate directory, containing an identifier and number of records per leaf. The root node stores the blocking factor b and current maximum depth of the tree l .

4.1. Insertion algorithm

Insertion of data interval into a TD-tree is performed according to Algorithm 3. We first descend the tree from the root to the virtual leaf at maximum tree depth containing the interval. This is done arithmetically, without disk access, by repeatedly selecting the lower or upper child of each node depending on the value of the interval.

The leaf found is called “virtual” because that branch of the tree may have length less than the maximum depth. For example, the upper child of the root node in Fig. 6 labelled ‘g’ is a leaf on a path of length 2, whereas the tree has maximum depth 7, as it can be seen in Table 1.

Input: (object_for_insertion: OBJ , Directory: D , blocking_factor: b , max_population, max_depth)

```

begin
  Find maxregion at max_depth where OBJ would belong;
  target_region = region in D with longest number of bits in common
                  left to right with the maxregion;
  Find target_region population;
  if (target_regionpopulation > max_population) then
    | Increment the population in D of target_region;
    | Update region with target_region;
  else
    | perform Split;
  end
end

```

Algorithm 3: Insertion

Given the sample decomposition from Fig. 6, the directory would be as shown in Table 1. This table shows the label, identifier (in both binary and decimal) and identifier extension (in both binary and decimal) for each leaf of the

tree. The identifier extension is the unique extension of the leaf identifier with zeros so that it is of length l , where l is the maximum depth of the tree. Values for binary identifier and both binary and decimal identifier extension are not stored as they can be calculated from decimal identifier and max depth of the tree 'l'.

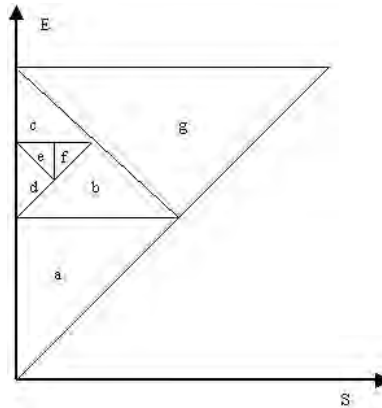


Fig. 6. Running example regular decomposition

Table 1. Directory for sample tree

Label	Identifier (binary)	Identifier (decimal)	Extension (binary)
a	100	4	1000000
b	1010	10	1010000
c	10111	23	1011100
d	101100	44	1011000
e	1011010	90	1011010
f	1011011	91	1011011
g	11	3	1100000

We attempt to insert the data interval into the actual leaf that is an ancestor of the virtual leaf found by the above traversal.

If the identifier of the virtual leaf w containing the interval is z , then the identifier of the actual leaf v that is ancestor of w is given by the longest identifier in the directory that is a prefix of z . For example, if the identifier of the virtual leaf containing the interval is 1010010, then the identifier of the actual leaf in which the interval should be stored is 1010.

Considering the sample from Figure 6 and Table 1, let for example an interval, according to the start and end points, would belong to region on max depth '1010010'. This max depth region '1010010' at the maximum depth doesn't exist so it is required to locate the actual region to store the interval in. That region is given by the longest identifier in the directory that is a prefix of max depth region '1010010' and in our case it is the region '1010'. Having found the region which to store the interval, we simply update leaf identifier in table with that identifier and in directory increment number of records that region holds by one.

To ensure efficient retrieval, we store at most b data intervals with each leaf. If a leaf already has that many intervals, we construct the two children of the leaf, replace the parent with the two children in the directory, distribute the current (and new) intervals between the two children as appropriate, and repeat this process recursively if all intervals go into the same child. If this operation increases the maximum tree depth, we record the new maximum depth. This split is performed according to Algorithm 4.

Input: (SR : Split Region, D : Directory, $blocking_factor$, max_pop , max_depth)

```

begin
  while not both child regions population < max_pop do
    divide data of SR into children; current_depth = SR depth + 1;
    if current_depth > max_depth then
      | max_depth = max_depth + 1;
    end
    if child region is POINT then
      | Exit;
    end
  end
end
end

```

Algorithm 4: Split

It is possible that all intervals in a region that has to be split are located within one newly created smaller region, which will cause a further split. Splits will be performed until intervals can be distributed between two child regions or the maximum split was reached (region represents a point). If maximum split was reached the population of the region is allowed to grow beyond blocking factor, which means that multiple blocks may associate with one region.

4.2. Query algorithm

Following the analysis of Section 3, we can assume that every query corresponds to a rectangular region of the two-dimensional interval space, defined by the top-left and bottom-right corners of this region. The task of the query evaluation is to find all data intervals that occur within this query region. The particular region chosen depends on whether we are performing an intersection query, an overlaps query, a contains query, and so on, but in each case the query evaluation algorithm is identical, an important property of our approach.

Query evaluation itself proceeds in two phases Algorithm 5. In the first phase we find those TD-tree nodes which are contained entirely within the query region and those TD-tree leaves which overlap (but are not contained in) the query region. This phase accesses the disk to retrieve nodes from the directory and to retrieve data intervals from overlapping leaves. In the second phase, we return the intervals in the first set of nodes, and scan the intervals in the second set of leaves for those that occur in the query region. This second phase requires no additional disk access.

The first phase may be implemented as follows. It takes as input the query region Q and the directory D . It returns the set of data intervals that occur within Q .

Input: (D : Directory, Q : Query region)
Output: $A1$: Containing leaves, $A2$: Overlapping leaves

```

begin
  Add all nonempty leaves in  $D$  to a LIST;
  let length  $L$  be 1;
  while LIST is not empty do
    let  $F$  be the first leaf in LIST;
    let  $R$  be the ancestor of  $F$  whose identifier consists
    of the first  $L$  digits of  $F$ 's identifier;
    if  $R$  is contained within  $Q$  then
      add all leaves in LIST with the same prefix as  $R$  to  $A1$  and
      remove them from LIST;
      set  $L$  to 1;
    else if  $R$  is disjoint from  $Q$  then
      remove all leaves in LIST with the same prefix as  $R$  from LIST;
      set  $L$  to 1;
    else if  $R$  equals  $F$  then
      add  $R$  to  $A2$  and remove it from LIST;
    else
      increment  $L$ ;
    end
  end
end

```

Algorithm 5: Query algorithm

This phase terminates with $A1$ containing the set of nodes whose descendent leaves are contained entirely within Q , and $A2$ containing the set of leaves which overlap Q . By testing whether the ancestor R of F is contained within Q , we can select all leaves under R in one operation. This property of our algorithm significantly reduces the number of disk accesses and improves its overall performance. To test whether a triangle is contained within a rectangle or whether a triangle intersects a rectangle are straightforward geometric operations based on the vertices of the two operations.

On Figure 7 and 8 virtual region '100' (dark grey) lies completely outside the region for point query at T_0 , and for that reason all leaf regions that are children of virtual region '100' are excluded from the answer (10001, 100000, 100001,

10010, 100110, 100111). This bulk exclusion improves the query efficiency. On Figure 7 and 8 we show total exclusion and total inclusion with the dark and light grey respectively.

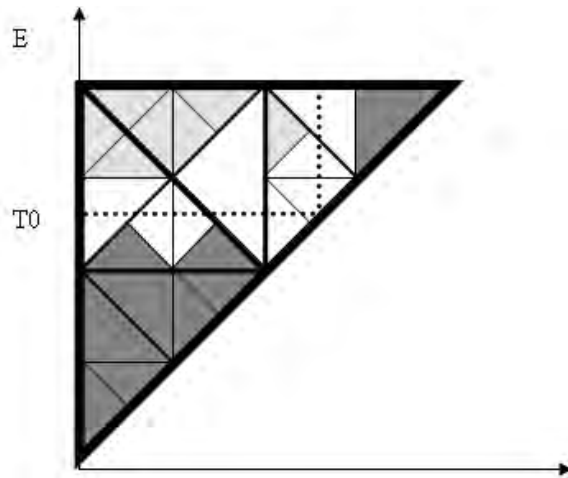


Fig. 7. Point query on transformed region

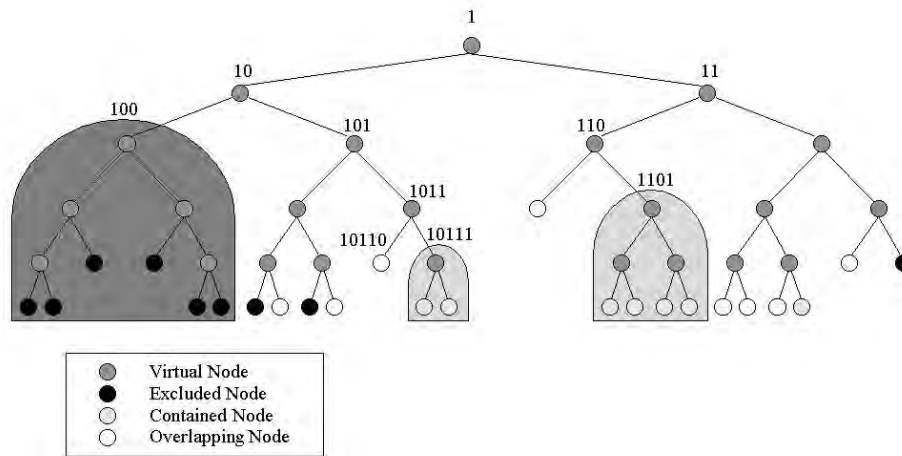


Fig. 8. Unbalanced Binary tree

In the second phase, it suffices to return all data intervals in all descendent leaves of nodes of \mathcal{O}_1 and to scan all data intervals in all leaves of \mathcal{O}_2 , returning those intervals that occur within \mathcal{Q} . This latter test is a simple arithmetic comparison. No additional disk accesses are required in this phase.

Deletion Algorithm removes regions from the directory that contain zero objects due to the decrement of population. Also, this algorithm merges two children into parent region if sum of population of both children falls under the one third of the blocking factor.

```

Input:  $D$ : Directory, object_for_deletion, blocking_factor)
begin
    delete_region = region where object_for_deletion belongs; Delete
    object_for_deletion; Decrement the population of delete_region; if
    combined population of delete region and its sibling < blocking_factor/3
    then
        | merge two children into parent regions;
    end
end

```

Algorithm 6: Deletion

Update can be seen as delete and insert and therefore is handled by *Deletion* and *Insert* algorithms.

It is straightforward to show that the query evaluation algorithm of the previous subsection returns all data intervals that occur within the query region and only these. In the interests of brevity we omit the details. It is perhaps more important to note the following complexity result.

Proposition 1: An intersection query on a TD-tree with blocking factor b and n data intervals with answer size of phase one of the query algorithm a having the directory size m , performs $O(m/b + \log_b n + a/b)$ disk accesses.

Proof Because the TD-tree has no internal nodes, to find the answer size of phase one of the query algorithm a having the directory size m and block size b , $O(m/b)$ I/O complexity is performed. Additionally, scanning the index organised table has I/O complexity of $O(\log_b n)$ and reporting the total of a results requires $O(a/b)$ operations. Therefore, the I/O complexity for TD-tree is $O(m/b + \log_b n + a/b)$.

Note that in worst case scenario, due to the secondary filtering of phase two of the query algorithm, a can be equal to n . Similarly to any query algorithm that has secondary filtering in worst case scenario I/O complexity can be basically $O(n/b)$. However, in our experiments we could not replicate such scenario.

In Table 2, we present the complexity cost for TD-tree and several other access methods of interest. a denotes the size of the point query while n represents the number of interval objects. For RI-tree h is a virtual backbone tree that corresponds to the current expansion and granularity of the data space but does not depend on n . Because there is no internal nodes in TD-tree and the size of the directory is significantly smaller than the data table, the complexity cost for TD-tree practically depends only on answer size. Our experimental results support this claim.

Table 2. Performance analysis of access methods of interest

<i>Access Method</i>	<i>Space Usage</i>	<i>Point Query</i>
<i>R - Trees</i>	$O(n/b)$	$O(n/b)$
<i>Time Index</i>	$O(n^2/b)$	$O(\log_b n + a/b)$
<i>Bitemp.R - Tree</i>	$O(n/b)$	$O(\log_b n + a/b)$
<i>RI - tree</i>	$O(n/b)$	$O(h \log_b n + a/b)$
<i>TD - tree</i>	$O(m/b)$	$O(m/b + \log_b n + a/b)$

5. Experimental evaluation

To show the practical relevance of our approach, we performed an extensive experimental evaluation of the TD-tree and compared it to the RI-tree [11].

The RI-tree was chosen, since it provides the same practically important properties as our approach. It is easy to implement and integrate, it uses standard RDBMS methods which provides scalability, update-ability, concurrency control and space efficiency. Furthermore it has been proven [11] that the RI-tree is superior to the Window-List [15], Oracle Tile Index (T-Index) and IST-technique [7] so performance results of the TD-tree can be transferred to these indexing techniques. We could not compare our TD-tree with improved implementation of RI-tree [6] as it indexes Interval-and-Value tuples together while our method only index intervals.

All experimental results presented in this section are computed on eight 850MHZ CPU - SUN UltraSparc II processor machine, running Oracle 10.2.0 RDBMS, with a database block size of 8K and SGA (System Global Area) of 500MB. At the time of testing database server did not have any other significant load. We used Oracle built-in methods for statistics collection, analytic SQL functions and the PL/SQL procedural runtime environment.

5.1. Data sets

In order to simulate different real applications scenarios we used different data distributions. The start position of the intervals was always uniformly distributed on the interval domain, while the duration was varied. Following data distributions have been considered:

- Uniformly distributed start and uniform distributed length within the range [1, 10000] with 20% of uniformly distributed *now-relative* data.
- Uniformly distributed start and exponentially distributed length according to the exponential distribution function $y = e^{-0.00041*x}$ with 20% of uniformly distributed *now-relative* data.

Uniform distribution of interval start, appearance of *now-relative* data and exponential distribution of the duration reflects most real world applications

where short intervals are more likely to occur than long intervals. We used maximum timestamp approach to represent current time. Furthermore, in real world applications there is usually a upper bound for the interval duration and in our case we have chosen 10,000 (days) for the upper bound, not considering *now-relative* data, which are represented with maximum timestamp approach.

All data set distributions had separate relations with different number of tuples, 250,000, 500,000 and 1,000,000.

5.2. Query sets

In our experiment we tested performance on intersection queries and particularly on point query as its specific case. Because of the nature of our query algorithm, by comparing the data region with the rectangular query region, as has been shown in subsection 3, results for performance evaluation apply to the other query types.

The point query that timeslices the timeline at the current time was used to determine how access method performs with *now-relative* data. The point query that timeslices the time line at the current time is considered to be the most important because most often we will ask queries about the current state of reality.

5.3. Update sets

Most often updates in Temporal databases happen when facts cease to be valid (in valid time databases) or tuple is logically deleted (in transaction time databases). In both cases ending time of interval that contain semantic for 'now' (*now-relative* data) is replaced with the current time. We tested performance of our TD-tree on updates of randomly selected *now-relative* interval data of 100 tuples. As explained in update algorithm to perform update it is required to perform delete from the previous region and insert interval into the new region.

5.4. Experiments

The same data set is used both for RI-tree and TD-tree testing experiment. The initial relations with structure Employment(ID, Name, Position, Start, End) were replicated and altered accordingly to suit each particular method.

Relations for testing the performance of the RI-Tree, were altered with column *node*, which is calculated for every row of data by algorithm as explained in paper [10]. Two B⁺-tree composite indexes have been created *LowerIndex* (node, Start) and *UpperIndex* (node, End). A point query is performed by calling the dedicated procedure that collects leftnodes and rightnodes and then performs the transformed SQL statement as instructed in [11].

Relations for testing the performance of the TD-tree were altered with column *Region*, which is calculated according the algorithm as explained in Subsection 4.1. The root node, which contains information for λ , blocking-factor,

adjustment date and maximum depth of the tree we stored as one tuple relation. Because TD-tree has only leaf nodes it can be organised as a list and stored in directory tables. To ensure that the population of a region corresponds to one block, so it can be efficiently retrieved, we introduced a blocking factor. We built relation Employment as index organised table using *Region* and *ID* as a primary key.

5.5. Results

To compare the space requirements for RI-tree and TD-tree, we considered tables with different number of rows. We generated tables with 250,000 rows, 500,000 rows and 1,000,000 tuples. All tables are altered to suit the particular approach and all required primary and secondary indexes are created. In Fig. 9 we show the space requirement for the TD-tree and RI-tree. Results represent the sum of used space for table, primary/secondary indexes and for the TD-tree we also added required space for the directory table.

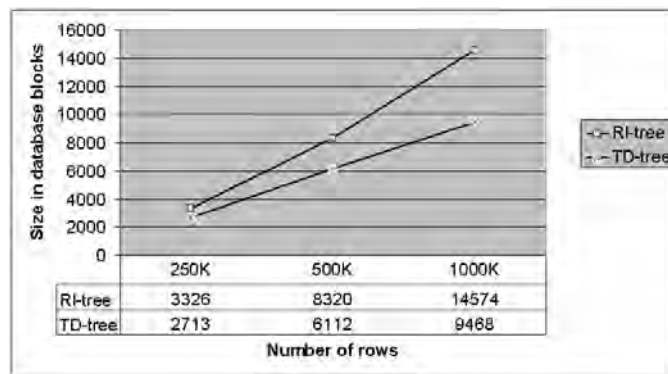


Fig. 9. Comparison of the space usage (Table plus indexes)

To measure the query performance we used a data set of one million tuples. Results shown in Fig. 10 are for the point query with uniformly distributed start and exponentially distributed length with 20% of *now-relative* data. Results represent disk I/O and average CPU usage for different points on timeline which contain different answer sizes. We performed tests with all data distributions mentioned in subsection 5.1, but testing resulted in similar qualitative results as those presented here.

The Theory of Indexability [8] identifies I/O complexity cost, measured by the *number of disk accesses*, as one of the most important factors for measuring query performance. Other measures of importance such as *CPU usage* and query response time are also used in conjunction with the number of disk accesses to assess the performance of the query processes.

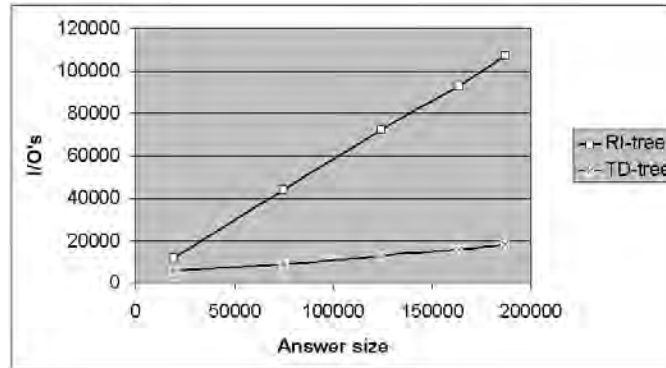


Fig. 10. Physical disk I/O as a factor of answer size

Table 3. Average number of answers per one Physical disk I/O

<i>Answer Size</i>	<i>TD – tree DiskI/O</i>	<i>Answers/ DiskI/O</i>	<i>RI – tree DiskI/O</i>	<i>Answers/ DiskI/O</i>
19365	6102	3.17	11902	1.63
74727	8952	8.35	43891	1.70
124280	12958	9.59	72426	1.72
163530	16012	10.21	92776	1.76
186795	18054	10.35	107068	1.74

For the TD-tree the number of leaf regions accessed to answer the query is simply the number of regions returned in the *Primary* filter. *Secondary* filtering only does pruning so it does not require any additional disk access, it only adds CPU usage. When the answer is smaller, interval objects pruned with the secondary filter effect the performance of the TD-tree and number of answers per one physical disk I/O is relatively smaller. In Table 3 we can see that TD-tree even for a small answer size has better factor of answers per physical disk reads. For the RI-tree the number of answers per physical disk read is not dependent on query load, however for the TD-tree, due to the secondary filter features, the number of answers per physical read is dependent on query load and reaches the best performance on larger query loads.

Beside the queries mentioned in subsection 5.2, we tested applicability and performance of the TD-tree on several other query types, such as during, contain, and even before and after. These results will be mentioned and analysed in the next subsection.

The TD-tree performs well on *now-relative* data using *MAX* approach to represent current time, because the area where these intervals are stored can be divided as often as required as shown in Figure 12. If we represent interval objects that belong to the particular RI-tree nodes in two dimensional space, it can

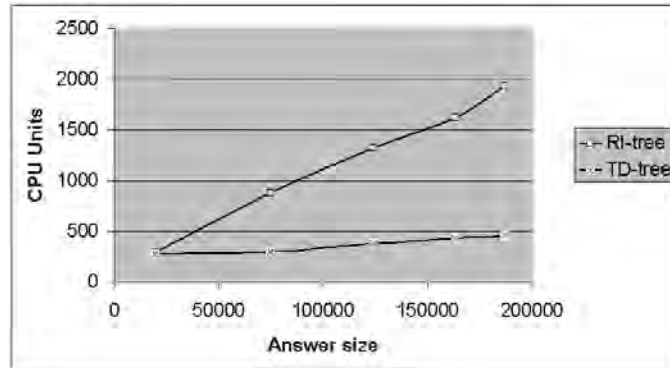


Fig. 11. CPU usage as a factor of answer size

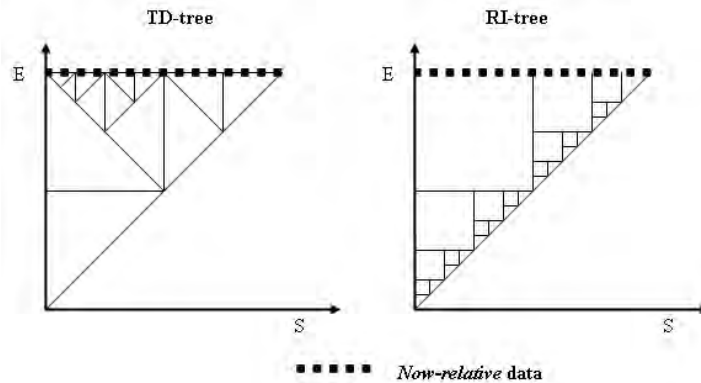


Fig. 12. Now-relative data in TD-tree and RI-tree

be seen that all *now-relative* data will be located in the root node (biggest rectangle in Figure 12) or on the right edge of the RI-tree. Knowing the importance of *now-relative* data in temporal databases this is a significant constraint.

5.6. Comparative analysis

When making performance measurements of index structures, it is important to not only consider response time but also other parameters such as space requirements, clustering, CPU usage, updates, and locking. In our analysis, we have concentrated on space requirements, physical disk reads, CPU usage and clustering of data. Because both the RI-tree and TD-tree rely on the relational paradigm, updates and locking are handled well by the RDBMS itself.

The TD-tree requires only one virtual index structure, which means only leaf nodes have to be stored. The list of leaf nodes are stored in the directory table and its size is very small comparing to the table itself. In our experiment the TD-tree directory for one million interval objects required only 26 data blocks. In addition to directory table there is a need for extra space considering that table is index organised by region, which is comparable with the primary index of RI-tree method.

The RI-tree requires two composite index structures *lowerIndex* and the *upperIndex*. One composite index is on *node* and *Start* - start of the interval, and another composite index is on *node* and *End* - end of the interval. The size of the indexes depend on the number of interval objects and in our experiment one million interval objects required 6708 data blocks (3354 each index), which is significantly bigger than the 26 blocks required for the TD-tree directory. For this reason, the total number of blocks required for table and index structures for TD-tree is much smaller than the number of total blocks required for RI-tree. This difference increases with increasing number of interval objects, as shown in Fig. 9.

The TD-tree enables efficient usage of clustering of the data by one dimension, i.e region, as every region associate with block size. Clustering data improves the query performance and reduces the number of physical I/O, as shown in Table 3, clustering ensures higher number of answers per physical disk I/O. In contrast, the RI-tree can not efficiently use clustering of data as it has to decide which dimension to use start or end. If it is clustered by *node* it will not result in similar improvements, as in RI-tree *node* are fixed size and are too large to provide effective clustering.

In Figure 10 it can be seen that the virtual structure of the TD-tree, clustering of data and the query algorithm significantly reduces the physical disk I/O reads. This is particularly the case when the answer size is bigger due to the good clustering, which is achieved by dividing the regions as often as needed.

The TD-tree performs as good on *now-relative* data as on any other data. We could not notice any difference in the number of physical disk I/O's and CPU usage for a point query, which timeslices time line at the current time and at any other time point on the time line. This is because the area where these intervals are stored can be divided as often as required.

Our experimental testing shows that the TD-tree query algorithm performs well on other query types such as: during, contain, before and after. This is because it compares the data region with the query region and uses the same algorithm. It is important to mention that the RI-tree needs dedicated query transformation for specific query type. Despite the TD-tree performing well on before and after query types it has worse performance in comparison with the straight forward usage of one dimensional indexes, as was anticipated and highlighted in section 3. The TD-tree does not perform well on query types such as start and finish because the query region is a line. However, these query types can be efficiently answered with one dimensional index, which is also highlighted in section 3.

6. Conclusions

We described a new approach that is demonstratively better than existing approaches for handling temporal, and more generally, interval data. More specifically, in this study, we:

- Presented a two-dimensional interval space representation of intervals and interval relationships to reduce all interval relationship problems to simpler spatial intersection problems;
- Showed that a wide range of interval query types can be reduced to an intersection of data with a rectangular region, so one algorithm can be applied uniformly;
- Proposed the “Triangular Decomposition Tree” (TD-tree) and associated algorithms that can efficiently answer a wide range of query types including point or timeslice queries;
- Experimentally evaluated the TD-tree by comparing its performance with RI-tree, and demonstrated its overall superior performance.

The TD-tree is a unique access method as it uses tree structures, and at the same time has some characteristics of hashing approaches due to it only stores data in leaf nodes. In contrast to hashing methods that do not perform well on range queries, the TD-trees can efficiently answer a wide range of different query types. It is important to mention that the management of the virtual structures is done automatically by using database triggers, which fire on insert, calculates, updates the region of the record and also increments the region in the directory. If required, it also initiates and performs splits. Similarly, database triggers fire on updates/deletes and performs actions in line with Deletion and Insert Algorithms.

As a wide range of query regions of interest can be reduced to rectangles, it is possible to answer such queries using a single algorithm without requiring any query transformation. This itself, and the fact that the TD-tree can be incorporated within commercial RDBMS, makes the TD-trees superior to other methods proposed for temporal data.

References

1. J. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.
2. C. Ang and K. Tan. The Interval B-tree. *Information Processing Letters*, 53(2):85–89, 1994.
3. R. Bliujute and et al. Light-Weight Indexing of General Bitemporal Data. In *Statistical and Scientific Database Management*, pages 125–138, 2000.
4. C. Date, H. Darwen, and N. Lorentzos. *Temporal Data and the Relational Model*. Morgan Kaufmann, 2002.
5. R. Elmasri, G. Wu, and Y. Kim. The time index: An access structure for temporal data. *Proc. 16th Conf. Very Large Databases*, pages 1–12, 1990.

6. J. Enderle, N. Schneider, and T. Seidl. Efficiently processing queries on interval-and-value tuples in relational databases. In *VLDB '05: Proceedings of the 31st international conference on Very large data bases*, pages 385–396. VLDB Endowment, 2005.
7. C. H. Goh and et al. Indexing temporal data using existing b+-trees. *Data and Knowledge Engineering*, (18):147–165, 1996.
8. J. Hellerstein, E. Koutsupias, and C. Papadimitriou. On the Analysis of Indexing Schemes. *16th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, 1997.
9. C. Kolovson and M. Stonebraker. Segment indexes: Dynamic indexing techniques for multi-dimensional interval data. *Proc. ACM SIGMOD*, pages 138–147, 1991.
10. H.-P. Kriegel, M. Potke, and T. Seidl. Object-relational indexing for general interval relationships. In *Proc. 7th Intl Symposium on Spatial and Temporal Databases (SSTD01)*, 2001.
11. H.-P. Kriegel, M. Ptke, and T. Seidl. Managing intervals efficiently in object-relational databases. *Proceedings of the 26th International Conference on Very Large Databases*, pages 407–418, 2000.
12. A. Kumar, V. Tsotras, and C. Faloutsos. Designing Access Methods for Bitemporal Databases. *IEEE Transactions on Knowledge and Data Engineering (TKDE'98)*, 10(1):1–20, 1998.
13. S. Lanka and E. Mays. Fully persistent b + trees. *Proc. ACM SIGMOD Conf. on the Management of Data*, pages 426–435, 1991.
14. M. A. Nascimento and M. H. Dunham. Indexing Valid Time Databases via B^+ -Tree. *IEEE Transactions on Knowledge and Data Engineering*, 11(6):929–947, 1999.
15. S. Ramaswamy. Efficient Indexing for Constraint and Temporal Databases. In *Proceedings of the 6th International Conference on Database Theory*, pages 419–431, 1997.
16. R. Elmasri, G. Wu, and V. Kouramajian. The Time Index and the Monotonic B^+ -Tree. In *A. Tansel et al., editors Temporal Databases: Theory Design and Implementation*, pages 433–456, 1993.
17. B. Salzberg and V. J. Tsotras. Comparison of Access Methods for Time Evolving Data. *ACM Computing Surveys*, 31(1), 1999.
18. R. T. Snodgrass. *Developing Time-Oriented Database Applications in SQL*. Morgan Kaufmann, 2000.
19. B. Stantic, S. Khanna, and J. Thornton. An Efficient Method for Indexing Now-relative Bitemporal data. In *Proceeding of the 15th Australasian Database conference (ADC2004), Denidin, New Zealand*, 26(2):113–122, 2004.
20. B. Stantic, J. Terry, R. Topor, and A. Sattar. Indexing Temporal Data with Virtual Structure. In *Advances in Databases and Information Systems - ADBIS2010*, pages 591–594, 2010.
21. B. Stantic, J. Thornton, and A. Sattar. A Novel Approach to Model NOW in Temporal Databases. In *Proceeding of the 10th International Symposium on Temporal Representation and Reasoning (TIME-ICTL 2003), Cairns*, pages 174–181, 2003.
22. A. S. Szalay, J. Gray, G. Fekete, P. Z. Kunszt, P. Kukol, and A. Thakar. Indexing the Sphere with the Hierarchical Triangular Mesh. *CoRR*, abs/cs/0701164, 2007.
23. V. J. Tsotras, B. Gopinath, and G. Hart. Efficient management of time-evolving databases. *IEEE Trans. Knowledge and Data Eng*, 7(4):591–608, 1995.
24. V. Kouramajian and et.al. The time index+: An incremental access structure for temporal databases. In *Proceedings of the Third International Conference on Knowledge and Data Engineering (CIKM'94)*, pages 232–242, 1994.

Dr Bela Stantic is Senior Lecturer at the School of Information and Communication Technology, Griffith University, and member of the Institute for Integrated and Intelligent Systems (IIIS). He is also a Senior Researcher at National ICT Australia (NICTA) Queensland Research Lab (QR). He has been an academic staff member at Griffith University since 2001. His research interests include efficient management of complex data structures, temporal and spatio-temporal databases, bioinformatics, and database systems.

Rodney Topor has been a Professor of Computing Science in the School of Information and Communication Technology at Griffith University since July 1991. Prior to that he worked in the Computer Science departments at The University of Melbourne and Monash University. He has served as Head of School and in other management roles at Griffith. His research interests include programming methodology, programming languages, database systems, knowledge representation and Web application development.

Dr Justin Terry is member of the Institute for Integrated and Intelligent Systems (IIIS). He completed his PhD in 2010 and his topic was on indexing multi-dimensional data. His research interest includes efficient management of high-dimensional data.

Professor Abdul Sattar is the founding Director of the Institute for Integrated and Intelligent Systems and a Professor of Computer Science and Artificial Intelligence at Griffith University. He is also a Research Leader at National ICT Australia (NICTA) Queensland Research Lab (QRL). He has been an academic staff member at Griffith University since February 1992 within the School of Information and Communication Technology. His research interests include knowledge representation and reasoning, constraint satisfaction, intelligent scheduling, rational agents, propositional satisfiability, temporal reasoning, temporal databases, and bioinformatics.

Accepted: October 20, 2010.

COLIBROS: Educational Operating System

Žarko Živanov¹, Predrag Rakić¹, and Miroslav Hajduković¹

¹Faculty of Technical Sciences, Trg D. Obradovića 6,
21000 Novi Sad, Serbia
{zzarko, pec, hajduk}@uns.ac.rs

Abstract. This paper gives an overview of educational operating system called COLIBROS. It is small, object oriented, library operating system, based on micro-kernel concepts, supporting high level concurrency and synchronization primitives. In fact, COLIBROS is simplified operating system kernel accompanied with hardware emulation layer that emulates keyboard, monitor, disk and interrupt mechanism. A concurrent COLIBROS program behaves like stand alone program executing in emulated environment, in our case as plain GNU/Linux process. Encapsulating all critical concepts in host operating system user space makes COLIBROS development and debugging easier and more user friendly.

Keywords: Operating System, Programming Library, Education.

1. Introduction

In order to teach basic operating system undergraduate course, specially designed educational operating system is useful. This operating system should be simple enough that it can be presented with all details to students in one-semester course. Also, it should give students insight in all basic operating system concepts like multitasking, synchronization, memory management, interrupt handling, device controlling and preemption. So, we developed COLIBROS (COncurrent LIBRARY Operating System) [15], educational operating system with these properties.

In the rest of this paper COLIBROS project is discussed in more details. Rationale for COLIBROS development is presented in chapter two, followed by project history in chapter three. COLIBROS implementation details such as modules, core implementation and hardware emulation are explained in chapter four. Overview of COLIBROS interface is given in chapter five. This chapter focuses on multithreading, thread synchronization, atomic variables and device controlling mechanisms provided by COLIBROS. Our conclusions and further development directions are presented in chapter six, followed by references in chapter seven.

2. Rationale for COLIBROS Development

In introductory operating system course concurrency is the key new concept that distinguishes operating system from other "ordinary" programs. Without mastering concurrency concepts it is really difficult for students to comprehend operating system behavior and functionality. Consequently, we decided to put concurrency in the center of students attention during operating system course. Examples of concurrent problems are used to give students insight in operating system internals. Though, concurrency is placed in the center of student's attention, we think that well known educational concurrency tools, like BACI [3, 4] or Hartley's java library [17], are not best shaped for operating system course because they are not adequate to present other operating system aspects.

To support our approach, we developed COLIBROS. It is designed to support execution of student programs in emulated environment on GNU/Linux platform. Students, by solving all sort of concurrent problems, in fact write different parts of operating system. We are convinced that this offers invaluable experience that prepares student minds to accept "real" problems. On that basis, it is easy to complete the whole operating system picture and help students navigate in huge number of its details.

COLIBROS offers small and simple kernel suitable for students' projects aimed to change and improve its functionality. It is similar to the well-known operating system courses during which students improve minimal kernel through assignments [2, 12, 18, 13].

Our approach differs from other well-known operating systems developed as educational tools, which are designed as fully functional UNIX-like operating systems like MINIX [24] or XINU [7]. Using our approach we direct student attention only to fundamental concepts, supported by fairly small implementation (COLIBROS core consists of 1500 lines).

There is resemblance between COLIBROS and exokernel [10, 11] library operating system. Difference is that COLIBROS is educationally oriented and it stands above hardware emulation layer (virtual machine) instead of exokernel. Without hardware emulation, standalone COLIBROS program can be executed on bare hardware. In that constellation, COLIBROS core represents simplified operating system kernel executing in the same address space with application.

Since COLIBROS is minimal operating system kernel, it offers realistic insight into crucial aspects of operating system behavior. Modifying COLIBROS core students can change system behavior influencing process, memory management or device handling.

COLIBROS exports object oriented, high level programming concurrency abstractions interface. The same abstractions are used to build operating system kernel modules. Distinguishing feature of COLIBROS is that it can be used in two ways, for teaching concurrency in high-level programs and for teaching operating system kernel internals. COLIBROS simplified operating system kernel uses threads and their synchronization to provide some typical kernel functionalities.

3. COLIBROS Project History

COLIBROS history begins with conCert (CONcurrent C for Embedded RealTime) [14]. The conCert project started in early 1990-es with intention to develop small, educational tool for operating system course at Computer and Control Department of Faculty of Technical Science at University of Novi Sad. Its goal was to support concurrency in simple multiprocessing environment. The conCert was designed as monolithic but well structured (layered), single space, library operating system. It was developed on DOS platform, using Borland Turbo C compiler, like XINU [7] or MPX-PC [20]. Putting everything in one address space, without any hardware protection between threads and kernel, made it simple and fast. During 1990-es conCert was ported to several Intel based platforms (80x86, 80960) and used as real time executive for control industrial applications.

Couple of years later, conCert was renamed to COLIBRY, redesigned to become object oriented and migrated to C++ language. Since then, COLIBRY uses objects to represent system elements (*thread, memory, ready_list...*). Concurrency (thread definition and creation) and synchronization (cooperation between threads and cooperation between threads and interrupt handlers) are implemented using C++ classes, similar to Choices [5, 6], instead of introducing new language primitives.

COLIBRY programs were executed on DOS without any hardware protection (similar to [22]). Students had difficulties developing and debugging programs in such hostile environment. Thus, COLIBRY project was migrated again [21] to GNU/Linux platform and GNU gcc compiler and renamed to COLIBROS.

During this migration it was necessary to introduce hardware emulation layer between COLIBROS core and Linux kernel. This emulation layer emulates only devices (terminal, disk) and mechanisms (interrupt handling) that are not directly/completely accessible to unprivileged Linux process. Our goal was to help students to understand real hardware and master its asynchronous behavior of hardware devices/mechanisms.

In the latest version of COLIBROS, students can use GNU/Linux memory protection and debugging tools for safe COLIBROS program development and at the same time they can freely access (emulated/real) hardware devices.

4. COLIBROS Implementation

COLIBROS consists of two layers: core (COLIBROS kernel) and hardware emulation layer. Every COLIBROS program is statically linked with COLIBROS (kernel core and emulation layer) and executed as plain GNU/Linux process (as shown in Fig 1). This organization allows independent COLIBROS processes to be executed at the same time on the same GNU/Linux host.

COLIBROS kernel is set of library functions/objects, which are linked with user code, similar to Engler's exokernel [10]. That's why it is referred to as Library Operating System. This organization allows different kernel implementations/configurations to be chosen at compile time.

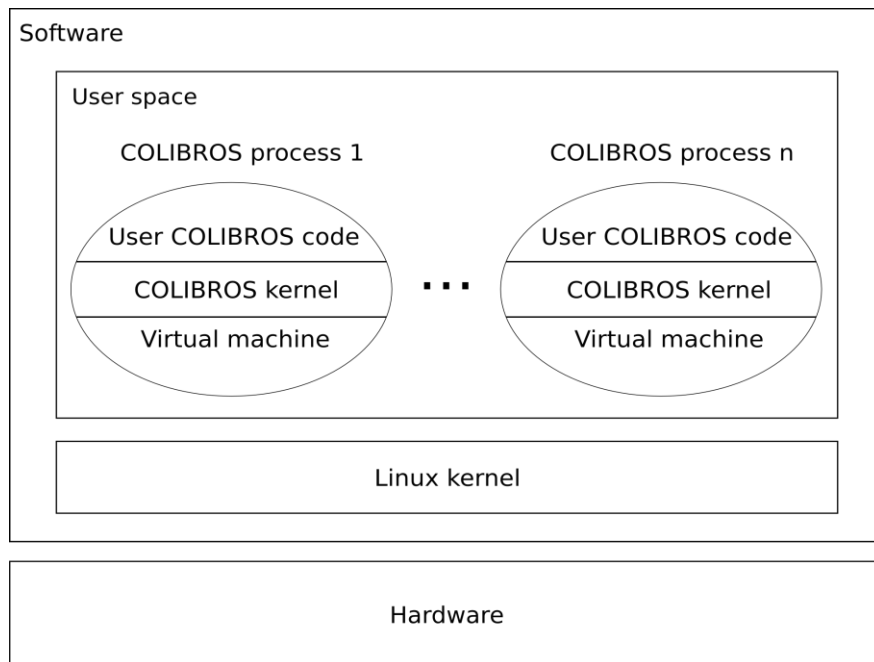


Fig 1. COLIBROS environment. Many COLIBROS processes (potentially created by different users) can coexist on the same GNU/Linux host. Every COLIBROS process executes in user space and encapsulates its own instance of emulated hardware layer.

COLIBROS source code is divided in modules that represent logical parts of COLIBROS project.

4.1. COLIBROS Modules

COLIBROS core consists of *exec* and *in_out* modules.

The *exec* module contains platform independent COLIBROS executable source code. For every platform there should be also platform specific code. Currently there is only one platform supported: i386-pc-linux.

The *in_out* module contains character and block drivers for devices used by COLIBROS. Only terminal and disk are currently implemented.

The *exec* and *in_out* modules (without emulation layer) contain around 1500 lines of code and can be executed in couple hundred KBs of RAM.

Besides these two modules COLIBROS also contains: *tests*, *measures* and *programs* modules.

The *tests* module contains carefully designed automatic and manual test programs used during development of COLIBROS.

The *measures* module contains COLIBROS programs used to measure efficiency of different COLIBROS implementations.

The *programs* module contains examples discussed through out course. Examples demonstrate basic concurrent and parallel problems like Producer/Consumer, Dining Philosophers, Readers/Writers, Disk Head Scheduling, Parallel Sorting, Matrix Multiplication, Parallel Contour Finding and so on.

At the end it should be noted that COLIBROS is accompanied with operating system course book [15] contained in *doc* module. This book contains COLIBROS reference manual and its source code with explanations.

4.2. COLIBROS Core Implementation

COLIBROS core implementation is based on several system objects. The objects *ready* and *kernel* are in charge of processor and numerical co-processor management, and support scheduling and synchronization. These objects support context switching, preemptive round robin scheduling, mutual exclusion and conditional synchronization with sorting of thread waiting lists as well as dealing with asynchronous events (interrupts).

The objects *exception* and *timer* represent drivers that take care of exceptions and system time.

The object *memory* is due to memory management. It supports contiguous first fit memory allocation as well as C++ new and delete operators.

The objects *delta* and *wake_up_daemon* deal with thread sleeping.

Character and block input and output are supported by the objects *display_driver*, *keyboard_driver* and *disk_driver*.

These system objects constitute simplified operating system kernel suitable for changing and extending. So, during the operating system course different improvements (like advanced scheduling mechanisms, memory management techniques (e.g. virtual memory) or file system) can be added to basic COLIBROS functionality.

Other, higher level, services can be built as user threads (similar to micro-kernel architecture [24]) without changes in COLIBROS core.

4.3. COLIBROS Hardware Emulation Layer

The hardware emulation layer is added to COLIBROS during migration to GNU/Linux platform. Using it COLIBROS program is executed as plain GNU/Linux process that freely accesses only emulated hardware and therefore there is no need for real hardware access privileges.

The emulation introduces system objects that represent keyboard, display and disk controller as well as terminal. Another three system objects represent emulated interrupt table and deal with Linux signals and Linux timer.

Emulated keyboard and display controllers are software objects that together represent terminal (serial device) in raw mode. This emulation is implemented using GNU/Linux terminal primitives.

Emulated disk controller is software object that represents hardware magnetic disk device. This controller emulates small capacity, block device which stores its data in COLIBROS process's memory. It, also, emulates some physical characteristics of magnetic disk like seek time and rotational delay.

Emulation of interrupt mechanisms requires emulation of interrupt notification mechanism, interrupt table and interrupt flag. GNU/Linux signals are used to emulate interrupt notification. Signals are designed for inter-process asynchronous communication and are well suited for emulation of interrupt notification.

Interrupt handling logic is implemented in signal handlers. Signal handlers invoke appropriate interrupt handlers (registered in emulated interrupt table) depending on emulated interrupt flag state (that enables or disables interrupt handling).

System timer emulation is based on periodic signaling provided by Linux kernel.

The emulation layer allows students to manage emulated hardware without constraints and without risk of crashing underlain operating system. At the same time, it does not hide device implementation. On the contrary, it ensures that every important implementation detail is visible.

COLIBROS emulation layer can be used on any Linux kernel based platform distribution, but install scripts are designed for and tested on Debian and some Debian-like distributions.

5. Overview of COLIBROS Interface

COLIBROS concurrency primitives are designed having in mind rich heritage of process synchronization papers [16, 25, 19, 1, 23]. COLIBROS also offers tools to synchronize threads and interrupt handlers in manner similar to ones used to synchronize threads. Besides that, COLIBROS offers flexible interface to determine order of continuation of thread activities, stopped for synchronization reasons. COLIBROS interface is intentionally designed to bare similarity to Java programming language [9] to prepare students for concurrent programming in Java.

5.1. COLIBROS Multithreading

Thread creation is conducted in three steps:

- Definition of thread class,
- Instantiation of thread object and
- Activation of thread.

Each user class that inherits COLIBROS class *Thread* represents thread.

Program 1: Thread class definition

```
class Thread {
    ...
    virtual void run(void) = 0;
    ...
    void* operator new(size_t type_size,
                      size_t stack_size = STACK_SIZE);
    void start(const int priority = DEFAULT_PRIORITY,
              const unsigned alias = 0);
    friend void destroy();
};
```

Such user class must implement member function *run()*, that contains thread body (see Program 1).

Thread objects are instantiated by C++ operator *new()*. Every thread object is therefore instance of previously defined user class.

Calling member function *start()*, inherited from class *Thread*, activates thread.

When thread activity is completed, thread objects are automatically destroyed. The friend function *destroy()* allows thread to stop its activity and terminate its existence.

The example of complete COLIBROS program shown in Program 2 illustrates definition, creation and activation of thread that prints string "Hello world".

Program 2: Example of Complete COLIBROS Program

```
class Example : public Thread
{
    public:
        void run();
};

void
Example::run()
{
    tout << "Hello World!";
}
```

```
void
Initial::run()
{
    Thread* t;
    t = new Example();
    t -> start();
}
```

The example of complete COLIBROS program is presented as Program 2. It contains member function *run()* of COLIBROS predefined class *Initial*. *Initial* class represents initial thread, which is automatically created and started at the beginning of the program execution. The object *tout* is predefined object for the program output (similar to standard C++ object *cout*).

5.2. COLIBROS Thread Synchronization Using Exclusive Variables

Cooperation between threads is achieved by using exclusive variables. Exclusive variable is instance of an exclusive class - a user defined class that inherits COLIBROS class *Exclusive*.

Program 3: Exclusive class definition

```
class Exclusive
    class Exclusive_block {
        ...
        Exclusive_block(Exclusive* ex);
        ~Exclusive_block();
    };
    class Condition {
        ...
        void await(unsigned t = 0);
        void signal();
        bool first(unsigned* t = 0);
        bool last();
        bool next(unsigned* t = 0);
        bool attach_tag(unsigned t);
    };
};
```

Mutual exclusion is implemented in *Exclusive_block* class (see Program 3). Exclusion enter protocol is implemented in constructor and exclusion exit protocol is implemented in destructor. Creation and destruction of an object of this class begins and ends exclusive regions.

Conditional synchronization is implemented through *Condition* class. Thread might need more than one condition to be fulfilled before it can

resume its activity in exclusive region. Because of that, COLIBROS exclusive class can contain several members of *Condition* type.

Member function *await()* is intended to stop thread activity until some condition is fulfilled, and function member *signal()* is intended to continue thread activity after some condition is fulfilled. It is possible to influence order in which threads continue their execution after necessary conditions have been fulfilled. This is accomplished by linking threads (their descriptors) into waiting lists. When linked in list, each thread can be assigned value (tag), which can be used to determine order in the list. Thread lists manipulation is supported by operations: *first()*, *next()*, *last()* and *attach_tag()* of class *Condition*.

The first operation (*first()*) positions internal pointer at first thread element in list, the second one (*next()*) enables sequential moving throughout the list, the third (*last()*) positions internal pointer at last thread element in list and the fourth (*attach_tag()*) enables changing of value (that determines a thread position in the list) associated with pointed thread element.

The Semaphore Example

Semaphores are not directly supported in COLIBROS but they can be easily implemented as in Example of Semaphore Definition (Program 4).

Program 4: Example of Semaphore Definition

```
class Semaphore : public Exclusive
{
    int state;
    Condition open;
public:
    Semaphore(int value = 1) : state(value) {};
    void wait();
    void resume();
};

void
Semaphore::wait()
{
    Exclusive_block set_up(this);
    if(state-- <= 0)
        open.await();
}

void
Semaphore::resume()
{
    Exclusive_block set_up(this);
    state++;
}
```

Žarko Živanov, Predrag Rakić, and Miroslav Hajduković

```
    open.signal();  
}
```

Semaphore definition (shown as Program 4) illustrates how creation and destruction of local variable *set_up* makes exclusive region. Also, it shows how conditional synchronization is achieved by *await()* and *signal()* operations on object *open* of *Condition* class.

Program 5 illustrates how *Semaphore* class can be used to achieve mutual exclusion.

Program 5: Example of Critical Region Protection Using Semaphore

```
Semaphore mutex;  
mutex.wait()  
    ... // mutually exclusive region  
mutex.resume()
```

The Delta List Example

As already mentioned, users can influence order in which threads continue their execution after necessary conditions have been fulfilled. For example delta list implementation (shown as Program 6) requires threads to be sorted in relative order of their expected awakening. Position of each thread in delta list depends of its sleeping time. Sleeping time of each thread is relative to its predecessor sleeping time (except the first thread with absolute sleeping time). Object *delta* is used by threads to access delta list.

Program 6: Example of Delta List Implementation

```
class Delta : public Exclusive  
{  
    Condition delta;  
public:  
    void sleep(unsigned time_to_wait);  
    void tick();  
};  
  
void  
Delta::sleep(unsigned time_to_wait)  
{  
    unsigned time;  
    Exclusive_block set_up(this);  
    if(delta.first(&time)) {  
        do {  
            if(time_to_wait >= time)  
                time_to_wait -= time;  
            else {
```

```

        time -= time_to_wait;
        delta.attach_tag(time);
        break;
    }
    } while(delta.next(&time));
}
delta.await(time_to_wait);
}

void
Delta::tick()
{
    unsigned time_to_wait;
    Exclusive_block set_up(this);
    if(delta.first(&time_to_wait)) {
        if(--time_to_wait)
            delta.attach_tag(time_to_wait);
        else {
            do {
                delta.signal();
            } while( (delta .first(&time_to_wait))
                    && (time_to_wait == 0) );
        }
    }
}

```

Operation *sleep()* links calling thread at appropriate place. This place is determined during sequential advance through delta list from its beginning (*delta.first()*) towards its end (*delta.next()*). Before thread is linked in delta list (*delta.await()*), relative sleeping time of its successor is changed (*delta.attach_tag()*).

Operation *tick()* decrements first thread's sleeping time (*delta.attach_tag()*). When first thread's sleeping time reaches zero, operation *tick()* wakes up this thread (*delta.signal()*), as well as its successors with zero relative sleeping time.

5.3. COLIBROS Device Drivers

Device drivers usually consist of two parts: synchronous and asynchronous. Synchronous part is used by computer system to ask device for some kind of service. Asynchronous part is used by device to notify computer system about (usually urgent) event [8].

In traditional operating system to implement device driver one must accomplish two unrelated operations: 1) register system service (for synchronous part) and 2) register interrupt handler (for asynchronous part).

In COLIBROS these two operations are conducted through definition and instantiation of atomic variable. Atomic variable is instance of atomic class – a

user defined class that inherits COLIBROS template class *Atomic* (Program 7).

Program 7: Atomic class definition

```
template<int VECTOR_NUMBER>
class Atomic {
    ...
    class Event {
        ...
        void expect(void);
        void notify(void);
    };
    class Atomic_block {
        ...
        Atomic_block();
        ~Atomic_block();
    };
    void start_interrupt_handling();
};
```

Atomic class must implement *interrupt_handler()* member function (inherited from *Atomic_base* class). This function is registered in interrupt table under interrupt number given as template argument for class *Atomic*.

Event class encapsulates mechanism that enables threads to wait for external events (*expect()*) and to resume activity after that events arrival (*notify()*).

Consistency of atomic variables is preserved using *Atomic_block* class. Constructor of this class disables interrupt handling, while destructor is returning it in previous state.

The example of timer device driver (shown as Program 8) contains atomic class that supports registration of time (*current_ticks*) and sleeping/wakeup (*countdown*). A single thread can postpone its activity by calling member function *sleep()*, providing sleep duration time. Creation and destruction of local variable *set_up*, in member function *sleep()*, begins and ends atomic region.

Program 8: Example of COLIBROS Timer Device Driver

```
class Timer : public Atomic<TIMER>
{
    unsigned long current_ticks;
    unsigned long countdown;
    Event alarm;
public:
    Timer(void);
    void sleep(unsigned duration);
    unsigned time_get() { return current_ticks; };
};
```

```

    protected:
        void interrupt_handler();
};

Timer::Timer()
{
    current_ticks = 0;
    countdown = 0;
    start_interrupt_handling();
}

void
Timer::sleep(unsigned duration)
{
    Atomic_block set_up;
    if(duration > 0) {
        countdown = duration;
        alarm.expect();
    }
}

void
Timer::interrupt_handler()
{
    current_ticks++;
    if((countdown > 0) && (--countdown) == 0) {
        alarm.notify();
    }
}

```

6. Conclusion and Future Work

For the last decade COLIBROS (and its predecessors) is continuously used as educational tool, providing generations of students with their first impressions of operating system details. COLIBROS joins concepts of classical and object oriented operating systems with multithreading concepts of high-level programming languages through simple and clean interface. Its implementation helps undergraduate students establish first contact with problems of concurrency, parallelism and operating systems.

Using COLIBROS in classroom has other advantages for students, like: better understanding of semantics and limitations of C++ language, introduction to Linux system calls and basic knowledge of hardware emulation (virtual machines).

In future, COLIBROS is going to be ported on bare hardware and used as small and fast object oriented, real-time (probably wireless sensor network) operating system.

References

1. Andrews, G. R., and Schneider, F. B. Concepts and notations for concurrent programming. *ACM Comput. Surv.* 15, 1 (1983), 3–43.
2. Atkin, B., and Sirer, E. G. Portos: an educational operating system for the post-pc environment. *SIGCSE Bull.* 34, 1 (2002), 116–120.
3. Ben-Ari, M. Principles of concurrent programming. Prentice-Hall, 1982.
4. Bynum, B., and Camp, T. After you, alfonse: a mutual exclusion toolkit. In *SIGCSE '96: Proceedings of the twenty-seventh SIGCSE technical symposium on Computer science education* (New York, NY, USA), ACM Press, pp. 170–174.
5. Campbell, R., Johnston, G., and Russo, V. Choices (class hierarchical open interface for custom embedded systems). *SIGOPS Oper. Syst. Rev.* 21, 3 (1987), 9–17.
6. Campbell, R. H., Islam, N., Raila, D., and Madany, P. Designing and implementing choices: an object-oriented system in c++. *Commun. ACM* 36, 9 (1993), 117–126.
7. Comer, D., and Fossum, T. V. Operating system design. Vol. 1: the XINU approach (PC edition). Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.
8. Corbet, J., Rubini, A., and Kroah-Hartman, G. *Linux Device Drivers*, 3rd edition ed. O'Reilly, 2005.
9. Eckel, B. *Thinking in Java*, 3rd edition ed. Prentice Hall Professional Technical Reference, 2002.
10. Engler, D. R. The exokernel operating system architecture. PhD thesis, Massachusetts Institute of Technology, 1998. Supervisor-M. Frans Kaashoek.
11. Engler, D. R., Kaashoek, M. F., and J. O'Toole, J. Exokernel: an operating system architecture for application-level resource management. In *SOSP '95: Proceedings of the fifteenth ACM symposium on Operating systems principles* (New York, NY, USA, 1995), ACM Press, pp. 251–266.
12. Gary, J. E. Using nachos is an upper division operating systems course. *J. Comput. Small Coll.* 18, 2 (2002), 337–345.
13. Goldweber, M., Davoli, R., and Morsiani, M. The kaya os project and the micromps hardware emulator. *SIGCSE Bull.* 37, 3 (2005), 49–53.
14. Hajdukovic, M. Konkurentno programiranje programskim jezikom conCert. Author's reprint, Novi Sad, 1996.
15. Hajdukovic, M. Operativni sistemi (problemi i struktura), tehnicke nauke - udzbenici ed. No. 74. Fakultet tehnickih nauka, 2004.
16. Hansen, P. B. The architecture of concurrent programs. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1977.
17. Hartley, S. J. *Concurrent programming: the Java programming language*. Oxford University Press, Inc., New York, NY, USA, 1998.
18. Kifer, M., and Smolka, S. *Introduction to Operating System Design and Implementation: The OSP 2 Approach*. Springer, 2007.
19. Lampson, B. W., and Redell, D. D. Experience with processes and monitors in mesa. *Commun. ACM* 23, 2 (1980), 105–117.
20. Lane, M. G., and k. Ghosal, A. Mpx-pc: an operating system project for the pc. *SIGCSE Bull.* 21, 1 (1989), 231–235.

21. Rakic, P. Migration of concurrent library COLIBRY from MS/DOS to GNU/Linux platform. Master's thesis, Faculty of Technical Science, Trg Dositeja obradovica 6, Novi Sad, Serbia, Feb 2006.
22. Reek, M. M. An undergraduate operating systems lab course. In SIGCSE '90: Proceedings of the twenty-first SIGCSE technical symposium on Computer science education (New York, NY, USA, 1990), ACM Press, pp. 171–175.
23. Reynolds, C. W. Signalling regions: multiprocessing in a shared memory reconsidered. *Softw. Pract. Exper.* 20, 4 (1990), 325–356.
24. Tanenbaum, A. S. Operating systems: design and implementation. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1987.
25. Wirth, N. MODULA: A Programming Language for Modular Multiprogramming, vol. 7. Eidgenössische Technische Hochschule, Institut für Informatik, 1977.

Žarko Živanov received Bachelor and Master of Science degree in Electrical and Computer Engineering from University of Novi Sad, Faculty of Technical Sciences. Hi is currently employed as assistant at University of Novi Sad, Computing and Automation Department.

Predrag Rakić received Bachelor and Master of Science degree in Electrical and Computer Engineering from University of Novi Sad, Faculty of Technical Sciences. Hi is currently employed as assistant at University of Novi Sad, Computing and Automation Department. His primary interests are: Unix-like systems, Parallel and GPU programming.

Miroslav Hajduković is currently employed as full professor at University of Novi Sad, Computing and Automation Department. He is teaching courses in Computer Architecture and Operating Systems.

Received: May 21, 2007; Accepted: October 12, 2009.

An Approach to Project Planning Employing Software and Systems Engineering Meta-Model Represented by an Ontology

Miroslav Líška¹ and Pavol Navrat¹

¹Faculty of Informatics and Information Technologies,
Slovak University of Technology,
Ilkovičova 3, 842 16 Bratislava, Slovakia
liska@semantickyweb.sk, navrat@fiit.stuba.sk

Abstract. Currently, one can witness a growing mutual influence between the Model Driven Architecture (MDA) and the Semantic Web. MDA is an approach that uses models for system development, but its architecture limits usability of these models for knowledge empowered solutions. A lot of research tackles applicability of MDA standards in the technical space of the Semantic Web. In this paper, we present an approach aimed at facilitating the use of Software and Systems Engineering Meta-Model (SPEM) for improvements that are rooted in knowledge engineering approaches. We show how SPEM can be used in the Semantic Web technical space. We describe how following our approach a project plan can be generated and verified. Finally, we present an example of project planning that uses ontology of a software requirements activity.

Keywords: Software and Systems Process Engineering Meta-Model, Web Ontology Language, Model Driven Architecture, Semantic Web.

1. Introduction

Software project management is the art of balancing competing objectives, managing risk, and overcoming constraints to deliver a product that meets the needs of the customers and the end users [1]. Project management is accomplished through the use of processes such as: initiating, planning, executing, controlling and closing [2]. Despite the fact that at present there exist many software developments process frameworks (e.g. Rational Unified Process, Eclipse Process Framework), the fact that relatively few projects are completely successful is an indicator of the difficulty of the task. One of the problems is that standard software development process frameworks are usually used as a navigable websites that contain only human-readable descriptions with supporting materials as documents templates etc. Thus, these kinds of frameworks cannot be used to represent machine interpretable content [3]. Moreover, these process frameworks are used in the technical spaces [4] that have model based architecture, such as MDA or Eclipse

Modeling Framework (EMF) [5]. These kinds of technical spaces also limit knowledge based processing, owing to their weakly defined semantics [6]. However, at present the emerging field of Semantic Web technologies promises new stimulus for Software Engineering research [7].

The Semantic Web is a vision for the future of the Web, in which information is given explicit meaning, making it easier for machines to automatically process and integrate information available on the Web [8]. The today's key Semantic Web technology is Web Ontology Language (OWL). OWL is intended to be used when the information contained in documents needs to be processed by applications, as opposed to situations where the content only needs to be presented to humans [9]. Aforementioned problems in software engineering and facts about the Semantic Web implies an opportunity to support project management with OWL, and thus to support project management with knowledge based techniques. In this work we address such opportunity and propose a method for project plan generation and verification that makes use of an ontology. To achieve it we need to move project planning in to the Semantic Web technical space. According to these requirements we attempt to make use of the potential of combining OWL and Software and Systems Engineering Meta-Model (SPEM).

1.1. Related works

SPEM is MDA standard used to define software and systems development processes and their components [10]. A SPEM process can be systematically mapped to a project plan by instantiating the different process' breakdown structure views. Therefore a SPEM model can represent a knowledge base that can be used for verification, whether a project plan conforms to this knowledge. However, the SPEM metamodel has the semiformal architecture, thus it is not possible to make and to verify created SPEM language statements with formal techniques such as the consistency or satisfiability verification [11]. But if we transform SPEM to the Semantic Web technical space, we can use the mentioned formal techniques due to facilities of OWL. Because SPEM is based on MDA, we can utilize the research results of transforming other MDA's standards to the Semantic Web technical space.

SPEM is specified in the Meta Object Facility (MOF) language that is the key language of MDA. MOF is a language for metamodel specification and it is used for specification of all model-based MDA standards [12]. It provides metadata management framework, and a set of metadata services to enable the development and interoperability of model and metadata driven systems [13]. On the Semantic Web side, OWL is intended to provide a language that can be used to describe the classes and relations between them that are inherent in Web documents and applications. OWL is based on Resource Description Framework Schema (RDFS) [14]. Both MOF and RDFS provide language elements, which can be used for metamodeling. Although they have similar language concepts such as `mof:ModelElement` with `rdf:Resource`, or `mof:Class` with `rdf:Class`, the languages are not equivalent. RDFS, as a

schema layer language, has a non-standard and non-fixed-layer metamodeling architecture, which makes some elements in model to have dual roles in the RDFS specification [15]. MOF is also used for specification of the Unified Modeling Language (UML) that is a language for specification, realization and documentation of software systems [16]. Even if UML and RDFS are similar in a domain of system specification, they are also substantially different. One issue that has been addressed was the problem that RDF properties are first class entities and they are not defined relative to a class. Therefore a given property cannot be defined to have a particular range when applied to objects of one class and another range when applied to objects of a different class [17]. Note this difference have also been propagated between OWL and UML [18]. It should be noted that efforts to transfer explicit knowledge into machine processable form encompass a much wider spectrum of works, e.g. [19, 20]. Still others attempt to develop domain specific languages, incorporating knowledge on the domain, that would be adaptable [21] improving in such a way the process of software evolution [22]. At present the main bridge that connects the Semantic Web with MDA is stated in the Ontology Definition Meta-Model (ODM) [23]. ODM defines the OWL Meta-Model specified in MOF (MOF – OWL mapping) and also the UML Profile for Ontology modeling (UML – OWL mapping). This architecture can be extended with additional mappings between the UML Profile for OWL and other UML Profiles for custom domains [24, 25]. We have already utilized this principle in our previous works where we created an approach to SPEM model validation with ontology [26] and ontology driven approach to software project enactment with a supplier [27]. However, our works are not the only one that concern with using of SPEM in the Semantic Web technical space. In short, the following subsection references to the three other related works.

The first work proposes to represent SPEM in DL [28]. The work creates mapping from MOF to DL and mapping from OCL (OMG, 2006b) constraints of SPEM to DL. The reason for the former mapping is to represent the SPEM MOF based metamodel with DL and the latter is to represent additional OCL constraints that supplement the SPEM metamodel with additional semantics. The second work presents a competency framework for software process understanding [29]. The motive is to create assessments for a correct understanding of a process that can be used in a software development company. The paper introduces creation of SPEM software process ontology for the SCRUM [30] software process with EPF Composer. However, only the third work is the most closest to our approach, since it proposes a project plan verification with ontology. The work intends to use SPEM process constraint definitions with the semantic rules with SWRL [31]. Note that SWRL is W3C Semantic Web Rule Language that combines OWL and RuleML [32].

1.2. Aims and objectives

We aimed in our research to devise a method that allows generation and verification of a project plan with the SPEM Ontology that use OWL-DL reasoning. Besides that, we present an example, where the subject of project planning is a requirements specification activity.

The rest of the paper is structured as follows. Section 2 presents our solution to the problem. First we define a conformance level with the SPEM compliance point and in short we discuss the SPEM transformation to the Semantic Web that we have already created and published. Then we describe the main principles of our approach to project plan generation and verification with ontology. Subsequently, Section 3 presents an example of ontology driven project plan verification with a requirements specification knowledge. Finally, Section 4 provides conclusion and future research direction.

2. An Approach to Ontology based SPEM

The key objective of project planning is to allocate tasks and responsibilities to a team of people over time and to monitor and manage progress relative to project plan [1]. To support this objective with knowledge based techniques, we need to have knowledge about software methods that could be used for project planning. We need to store this knowledge in form that is suitable for knowledge based processing. It ought to be possible to use this knowledge for a project plan generation. Following these requirements, we selected MDA language SPEM since is aimed for software process specification and is capable for process enactment with planning tools such as Microsoft Project by providing the means to map Processes to project plan [10]. Forasmuch as SPEM is not supportive to knowledge based techniques, we propose a way of transforming it into technical space of the Semantic Web.

2.1. Setting SPEM compliance point

SPEM metamodel is MOF-based and reuses UML 2 Infrastructure Library [33]. Its own extended elements are structured into seven main meta-model packages. Above these packages SPEM defines three compliance points (CP) which are: the SPEM Complete CP, SPEM Process with Behavior and Content CP and the SPEM Method Content CP. The scope of our solution is covered with Compliance Point "SPEM Process with Behavior and Content". The reason of this compliance point is because we need to work with separated reusable core method content from its application in processes, because a software method content can be used with arbitrary software process, such as iterative, agile etc. This separation is represented with two SPEM metamodel packages that are the Method Content and the Process with Method metamodel packages. The former provides concepts for SPEM

users and organizations to build up a development knowledge base that is independent of any specific processes and development projects. These concepts are the core elements of every method such as Roles, Tasks, and Work Product Definitions etc. The latter necessary metamodel package defines the structured work definitions that need to be performed to develop a system, e.g., by performing a project that follows the process. Such structured work definitions delineate the work to be performed along a timeline or lifecycle and organize it in so called breakdown structures. The most important elements of the Process with Method metamodel package are the Method Content Use elements. These elements are the key concept for realizing the separation of processes from method content. A Method Content Use can be characterized as a reference object for one particular Method Content Element, which has its own relationships and properties. When a Method Content Use is created, it shall be provided with congruent copies of the relationships defined for the referenced content element..

2.2. Moving SPEM into the Semantic Web

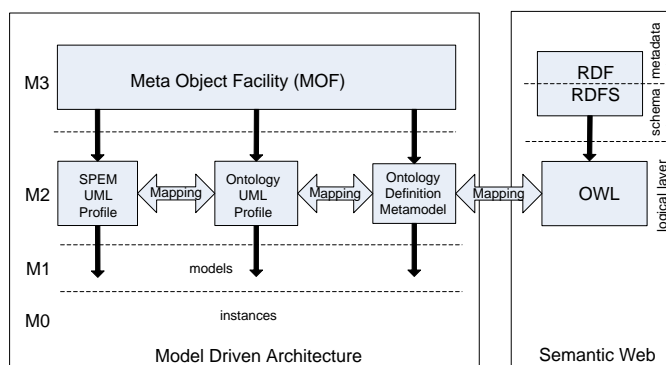


Fig. 1. Mappings between SPEM and OWL

Thus only a mapping between SPEM and OWL has to be created. Since the hallmark work [6] proposes the transformation of a MDA standard to the Semantic Web technical space with a mapping between UML Ontology Profile and an arbitrary UML Profile, we have either used this principle, thus we have created a mapping between the Ontology UML Profile and the SPEM UML Profile as it is shown in Figure 1.

However, the mapping between the Ontology UML Profile and the SPEM UML Profile were not sufficient to create the SPEM Ontology. The main problem was that the SPEM UML Profile does not contain SPEM semantics, and in addition for example, it was not possible to derive a domain and range of a relationship, etc. Therefore we had decided to create semiautomatic transformation that is based on merged SPEM metamodel to the SPEM UML Profile, where the result is the SPEM OWL DL Ontology. For more detailed

and comprehensive description about the SPEM transformation to the Semantic Web technical space and its utilizations, a reader may refer to [36, 37]. We have used OWL-DL, because this dialect of OWL retains computational completeness (all conclusions are guaranteed to be computable) and decidability (all computations will finish in finite time) [8]. To be conformed to this dialect, we have to adhere that an individual cannot be either a class, what is not violated in MDA technical space because of its 4 meta-layer architecture. For example, an analyst “Slávko Líška” is an instance of a Software Analyst SPEM class that is an instance of the Role Definition SPEM metaclass at the same time, thus the Software Analyst class is an individual and also a class. To avoid this problem in the Semantic Web technical space we have stated that a method content owl class is subclass of a SPEM owl class, and concrete individual is its instance. For example, the individual “Slávko Líška” is the instance of the Software Analyst owl class that is subclass of the Role Definition owl class from the SPEM Ontology. For the sake of clarity, follow Figure 2 illustrates the mapping in more detail.

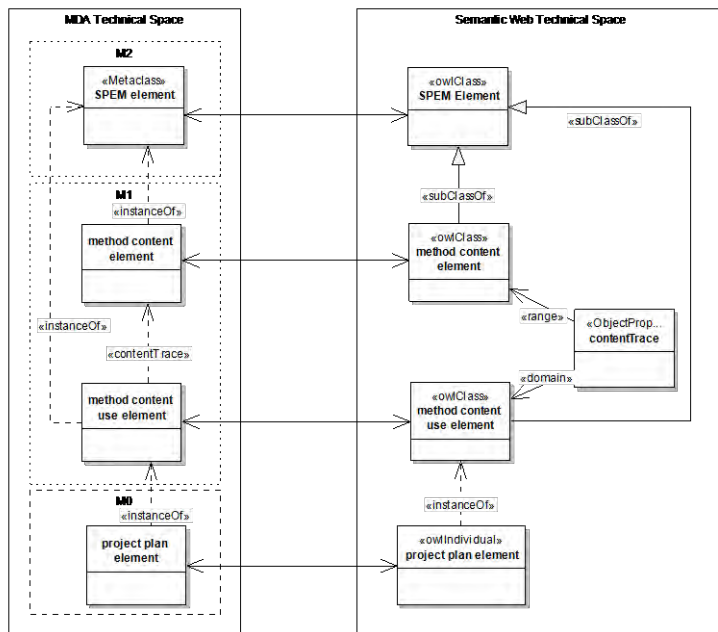


Fig. 2. SPEM in the Semantic Web Technical space

2.3. Project plan generation and verification

As we have already mentioned the SPEM Method Content is intended for a software method specification from the static point of view, whereas the SPEM Process provides concepts for representing method content elements in a process. Therefore once we have created the SPEM Ontology conformed to the SPEM Process with Behavior and Content Compliance Point, we can create ontology for a software method and process. To do so, we have created XSL transformations *SPEMMethodContent2OWL* and *SPEMProcess2OWL*. The former transforms a SPEM method content model to a SPEM method ontology and the latter transforms a SPEM process model to a SPEM process ontology.

At this point we can use OWL DL consistency verification between the SPEM Ontology, a SPEM method ontology and a SPEM process ontology. The first reason is to verify, whether a SPEM method ontology and a SPEM process ontology are correctly specified with the SPEM Ontology, for example whether a Task Definition element is performed only with a Role Definition element or whether a Role Use element is responsible just for a Work Product Use element. The second reason of the OWL DL verification at this point can be to ensure whether a SPEM process ontology is correctly traced to a SPEM method ontology, for example whether a Role Use element traces just a Role Definition element.

Since the scope of SPEM is purposely limited to the minimal elements necessary to define any software and systems development process, the SPEM metamodel does not include elements such as the Iteration, Phase etc. The reason is because not every software development process needs to have iterations for example. Therefore we had to extend the engine for a project plan validation either with the SPEM Base Plugin that is included in the SPEM specification. It provides commonly used concepts for the domain of software engineering such as the Phase, Iteration, Checklist etc.

However, none of mentioned ontologies does represent concrete project plan. To include it to the OWL DL verification we simply use the enactment between SPEM and a project planning system as it SPEM defines. The enactment is based on instantiation relationships between SPEM process elements and project plan elements. Thus we have created additional XSL transformation *MPP2OWL* that transforms a project plan to the individuals of a SPEM process ontology. The complete engine for project planning support with the SPEM ontology we propose is depicted in Figure 3.

When the result of the OWL DL verification is the inconsistency its source should be removed. In our case the inconsistency should be removed from the project plan. Finally, since a project plan can be mapped to a SPEM process we have created an additional XSL transformation *SPEMProcess2MPP*. The transformation generates a project plan from a SPEM process ontology. So, as it is shown in Figure 3 the OWL DL reasoning is based on the consistency verification between the SPEM Ontology, SPEM Base Plugin Ontology, a SPEM method ontology, a SPEM process ontology

and SPEM process individuals obtained from a project plan. Such solution provides two major utilization scenarios:

- Scenario 1. **Project plan generation with ontology.** When a project manager want to create a project plan, he can create a SPEM method and process models first and then use OWL DL consistency reasoning to ensure that they are consistent. Then he can just simply transform his SPEM process model to the SPEM process ontology.

- Scenario 2. **Project plan verification with ontology.** This scenario is essential when a project manager wants to ensure that his already created project plan is consistent with desired method content and process. However, this second scenario usually follows upon the first. A project manager obviously makes many changes to his project plan; therefore it is necessary to ensure that these changes do not break the required consistency.

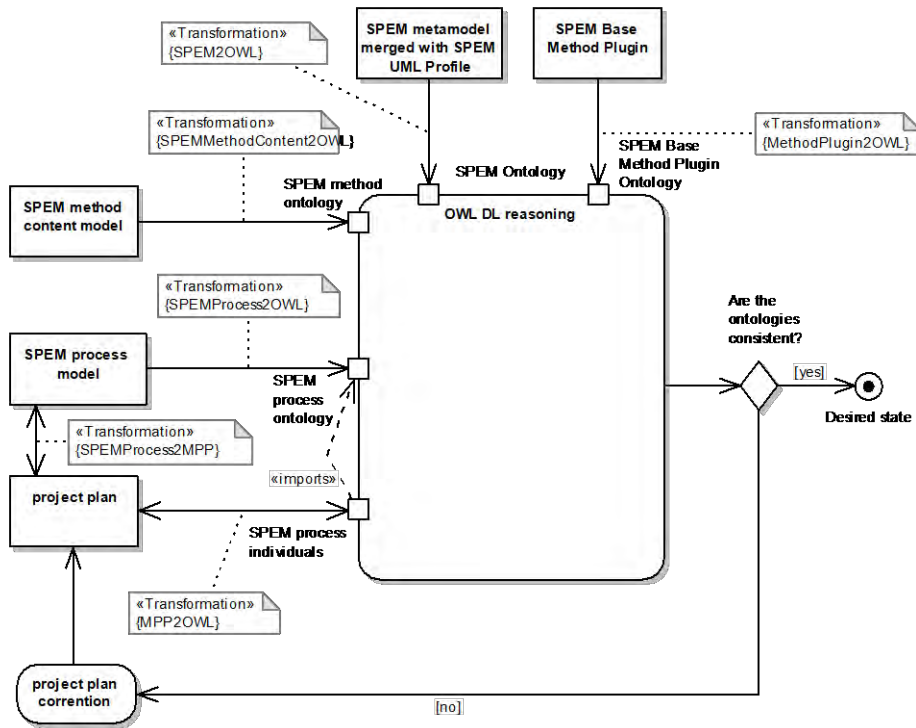


Fig. 3. An approach of project planning with SPEM ontology

To be more precise, we give the formally defined conditions that cover the mentioned utilization scenarios. Since the first scenario is included in the second, we focus only on the Scenario 2. First we define the Project Plan Knowledge as the union of the SPEM Ontology, SPEM Base Plugin Ontology, a SPEM method ontology and a SPEM process ontology, as it is shown in Formula 1.

$$\text{Project Planning Knowledge} = \text{SPEM Ontology} \sqcup \text{SPEM Base Plugin Ontology} \sqcup \text{SPEM method content ontology} \sqcup \text{SPEM process ontology} . \quad (1)$$

Then we say, that the Project Planning Knowledge is satisfied in a project plan if it is true that

$$\text{Project Plan} \models \text{Project Planning Knowledge} . \quad (2)$$

From the First Order Logic point of view, the Project Plan Knowledge is the theory and a project plan is its model. Since a theory can have a model only if a theory is consistent [38], it is necessary, that the Formula 3 is either true

$$\text{SPEM Ontology} \vdash \text{SPEM Base Plugin Ontology} \vdash \text{SPEM method content ontology} \vdash \text{SPEM process ontology} . \quad (3)$$

2.4. Implementation

Ontologies rely on well-defined and semantically powerful concepts in artificial intelligence [39], such as description logics, reasoning, and rule-based systems [40]. Since we use OWL DL form of ontology, the implementation has goal to present the proposed utilization scenarios with a Knowledge Representation System that supports description logics. Developing a knowledge base using a description logic language means setting up a terminology (the vocabulary of the application domain) in a part of the knowledge base called the TBox, and assertions about named individuals (using the vocabulary from the TBox) in a part of the knowledge base called the ABox [41]. In other words, the ABox describes a specific state of affairs in the world in terms of the concepts and roles defined in the TBox [6].

Table 1. Mapping between components of a knowledge based representation system to our approach's ontologies

Ontology type	KBRS component
SPEM Ontology	TBox
SPEM Base Method Plugin	TBox
SPEM method content ontology	TBox
SPEM process ontology	TBox
SPEM method plugin ontology	TBox
Individuals of a SPEM process ontology	ABox

As we have it discussed in Subsection 2.2, our approach is conformed to the OWL DL dialect that disallows to an individual to be either a class. Therefore all classes of the ontologies used in our approach constitute TBOX, whereas only individuals obtained from a project plan create ABOX as it is depicted in Table 1.

3. Example of project plan verification with ontology

For the sake of clarity we present an example of a project plan verification with the SPEM ontology. The example shows the OWL DL consistency reasoning of simple Create Requirements Method Content and Create Requirements Process with the project plan that also contains simple plan for requirements specification. The reasoning fully supports the engine presented in Figure 3, therefore the SPEM Ontology, SPEM Base Plugin Ontology, Create Requirements Method Ontology, Create Requirements Process Ontology and the ontology of the project plan are the input ontologies to the reasoning process. Intentionally, the first result of the reasoning is inconsistency, thus the project planning knowledge is not satisfied in the project plan. However, when the origin of the inconsistency is removed, we get desired project plan that is consistent with the project planning knowledge.

Figure 4 presents an excerpt of the SPEM Ontology. The asserted axiom depicted in the figure represents the key concept of SPEM that is the separation of a method content from process.

<i>SPEM Ontology</i>
<u>owlClasses:</u> BreakDownElement, MethodContentElement, WorkDefinition...
<u>objectProperties:</u> performs, mandatoryOutput, optionalOutput, responsible ...
<u>asserted axioms:</u> MethodContentElement \sqcap MethodContentUse $\equiv \emptyset$...

Fig. 4. An excerpt of the SPEM Ontology

For the purpose of this article we do not show an excerpt of the SPEM Base Plugin Ontology, because it just defines additional concepts such as the Iteration, Process etc. which are the specialized classes from the SPEM Ontology classes. Instead, we focus on the Create Requirements Method Content Model that is depicted in Figure 5 in an instant.

The method content model defines role definitions, work product definitions, task definitions and their appropriate relationships which are needed to create software requirements. As it is shown in Figure 2 we have created the XSL transformation MethodContent2OWL that transforms a XML serialization of a method content model to the XML format of the OWL DL ontology. In our case, an excerpt of the resulted Create Requirements Ontology is depicted in Figure 6.

An Approach to Project Planning Employing Software and Systems Engineering Meta-Model Represented by an Ontology

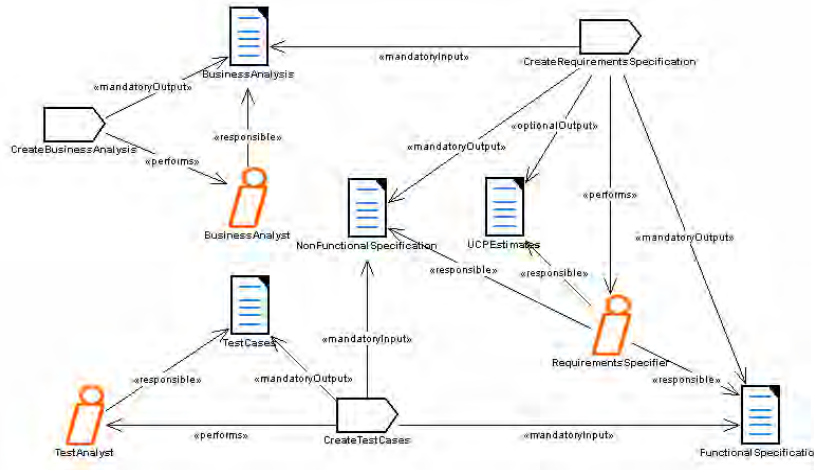


Fig. 5. Create Requirements Method Content Model

```

Create Requirements Method Ontology
imports:
  SPEMontology
owlClasses:
  BusinessAnalyst, RequirementsSpecifier, TestAnalyst ...
asserted axioms:
  RequirementsSpecifier  $\sqsubseteq$  RoleDefinition  $\sqcap$ 
   $\forall$  responsible.(FunctionalSpecification  $\sqcup$  UCPEstimates  $\sqcup$  ...
    
```

Fig. 6. Create Requirements Method Content Model Ontology

```

Create Requirements Process Ontology
imports:
  CreateRequirementsMethodOntology, SPEMBasePluginOntology
owlClasses:
  myProcess, Iteration_I1, CreateRequirements_I1 .....
asserted axioms:
  myProcess  $\sqsubseteq$  Process,
  Iteration_I1  $\sqsubseteq$  Iteration  $\sqcap$  nestedBreakdown.myProcess,
  CreateRequirements_I1  $\sqsubseteq$  nestedBreakdown.Iteration_I1 ...
    
```

Fig. 7. Create Requirements Process Ontology

The Create Requirements Ontology imports the SPEM Ontology for the purpose of SPEM classes specialization. For example, the Requirements Specifier class is specialization from the Role Definition SPEM class. However, as it can be seen in Figure 5 or Figure 6, the method content ontology does not represent any dynamics aspect of the create requirements software method. Thus, we have to additionally define even a process. An excerpt of the process is depicted in Figure 7. The excerpt defines that the

Process “myProcess” consists of the Iteration “Iteration_I1” that consists of the Activity “Create Requirements I1”.

Since we have presented the SPEM Ontology, Create Requirements Method Ontology and Create Requirements Process Ontology, we can define the Create Requirements Knowledge, as it is depicted in Formula 4.

$$\text{Create Requirements Knowledge} = \text{SPEM Ontology} \sqcup \text{SPEM Base Plugin Ontology} \sqcup \text{Create Requirements Method Ontology} \sqcup \text{Create Requirements Process Ontology} . \quad (4)$$

At this point, all we need for the verification process is a project plan that also defines how to create requirements. The plan is depicted in Figure 8.

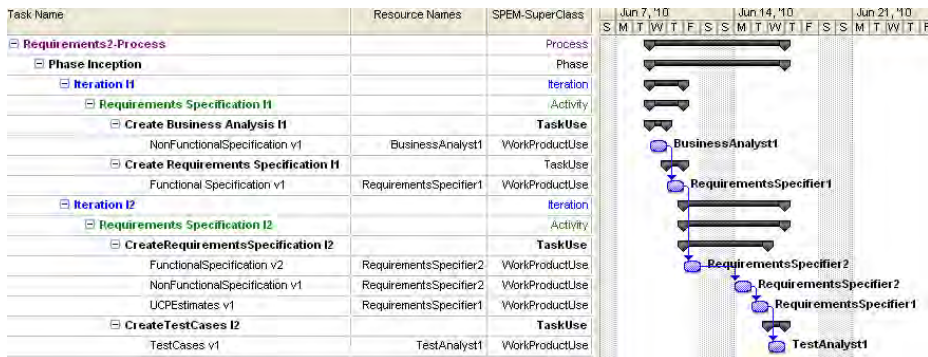


Fig. 8. Project Plan 1X

The project plan is created with the MS Project Plan that allows saving the plan into the XML format. Thus we can use our XML transformation MPP2OWL that transforms a project plan to the individuals of the Create Requirements Process Ontology. An excerpt of the resulted project plan ontology Project Plan 1X Ontology is depicted in Figure 9. Note that the depicted instantiation represents the enactment of SPEM with a project planning system.

```

Project Plan 1X Ontology
imports:
  CreateRequirementsProcessOntology
individuals:
  Requirements2-Process, PhaseInception ...
instantiation:
  Process("myProcess"),
  BusinessAnalyst_I1("BusinessAnalyst1"),
  BusinessAnalyst_I1("RequirementsSpecifier1") ...
object property assertions:
  responsible("BusinessAnalyst1", "NonFunctionalSpecification_I1") ...
    
```

Fig. 9. An excerpt of the Project Plan 1X Ontology

So, when we execute the OWL verification at this point, we find out that the result is inconsistency. This means that the project plan is not properly defined with regard to the Create Requirements Knowledge, or by the more precise words, that the project planning knowledge is not satisfied in the project plan. It is true that

ProjectPlan1X \neq Create Requirements Knowledge . (5)

The source of inconsistency lies in Figure 7, where it is intentionally stated that the Role Definition "Business Analyst 1" is responsible for the Work Product Definition "NonFunctional Specification". The assertion is inconsistent with the Create Requirements Method Ontology, where it is stated, that the Role Definition "Requirements Specifier" is responsible for the work product definition. Thus, after this inconsistency is removed, it is true that

ProjectPlan1 \neq Create Requirements Knowledge . (6)

4. Conclusion

We presented our approach to project plan generation and verification with the SPEM Ontology. When we compare our approach with the most similar work [29] we conclude that we created not only wider method specification, but we have also presented its implementation. Moreover, we presented the engine for project plan verification with ontology that supports key property of SPEM that is the Method Content separation from a Process and either the separation from the SPEM Base Plugin. Additionally, since a Method Plugin consists of a Method Content and a Process, our approach can be easily extended with any Method Plugin, for example, with the Rational Unified Process Plugin. However, we have to admit that our research must continue in this topic, in order to succeed in real commercial projects. It is very difficult to imagine that for a purpose of project plan verification a project manager will use a knowledge based framework directly, without appropriate user interfaces. Therefore, we have started implementation of a macro for the MS Project that will remotely access OWL API for OWL-DL reasoning purposes and it will print verification results back into MS Project Plan. Additionally, like the most similar work does, we have to include either SWRL to our approach to extend the expressiveness of description logic with the rule based expressions. All these mentioned deficiencies of our solution are the objectives of our future development.

Acknowledgments. "This work was supported by the Scientific Grant Agency of Slovakia, grant No. VG1/0508/09, and it is a partial result of the Research & Development Operational Program for the project Support of Center of Excellence for Smart Technologies, Systems and Services II, ITMS 25240120029, co-funded by ERDF."

References

1. Kruchten, P.: The Rational Unified Process: An Introduction. (3rd edition). Addison-Wesley, USA. (2003)
2. Project Management Institute: A Guide to the Project Management Body of Knowledge (PMBOK– 4th edition). Project Management Institute, USA. (2008)
3. Fujita, H., Zualkernan, I. A. (ed.): An Ontology-Driven Approach for Generating Assessments for the Scrum Software Process. In Proceedings of the seventh SoMeT_08. IOS Press, The Netherlands, 190-205. (2008)
4. Kurtev, I., Béziuin, J., Aksit, M.: Technological spaces: An initial appraisal. In Proceedings of the Confederal International Conferences, CoopIS, DOA, and ODBASE, Industrial Track, Irvine, CA, USA, (2002)
5. Steinberg, D., Budinsky, F., Paternostro, M., Merks, E.: EMF: Eclipse Modeling Framework (2nd Edition). Addison-Wesley Longman, Amsterdam, (2009)
6. Gašević, D., Djurić, D., Devedžić, V.: Model Driven Engineering and Ontology Development, 2nd ed., Springer, Berlin, (2009)
7. Happel, H.J., Seedorf, S.: Applications of ontologies in software engineering. In: International Workshop on Semantic Web Enabled Software Engineering (SWESE'06), Athens, USA. (2006)
8. McGuinness, D. L., Harmelen, F.: OWL Web Ontology Language Overview, W3C Recommendation, (2004). [Online]. Available: <http://www.w3.org/TR/owl-features/> (current November 2009)
9. Smith, M.K., Welty, Ch., McGuinness, D.L.: OWL Web Ontology Language Guide, W3C Recommendation, (2004). [Online]. Available: <http://www.w3.org/TR/owl-guide/> (current November 2009)
10. Object Management Group: Software and Systems Process Engineering Meta-Model 2.0, formal/2008-04-01. Object Management Group, USA, (2008). [Online]. Available: <http://www.omg.org/technology/documents/formal/spem.htm> (current November 2009)
11. Krdžavac, N., Gašević, D., Devedžić, V.: Model Driven Engineering of a Tableau Algorithm for Description Logics. Computer Science and Information Systems, Vol. 6, No. 1, (2009)
12. Frankel, D.S.: Model Driven Architecture. Applying MDA to Enterprise Computing. Willey, USA. (2003)
13. Object Management Group: Meta Object Facility (MOF) 2.0 Core Specification, formal/2006-01-01. Object Management Group, USA, (2008). [Online]. Available: <http://www.omg.org/spec/MOF/2.0/> (current November 2009)
14. Brickley, D., Guha, R. V., McBride, B.: RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, (2004). [Online]. Available: <http://www.w3.org/TR/rdf-schema/> (current November 2009)
15. Pan, J., Horrocks, I.: Metamodeling Architecture of Web Ontology Languages, In Proceedings of the First Semantic Web Working Symposium, Stanford, 131-149. (2001)
16. Object Management Group: UML 2.2 Superstructure Specification, formal/09-02-03. Object Management Group, USA, (2009). [Online]. Available: <http://www.omg.org/technology/documents/formal/uml.htm> (current November 2009)
17. Cranefield, S.: Networked Knowledge Representation and Exchange using UML and RDF. Journal of Digital Information, Volume 1 Issue 8, (2001)
18. Hart, L., Emery, P., Colomb, B., Raymond, K., Taraporewalla, S., Chang, D., Ye, Y., Kendall, E., Dutra, M.: OWL Full and UML 2.0 Compared. OMG TFC Report, (2004)

An Approach to Project Planning Employing Software and Systems Engineering Meta-Model Represented by an Ontology

19. Polášek, I., Kelemen, J.: Ontologies in Knowledge Office Systems. In: KEOD 2009, 1st International Conference on Knowledge Engineering and Ontology Development, Funchal - Madeira, Portugal. INSTICC PRESS, 400-413. (2009)
20. Polášek, I., Chudá, D., Kristová, G.: Modelling System Dynamics in a Newer Version of UML (in Slovak). In: Proc. Systémová integrácia 2006. Žilina University, Žilina, 311-317. (2006)
21. Hrnčíč, D., Mernik, M., Forgáč, M., Kollár, J.: Evolution and Adaptation of Domain Specific Languages. In: Proc. of the Tenth International Conference on Informatics 2009, Technical University of Kosice, Kosice, 154-159. (2009)
22. Kollár, J., Porubán, J., Václavík, P., Bandáková, J., Forgáč, M.: Adaptive Language Approach to Software Systems Evolution. In: Proc. International Multiconference on Computer Science and Information Technology: 1st Workshop on Advances in Programming Languages (WAPL'07), Polish Information Processing Society, 1081-1091. (2007)
23. Object Management Group: Ontology Definition Meta-Model 1.0. formal/2009-05-01. Object Management Group, USA, (2009). [Online]. Available: <http://www.omg.org/spec/ODM/1.0/> (current November 2009)
24. Gašević, D., Djurić, D., Devedžić, V.: MDA and Ontology Development. Springer, Berlin, Heidelberg. (2006)
25. Gašević, D., Djurić, D., Devedžić, V.: Bridging MDA and OWL Ontologies. Journal of Web Engineering, Vol. 4, no. 2, pp. 119-134. (2005)
26. Liška, M.: An Approach of Ontology Oriented SPEM Models Validation. In Proceedings of the First International Workshop on Future Trends of Model-Driven Development (FTMDD) in the context of the 11th International Conference on Enterprise Information Systems. Milan, Italy, 40-43. (2009)
27. Liška, M., Navrat, P.: An Ontology Based Approach to Software Project Enactment with a Supplier. In 14th East-European Conference on Advances in Databases and Information Systems (ADBIS2010), Lecture Notes in Computer Science 6295, Novi Sad, Serbia, Springer, pp. 378-391. (2010)
28. Wang, S., Jin, L. J. CH. Represent Software Process Engineering Metamodel in Description Logic. In Proceedings of World Academy of Science, Engineering and Technology, vol. 11. (2006)
29. Zualkernan, I. A. An Ontology-Driven Approach for Generating Assessments for the SCRUM Process. New Trends in Software Methodologies, Tools and Techniques. IOS Press. (2008)
30. Schwaber, K., Beedle, M. Agile Software Development with SCRUM. Prentice Hall. (2002)
31. Rodríguez, D., Sicilia, M., A.: Defining SPEM 2 Process Constraints with Semantic Rules Using SWRL. In Proceedings of the Third International Workshop on Ontology, Conceptualization and Epistemology for Information Systems, Software Engineering and Service Science held in conjunction with CAiSE'09 Conference. Amsterdam, The Netherlands, pp. 95-104. (2009)
32. Horrocks, I., Patel-Schneider, P., F., Boley, H., Tabet, T., Grosz, B., Dean, M.: SWRL: A Semantic Web Rule Language, Combining OWL and RuleML. W3C Member Submission, (2004). [Online]. Available: <http://www.w3.org/Submission/SWRL/> (current November 2009)
33. Object Management Group: UML 2.2 Infrastructure Specification, formal/2009-02-04. Object Management Group, USA, (2009). [Online]. Available: <http://www.omg.org/spec/UML/2.2/> (current November 2009)
34. Object Management Group: MOF 2.0 / XMI Mapping Specification, v2.1.1, formal/2007-12-01. Object Management Group, USA, (2007). [Online]. Available: <http://www.omg.org/spec/XMI/2.1.1/> (current November 2009)

35. Djurić, D.: MDA-based ontology infrastructure, *Computer Science and Information Systems*, Vol. 1, no. 1, pp. 91–116. (2006)
36. Líška, M.: *Extending and Utilizing the Software and Systems Process Engineering Metamodel with Ontology*. PhD Thesis, ID: FIIT-3094-4984. Slovak University of Technology in Bratislava. (2010)
37. Líška, M. *Extending and Utilizing the Software and Systems Process Engineering Metamodel with Ontology*. *Information Sciences and Technologies, Bulletin of the ACM Slovakia*, Vol. 2, No. 2, pp. 13-20 (2010)
38. Kvasnička, V., Pospíchal, J.: *Mathematical logic*. Slovak University of Technology in Bratislava. STU Press. (2005)
39. Navrat, P. et al.: *Artificial Intelligence, 2002*, Slovak University of Technology in Bratislava. STU Press. (2002)
40. Vianu, V. *Rule-based languages*. *Annals of Mathematics and Artificial Intelligence*, vol. 19, no. 1–2, pp. 215–259. (1997)
41. Baader, F., Horrocks, I., Saatler, U.: *Description Logics*. In Steffen Staab and Rudi Studer, editors, *Handbook on Ontologies*, *International Handbooks on Information Systems*, Springer. 3-28. (2004)

Miroslav Líška received the M.S. degree in informatics from the Technical University in Košice, Slovakia in 2002, and the PhD. degree in software and information systems from the Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava in 2010. His interests include semantic web, ontologies, software process engineering and semantic enterprise oriented architectures. He currently works as a requirements analyst and software methodologist in Datalan, in Bratislava, Slovakia.

Pavol Navrat received his Ing. (Master) cum laude in 1975, and his PhD. degree in computing machinery in 1984 both from Slovak University of Technology in Bratislava. He is currently a professor of Informatics at the Slovak University of Technology and serves as the director of the Institute of Informatics and Software Engineering. During his career, he was also with other universities abroad. His research interests include related areas from software engineering, artificial intelligence, and information systems. He published numerous research articles, several books and co-edited and co-authored several monographs. Prof. Navrat is a Fellow of the IET and a Senior Member of the IEEE and its Computer Society. He is a Senior Member of the ACM and chair of the ACM Slovakia Chapter. He is also a member of the Association for Advancement of Artificial Intelligence, Slovak Society for Computer Science and Slovak Artificial Intelligence Society. He serves on the Technical Committee 12 Artificial Intelligence of IFIP as the representative of Slovakia.

Received: November 10, 2009; Accepted: October 19, 2010.

Facilitating information system development with Panoramic view on data

Dejan Lavbič¹, Iztok Lajovic², and Marjan Krisper¹

¹ University of Ljubljana, Faculty of Computer and Information Science, Slovenia
{Dejan.Lavbic, Marjan.Krisper}@fri.uni-lj.si

² KreS, Kreativni sistemi, Slovenia
KreS@siol.net

Abstract. The increasing amount of information and the absence of an effective tool for assisting users with minimal technical knowledge lead us to use associative thinking paradigm for implementation of a software solution – Panorama. In this study, we present object recognition process, based on context + focus information visualization techniques, as a foundation for realization of Panorama. We show that user can easily define data vocabulary of selected domain that is furthermore used as the application framework. The purpose of Panorama approach is to facilitate software development of certain problem domains by shortening the Software Development Life Cycle with minimizing the impact of implementation, review and maintenance phase. Our approach is focused on using and updating data vocabulary by users without extensive programming skills. Panorama therefore facilitates traversing through data by following associations where user does not need to be familiar with the query language, the data structure and does not need to know the problem domain fully. Our approach has been verified by detailed comparison to existing approaches and in an experiment by implementing selected use cases. The results confirmed that Panorama fits problem domains with emphasis on data oriented rather than ones with process oriented aspects. In such cases the development of selected problem domains is shortened up to 25%, where emphasis is mainly on analysis, logical design and testing, while omitting physical design and programming, which is performed automatically by Panorama tool.

Keywords: software development, associative thinking, object recognition, rapid application development.

1. Introduction

There is a vast amount of data available on a daily basis from various sources that we have to perceive, analyse and comprehend. Several approaches to organising data on computers exist, starting from straightforward support for simple data structures that majority of operating systems offer, to sophisticated database management systems for storage of composed and

complex data structures [1, 2]. Every approach deals with management of data in its own distinct way, where user is able to enter and edit the data as well as prepare and view reports. The key question is how a user without extensive IT knowledge can successfully cope with the complexity of all perceived data.

The development of Computer Science was always focused on aiding human actors at performing tasks [3-6]. In this paper, Panorama as an implementation of our approach to support human actors in information management and software development will be discussed. Panorama is based on principles of associative thinking and it supports inclusion of various data structures that compound our information space and facilitates traversing through data by following associations. This concept is based on the human observation of the environment and its context – the complete information about selected object is available at certain time with a possibility of further exploration of related concepts through associations. The present paper presents Panorama approach with object recognition process, based on context + focus information visualization techniques [7], that starts with observation and perception of the environment and continues with context of observed object. This paper also addresses how software development is facilitated with Panorama tool. There a user can develop, use and maintain an information system that would suit his/her needs for information management without an extensive technical knowledge. The advantage of Panorama is in its efficient coding of information with the principle of searching for information introduced in our solution being very similar to human information processing within memory system. Panorama's main goal is to facilitate the process of transferring concepts from long-term memory to short-term memory by associations [7-9] among concepts that user can freely traverse. The emphasis is given to intuitive user interface [10] that enables user with minimal technical skills to start browsing and building the information space without extensive previous training. The ideas of Memex [11-13], associations, context + focus techniques and particularly frames had a profound influence on our notion of how Panorama was designed and implemented. Panorama follows those concepts and presents an approach in presenting data based on principles of associative thinking. In employing Panorama approach for information system development several good practices from existing object-oriented, rapid development and people-oriented methodologies were considered, while considering low level of IT expertise required for development.

The remainder of this paper is structured as follows. In section 2, an overview of our approach to implementation of information systems is given. The following section 3 includes detailed comparison of Panorama approach and similar approaches in different themes in information systems development. An experiment is also presented that compares Panorama approach to object-oriented approach on selected use cases. This is followed by the final section that presents the conclusions and plans for future work.

2. Panorama approach

In the following section first architecture of Panorama will be introduced with elements of data vocabulary and traversing the information space. Then some aspects of Panorama, as input and reporting module, will be discussed with other elements of panoramic view on data. This section will also present information system development with Panorama approach. After short comparison to existing approaches detail steps will be depicted and their impact on the process.

The distinct value of Panorama approach is in supporting tool Panorama that is used throughout the process – in development and in using the implemented system. When developing information system Panorama facilitates definition of data vocabulary and enables automatic transformation of logical model to implementation model and therefore minimizes the required technical knowledge of users. By enabling this transformation in Panorama approach users don't deal with programming phase, but rather put more attention to analysis and design. After successful implementation of information system, Panorama serves as intuitive user interface with employing of design patterns [14] and enabling traversing the information space by following associations and focus + context visualization techniques that improve user experience.

2.1. Data vocabulary and traversing the information space

Panorama has a two-layer architecture. The top layer is the data vocabulary containing all information about concepts being used. Data vocabulary serves for creating underlying tables with all fields corresponding to data vocabulary definition including fields that bind tables with links. Data vocabulary characterizes objects of classes and describes possible interconnections of objects from different classes. Based on the content of data vocabulary Panorama “interprets” individual attributes and presents appropriate visualization to the user.

Panorama's advantage is in **bidirectional nature of links** in opposite to unidirectional links in documents for example found on the internet. Panorama is using network data structure with all input and output links indexed thus providing fast and elegant data retrieval. User has the possibility to compare level of authority, i.e. number of incoming links, just by clicking neighbour records of the same class and after choosing one of them he can enable filter on selected record to broaden record which are shown in subsequent search.

When using Panorama user deals with 5-dimensional space as depicted in Fig. 1. It follows a visual information-seeking mantra for designers [15]: overview first, zoom and filter, then details-on-demand. User interface supports traversing through this multi-dimensional space and is furthermore divided into two parts (Fig. 2): (i) on the left hand side are **selection windows**

for setting restrictions and (ii) on the right hand side is the **viewing window** for displaying the observed object and its context.

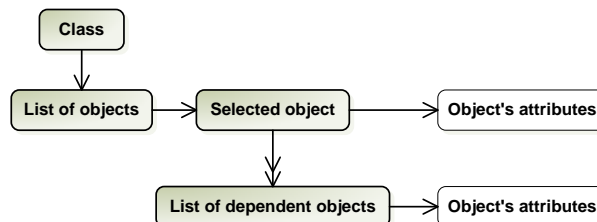


Fig. 1. Panorama's 5 dimensional space

Selection windows contain a list of **classes**, **objects** and **filters**. With the selection of a class (1st dimension), list of objects (2nd dimension) is simultaneously updated.

The class interconnection is the way of changing point of view of data. Record attributes that belong to class interconnection data type are implemented as links between data records. All links in Panorama are bidirectional so the user can jump to records in both directions. Jump to record pointed to by selected link opens new possibilities for link selection and further data investigation. There is no need to go back in the tree of selected choices to change the path of investigation as is the case in tree data structure.

In Fig. 2 an example of opera works, developed by one of the world famous conductors using Panorama tools is presented. When one selects class Opera Works, the list of operas Don Giovanni, Fidelio, Figaro's wedding and others is displayed. Furthermore, by selecting the object (Madame Butterfly in our example), all the related data within its context is displayed in **viewing window**. This first of all includes object's attributes (3rd dimension) and the list of dependent objects (4th dimension) with its attributes (5th dimension) respectively. Object's attribute in our examples is Opera Work Madam Butterfly while dependant objects are objects from various classes – Roles (Cio-Cio-San, F. B. Pinkerton etc.), Small Roles (Bonze, Cio-Cio-San's mother etc.), Silent Roles (Dolore, Cio-Cio-San's child), Choir (Female Choir), Orchestra Cast and Scenic Music. All record links are displayed underlined.

Traversing from one information node to another is feasible simply by a mouse click on a selected concept that consequently becomes the new object of observation. As aforesaid one of the key elements of Panorama are bidirectional links. They enable unlimited traversal across information space. The observed object can also be marked as anchor which denotes that, while traversing through related links is limited only to selected object.

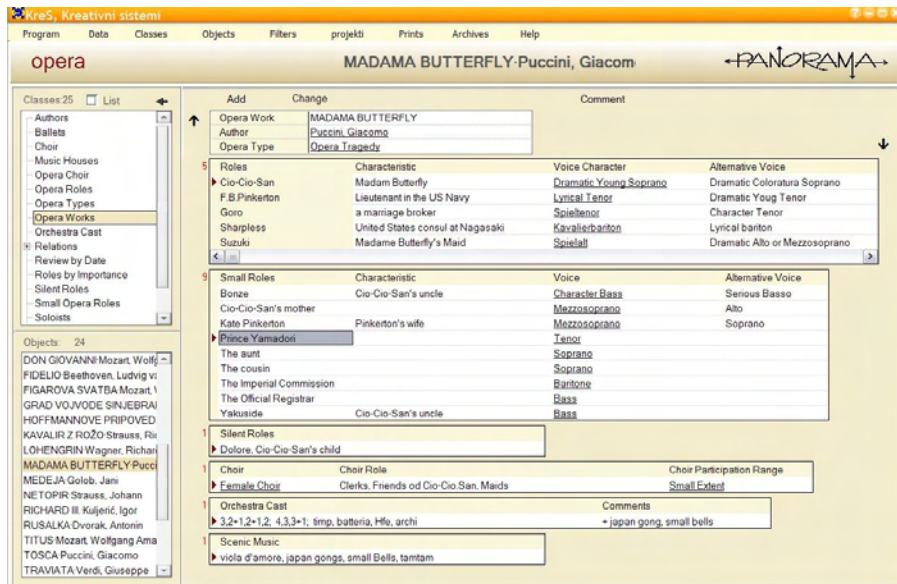


Fig. 2. Panorama's user interface

The starting point for all mechanisms in Panorama that facilitate traversing from one object to another is in the design of specially structured database that enables to establish n-way relationships between objects and support object recognition process. Objects within the context of observation are stored in the database and grouped into classes of objects. All objects, instances of selected class, share the same set of properties, but differ in values of attributes. Among multi-value attributes, linking attributes can be found and they represent a connection between object of observation and its neighbourhood. Database used in Panorama is based on concepts of relational databases (RDB) [16, 17] with functionalities of object oriented databases (OODB) [18, 19]. Access to data in OODB can be faster because joins are often not needed (as in a tabular implementation of a RDB). This is because an object can be retrieved directly by following pointers and without a search. At the ground level navigational databases approach [20] is followed which incorporates both the network model and hierarchical model of database interfaces. Navigational techniques use pointers and paths to navigate among data records. This is in contrast to the relational model (implemented in RDB), which strives to use declarative or logic programming techniques where system is being queried for result instead of being navigated to.

Panorama enables a simple definition of relationships between objects, without restrictions in quantity. Objects in Panorama can be linked in the following ways: (i) **1:1** or **1:n** relationship by a pointer attribute that is a member of source object's set of attributes. Panorama uses this type of relationship for defining **links**. For example: compositions can have the same title (Symphony No. 1), but different authors (Ludwig van Beethoven,

Johannes Brahms etc.). When traversing through information space, Panorama offers a list of all known authors of selected compositions and on the other hand all compositions of each author are displayed regarding to the context. (ii) **m:n** relationship can be established by an intermediate class or an object and in its simplest form the intermediate object contains a two pointer attribute, forming the concatenated key of the object. (iii) **m:n:q** relationship (and subsequent cardinalities of relationships) can also be established by intermediate objects. The key is formed from concatenation of pointers and cardinality equals to number of relationships. Intermediate objects behave as normal objects and can be linked to subsequent relationships as basic objects can. They have their own identity and cannot be fully accessible through concatenated key. The multiple relationships with cardinality two or more represent the **information node**.

So far data vocabulary design of Panorama and structure of user interface was presented, but this is only a prerequisite for computer aided associative thinking process that is performed by the user.

2.2. Panoramic view on data

Besides controlled input of data and reporting module, Panorama also offers user friendly, easy to use, intuitive interface for data review and for traversing the information space. In ordinary applications all these functionalities have yet to be programmed.

Computer aided associative thinking process supported in Panorama is based on object recognition process (see Fig. 3) that starts with observation and perception of the environment. When observing the surrounding environment the focus is on the specific object of observation, nevertheless all the neighbouring objects are also comprehended. Therefore the context of the observed object is identified by the set of objects in the neighbourhood of that object. Classification of objects into classes is based on the equality of attributes' sets (and interval of values) that belong to respective objects from a set of objects.

The question that arises is what kind of mechanism to use for neighbourhood observation and how to recognize each object based upon their attributes? Obviously all information about classes, their attributes and possible values for each of the attributes is stored somewhere in our brains. This leads to the conclusion that we need a very powerful data vocabulary, progressive by quantity of stored information and by search methods and data classification. When a new object is noticed, visible attributes of the object are perceived and we try to find a match in our data vocabulary. The pattern of recognized attributes defines the most probable class that the object belongs to. In some cases further analysis of other attributes is needed to confirm the correct class membership or to classify object into different class.

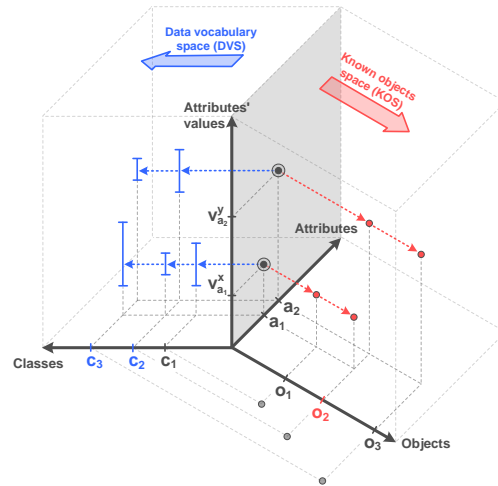


Fig. 3. Object recognition process in information space

In Panorama object recognition is based on information residing in two information spaces: **data vocabulary space (DVS)** and **known object space (KOS)** (see Fig. 3). The techniques used for recognition of new objects and traversing through information space are based on several techniques in information visualization [7], especially overview + detail and focus + context approaches.

The simplicity of usage is one of the key elements of Panorama, because target users are users with very little technical knowledge of details in implementation. Querying complex data structures is based on intuitive graphical interface, whereas Panorama does not use query language to explore the information space, but it's rather based on drag-and-drop approach by selecting objects with a mouse. This leads to the following advantages: (i) **User doesn't need to be familiar with the query language such as SQL** (similar approach to Query-by-Example and Visual Query Builder methods). (ii) **There is no need for being familiar with the data structure**, where in navigational approach users not need to know the names of appropriate attributes to build a query, because a user only follows associations. (iii) For traversing the information space, **user also does not need to know the problem domain fully**. A major disadvantage of using query languages is responding with 'no result found' message to user in case of a false condition. User is not interested in what cannot be found, but only wants to know what is available. With the use of Panorama and traversal that it supports it is impossible to make a wrong turn to an empty information space.

The problems introduced are solved with intuitive user interface – display of multiple concepts with filtered data that are in direct relationship with observed object. Traversing through information does not require entering commands, but only to follow the links that represent a connection with linked objects. The traversal is feasible in all directions and not limited by iteration. To facilitate

quick start usage of Panorama, several import options exist. Data can be imported for selected class from multiple sources, while Panorama will recognize the input structure and display it to the user who can take appropriate actions. To maintain high level of adaptability, filters and anchors are also available to help the user in coping with the large amount of information and the appropriate formatting and displaying at latter to the screen. Filter is used to simply restrict our traversal through information space, so we can focus only on desired concepts. Anchor can be set for multiple classes and it limits the displaying of objects of selected classes only. For integration with other applications various types (screen, printer, clipboard or file – Excel, DBF, SQL, HTML, XML, TXT, RTF, PDF) of exports are available. The common quality of creating reports is the analogy with traversing the information space. Report can simply be defined for selected object of observation and its context or list of objects with applied restrictions. Again, no technical knowledge is required in a form of entering query expressions, but simply by selecting appropriate restrictions from data vocabulary.

2.3. Panorama and IS development

The general approach to developing information systems is the Information Systems Development Life Cycle (SDLC) [21, 22]. Although there are many variants, it has the following basic structure: (1) feasibility study, (2) system investigation, (3) systems analysis, (4) systems design stage, (5) implementation and (6) review and maintenance. The use of methodology improves the practice of information systems development, but due to dynamic environment and constantly changing requirements it's hard to adopt only one approach with strictly defined phases, techniques and tools. Panorama with its novel approach to developing information systems follows examples mainly from **Object-oriented methodologies** [21, 23, 24], **Rapid development methodologies** [25-28], **People-oriented methodologies** [29-31] and **Model-Driven approach** [32, 33].

Panorama also includes some aspects of Method Engineering (ME). ME, sometimes referred to as methodology engineering, is the process of designing, constructing, and merging methods and techniques to support information systems development [34, 35]. ME is especially important in IS development as it is a sign of progress in the adaptation of methodologies to changing IS/IT environments and progress in methodology design itself. ME is often associated with the hierarchical and bureaucratic approaches of the 1980s as techniques were combined to form meta-methodologies. Normally it is concerned with the blending of methods and techniques into a methodology or framework. However, according to [22, 36], its most recent form is Enterprise resource planning (ERP) systems, which are combinations of application types rather than methods and techniques. There are also other known advances in ME such as The OPEN Process Framework (OPF) [37] and the ISO/IEC 24744 international standard [38].

The essential building blocks of information systems in object-oriented methodologies are captured by defining objects – both data and process encapsulated together. The resemblance with using Panorama for observation and perception of the environment can be found especially in Object-Oriented Analysis (OOA). In OOA five major activities exist: finding classes and objects, identifying structures, identifying subjects, defining attributes and defining services. The end result of finding classes and objects is a set of relevant classes, and for each class the associated objects modelled using the appropriate conventions. The second activity organizes basic classes and objects into hierarchies that will enable the benefits of inheritance. Afterwards, identifying subjects reduces the complexity of the model produced so far by dividing or grouping into more manageable and understandable subject areas. According to OOA, objects are composed of data and processing, where defining attributes defines the data and defining services defines the processing. The similar process is also conducted when building data vocabulary in Panorama. Implementation (step 5) and review and maintenance (step 6) in SDLC represent the majority of the time spent on project development [39]. To successfully complete projects that don't require a lot of calculation and don't emphasize process oriented aspects (e.g. office management, CRM or record management), Panorama offers a fast and efficient way. The first four phases (feasibility study, system integration, systems analysis and systems design) from SDLC are always present, because the development is not feasible without presence of requirements and designs. Panorama on the other hand tries to minimize the impact of phases 5 and 6 on the time required for completing the project. Our solution contains required mechanisms for updating and printing the content of data vocabulary and also controlled input of data with referential integrity checking. This leads to avoidance of the major part of the work that has to be done in implementation and review and maintenance phase. Furthermore the vast majority of complications are also avoided: (i) The lack of communication between designers and programmers leads to misunderstandings. (ii) Data model doesn't cover all the requirements. (iii) The final application is not fully tested and it contains several bugs. (iv) Subsequent changes to the application, based on the requirements defined after the completion of the project etc.

Guidelines similar to those of Panorama's can also be found in Rapid development methodologies, whereas Rapid Application Development (RAD) is a response to the need to develop information systems more quickly [22] because of rapidly changing business requirements. James Martin's RAD (JMRAD) and Web IS development methodology (WISDM) place emphasis on the early stages of systems development as is the case of using Panorama. This concerns the definition of requirements, analysis and design with the use of a language that is of the business and the users, rather than the more technical language of information systems. As mentioned before, common guidelines can also be found in People-oriented methodologies, particularly in KADS, CommonKADS and End-user computing (EUC). People-oriented methodologies encompass the socio-technical view that, in order to

be effective, the technology must fit closely with the social and organizational factors in the application domain. These methodologies are people oriented in the sense that they attempt to capture the expertise and knowledge of people in the organization. They emphasize the importance of knowledge management and leverage knowledge as an important organizational resource. Knowledge, the ability to use information in action for a particular purpose, is even more important than information, which is more commonplace. Panorama in some aspects follows End-user computing paradigm, found in people oriented themes, where application is developed by non-specialist IT people, particularly in the form of spread-sheets. Main advantages are an effective way of providing some functionality of the information systems in organizations and an increase in user satisfaction. As Method Engineering emphasizes usage of best practices from systems development, Panorama can also be analysed using this perspective, but it is limited to data oriented approaches. While Method Engineering approaches (i.e. OPF) mainly focus on the process (framework, libraries, reusable methods, usage guidelines), thou also mention tools, Panorama is rather focused on the tool support that follows proposed approach.

We can therefore conclude that Panorama follows a kind of knowledge-engineering methodology with iterative approach and following next steps: **(1) Determination of the domain, (2) Identification and grouping of information (project definition)**, (2.1) Assembling of important concepts, (2.2) Defining classes, (2.3) Defining attributes and relationships and **(3) Using the project.**

We start with defining the domain and scope (step 1) which coincides with step 1 from SDLC. This includes the determination of the problem domain, type of questions that the Panorama model should provide answers to, users responsible for using and maintaining data vocabulary and known objects space etc.

Step 2 deals with construction of data vocabulary that characterizes objects of classes and describes possible interconnections of objects from different classes. Based on the content of data vocabulary, Panorama interprets individual attributes and presents appropriate visualization to the user. Defining data vocabulary is furthermore consisted of three sub steps: assembling of important concepts, defining the classes and defining the properties.

In assembling of important concepts the main concepts are identified, what properties those concepts have and what would we like to say about them. It is important to get a comprehensive list of concepts without paying special attention to overlaps between concepts they represent, relationships among the concepts and properties.

The next action is to define the classes which are used to group objects with the same properties. When classes are described they alone don't provide enough information to answer the questions from step 1, therefore we must describe the internal structure of concepts.

The phase of defining classes has already selected classes from the list created in step 2.1. Most of the remaining concepts are likely to be attributes

of these classes, mandatory or optional. Besides these proprietary attributes, each object also consists of a set of additional attributes that position selected object in space and time. By defining relationships among concepts we define foundation for Panorama to enable traversing through information space. Actions that have been made so far have roughly covered steps 2 to 5 in SDLC and prepared all the necessary requirements to start using Panorama. By using Panorama we consider entering new data or defining known objects space (step 3). This requires selecting a class, creating individual instance of that class and filling in perceived attributes.

Panorama tool offers users complete support in using this approach. In initial steps users define data vocabulary with all the restrictions that apply. The tool supports building data vocabulary from scratch and also importing data from several resources. When data vocabulary is defined user can enter all the restrictions that apply and finally couple this schema with actual data again from scratch or from various data sources. At this stage the use of the project starts where Panorama deals with all translations from abstract concepts in data vocabulary to actual implementation at the information system level. The traversal through information space is supported by intuitive user interface which does not require users to learn query language.

3. Evaluation

In the evaluation part of this research we tackled the following research questions: (1) How does developing computer software with Panorama approach differs from existing, especially object-oriented approaches? (2) What are the problem domains where application of Panorama approach is the most feasible? (3) What is the impact of Panorama approach regarding time and technical knowledge of users for completing the software development life cycle?

First a **comparison** between Panorama approach, STRADIS, SSADM, IE, DSDM, OOA and RUP was conducted following framework for comparing methodologies defined in [22]. Based on the findings from that comparison an **experiment** was carried out with 4 experts and 24 students as raters, who rated 2 problem domains and 2 approaches. Raters heuristically evaluated the required a priori knowledge for each of the approaches and time required to complete individual phase was measured.

3.1. Comparison of approaches

Method

We are aware that Panorama approach is not a methodology, because a methodology encompasses much more than an approach. Nevertheless the framework mentioned before is still appropriate due to its very generic design. There are several questions that need to be answered to give a complete comparison: (i) What are the underlying philosophical assumptions of the methodology? What makes it a legitimate approach? (ii) What particular skills are required of the participants? (iii) What representation, abstractions, and models are employed? (iv) What aspects of the development process does the methodology cover? (v) What is the focus of the methodology? Is it, for example, people-, data-, process-, and/or problem-oriented? Does it address organizational and strategic issues?

Besides Panorama approach six other approaches from different themes in information systems development were considered for comparison as follows: (i) **Structured Analysis, Design and Implementation of Information Systems (STRADIS)** from process-oriented methodologies, (ii) **Structured Systems Analysis and Design Method (SSADM)** from blended methodologies, (iii) **Information Engineering (IE)** from blended methodologies, (iv) **Dynamic Systems Development Method (DSDM)** from rapid development methodologies, (v) **Object-Oriented Analysis (OOA)** from object-oriented methodologies and (vi) **Rational Unified Process (RUP)** from object-oriented methodologies. Each of the seven selected approaches was discussed and mutually compared according to elements of the framework for comparing methodologies.

Results

There are number of sub elements to **philosophy** which we will examine in turn. The first sub element is paradigm where science and systems paradigm are identified as to be of critical importance. In science paradigm complexity is handled through reductionism, breaking things down into smaller and smaller parts for examination and explanation. In systems paradigm concern is for the whole picture, the emergent properties (e.g. the whole is greater than the sum of parts), and the interrelationships between parts of the whole. All approaches (STRADIS, SSADM, IE, DSDM, OOA, RUP and Panorama) follow science paradigm with clear reductionist approach and they all accept the ontological position of realism. They all argue that universe comprises objectively given, immutable objects and structure, whereas these exist as empirical entities, on their own, independent of the observer's appreciation of them. One of fairly obvious clue to the methodology philosophy is the stated objectives. DSDM, although often resulting in the design of computer

systems, is sometimes used to address organizational or general problem-solving issues. STRADIS, SSADM, IE, OOA and RUP all claim that they are not general problem-solving methodologies, but have clear objectives to develop computerized information systems. Panorama is also not general problem-solving approach, but its objective is to combine very heterogeneous information into a stable and networked computerized information system. Very important aspect is to enable display of object and its context and enable two-way interactions with objects from the context. The third factor relating to philosophy is the domain of situations that methodologies address. IE is identified as being of the planning, organization, and strategy type. It is an approach adopting the philosophy that an organization needs a strategic plan in order to function effectively, and that success is related to the identification of information systems that will benefit the organization and help achieve its strategic objectives. STRADIS, SSADM, DSDM, OOA and RUP are classified as specific problem-solving methodologies; i.e. they do not focus on identifying systems required by the organization but begin by assuming that a specific problem is to be addressed. Panorama can like others also be classified as specific problem-solving methodologies with distinct top-down approach and identification of organization's focus. It is also important to notice that Panorama is not intended for process oriented domains. Most methodologies appear to claim to be general purpose. OOA and RUP are considered to be general purpose, although it is suggested that they are not very helpful for simple, limited systems or systems with only a few class-&-objects. STRADIS is also stated to be general purpose and applicable to any size of system, yet the main technique is data flow diagramming, which is not particularly suitable for all types of application, for example, the development of management information systems or web-based systems. SSADM and also STRADIS, IE, OOA and RUP have all been designed primarily for use in large organizations. Panorama is intended to help develop information systems in smaller environments or where the target system is PC-based. It is particularly suitable for the following types of application – office management, CRM, record management, planning, maintenance; and various user types, including management level.

The second element of the framework, the **model**, can be investigated in terms of the type of model, the levels of abstraction of the model, and the orientation or focus of the model. The primary process model used is the data flow diagram and it can be found in STRADIS, while in SSADM is an important model, although not the only one. It is also present in IE, but play a less significant role than, for example, in STRADIS. The data flow diagram is predominantly a process model, and data are only modeled as a by-product of the processes. In OOA and RUP the basic models are in integration of both process and data orientation, often in the same diagram, which is a key element of the object-oriented approach. Panorama in contrast to other approaches uses data-oriented modeling approach such as ER model whereas process orientation is not handled as an important aspect.

The third element of the framework is that of the **techniques and tools** that a methodology employs. STRADIS is an example of a methodology

which is largely described in terms of its techniques. A methodology which advocates the clear separation of the modeling of data and processes is SSADM. OOA and RUP use object-oriented techniques and models or models that incorporate the essential combination of data and process. Tools range from simple drawing tools through to tools supporting the whole development process, including prototyping, project management, code generation, simulation and so on. IE explicitly suggest that the techniques are not fundamental part of the methodology and that the current recommended techniques can be replaced. Panorama approach is mainly presented in a view of a tool that is used in throughout the development process by developers and users and as a final product by users. It supports the complete lifecycle of software development and is also deployed to a client and plays a vital role as a toolset which encompasses model, editor, reports, graphics screen interfaces, code and database generators, configuration tools etc.

Scope is an indication of the stages of the life cycle of stems development which the methodology covers (0 – it does not mention, 1 – briefly mentioned, 2 – address the area, 3 – covers the stage in some detail) and is presented in the following table.

Table 1. Scope comparison of different software development approaches

	STRADIS	SSADM	IE	DSDM	OOA	RUP	Panorama
Strategy	0	2	3	2	0	1	0
Feasibility	3	3	2	3	0	2	1
Analysis	3	3	3	3	3	3	3
Logical design	3	3	3	3	3	3	3
Physical design	3	3	3	3	2	3	3
Programming	2	0	2	2	0	2	3
Testing	2	0	3	2	0	3	2
Implementation	1	2	2	2	0	3	3
Evaluation	1	1	2	0	0	2	0
Maintenance	0	0	1	0	0	0	1

The main focus of most compared methodologies is at the analysis and design stages, while maintenance is by far the worst covered stage. Panorama is also very strong in analysis and design phase but it also addresses maintenance.

The next element in the framework evaluates the **deliverables** at each stage and, in particular, the nature of the final deliverable. This can vary from being an analysis specification to a working implementation of a system and all its related procedures. STRADIS, SSADM, IE, DSDM, OOA and RUP all include analysis and design specification, while STRADIS, SSADM, IE, DSDM and RUP also cover working implementation of a system. In Panorama approach the emphasis is on working implementation of a system while products at the end of phases include a working version of logical model and data vocabulary.

The background of the methodology broadly identifies its origins in terms of academic or commercial. STRADIS, SSADM, IE, DSSDM, OOA and Panorama lie in the commercial sphere, whereas RUP has academic backgrounds. There is a view that commercial methodologies are not in as widespread use as is claimed but finding evidence is difficult. According to a research [22] 57% of organizations were using a systems development methodology, but, of these, only 11% used a commercial development methodology, 30% used a commercial methodology adapted for in-house use, and 59% a methodology which was internally developed and not based on an a commercial methodology. Panorama approach has a wide user base among different levels in organizations. Because of its design it is interesting to point out that is especially suitable for less technically oriented users.

The traditional view of information systems development is that a specialist team of professional systems analysts and designers perform the analysis and design aspects and professional programmers design the programs and write the code. Although the exact roles have different names, in general systems development is undertaken by professional technical developers. This view is taken by STRADIS, SSADM, IE, DSDM, OOA and RUP. Panorama takes a different view, and users have a much more active role by following JAD (Joint Application Design) which includes domain experts and professional technical developers. Therefore in Panorama, the users are directly involved in the analysis and design with the professional analysts as consultants. It has turned out, as seen in the following experiment, that leaving out analysis of current state could have beneficial consequences. In almost all methodologies considerable training and experience is necessary for at least some of the players. This may significantly increase the time and costs required to develop a project. With Panorama time required for implementation is very short because majority of features are automatically built from data vocabulary.

The last element of the framework is what purchasers actually get for their money. This may consist of software, written documentation, an agreed number of hours' training, a telephone help service, and consultancy etc. With SSADM the product is large and copious sets of manuals. RUP has a range of documents, books, and specifications but also has a multimedia website, and indeed claims that the methodology product is actually delivered using this web technology. Panorama approach produces two types of products: (1) a tool that facilitates software development and is at later stage also used on a client side as implementation and (2) the project, based on JAD approach together with documentation.

Discussion

The following table summarizes the view about comparing selected methodologies. It can be argued that several differences exist among approaches, which is expected, because of their origin in different themes in information systems development.

Panorama stands out especially because of its data-oriented model and very strong orientation into a working implementation of a system with object and its context visualization approaches.

Table 2. Comparison of software development approaches (abbreviations used are further explained in the text, following the table).

	STRADIS	SSADM	IE	DSDM	OOA	RUP	Panorama
1. Philosophy							
a) Paradigm	Pa-1	Pa-1	Pa-1	Pa-1	Pa-1	Pa-1	Pa-1
b) Objectives	Ob-1	Ob-1	Ob-1	Ob-2	Ob-1	Ob-1	Ob-3
c) Domain	Do-1	Do-1	Do-2	Do-1	Do-1	Do-1	Do-3
d) Target	Ta-1	Ta-1	Ta-1	Ta-1	Ta-1	Ta-1	Ta-2
2. Model	Mo-1	Mo-2	Mo-2	Mo-3	Mo-4	Mo-4	Mo-5
3. Techniques and tools	TT-1	TT-2	TT-3	TT-4	TT-5	TT-6	TT-7
4. Scope	Sc-1	Sc-2	Sc-3	Sc-4	Sc-5	Sc-6	Sc-7
5. Outputs	Ou-1	Ou-1	Ou-1	Ou-1	Ou-1	Ou-1	Ou-2
6. Practice							
a) Background	Bg-1	Bg-1	Bg-1	Bg-1	Bg-1	Bg-2	Bg-1 Bg-2
b) User base	Us-1	Us-1	Us-1	Us-1	Us-1	Us-1	Us-2
c) Participants	Ba-1	Ba-1	Ba-1	Ba-1 Ba-2	Ba-1	Ba-1	Ba-1 Ba-2
7. Product	Pr-1	Pr-2	Pr-1	Pr-1	Pr-1	Pr-3	Pr-4

Abbreviations used in the table are as follows: **Paradigm:** (Pa-1) Science paradigm, clear reductionist approach, acceptance of the ontological position of realism. **Objectives:** (Ob-1) Not a general problem-solving methodology, but as having clear objectives to develop computerized information systems. (Ob-2) Often resulting in the design of computer systems, but is still sometimes used to address general problem-solving issues. (Ob-3) Not a general problem-solving approach, but to combine very heterogeneous information into networked IS. **Domain:** (Do-1) Specific problem-solving methodologies. Do not focus on identifying the systems required by the organization but begin by assuming that a specific problem is to be addressed. (Do-2) Planning, organization and strategy type, where organization needs a strategic plan in order to function effectively. (Do-3) Specific problem-solving approach. Not intended for process oriented domains. **Target:** (Ta-1) General purpose, primarily designed for use in large organizations, (Ta-2) Smaller environments, especially office management, CRM, record management, planning, project management, maintenance etc. **Model:** (Mo-1) Data flow diagram is the primary process model, (Mo-2) Data flow diagram is an important (although not only one) model, (Mo-3) Usually data flow diagram and data oriented model, (Mo-4) The basic model is an integration of both process and data orientation, often in the same diagram, which is a key element of the object-oriented approach. (Mo-5) Data-oriented model. **Techniques and tools:** (TT-1) Largely described in terms of techniques, (TT-2) Clear separation of modeling of data and processes.

Recommend the tools to some degree. (TT-3) Techniques are not fundamental part. (TT-4) No specific tools are recommended, (TT-5) OO techniques. Tools might be helpful but are not necessarily essential. (TT-6) OO techniques. Process should not be contemplated without the use of tools, the process being too complicated and time consuming. (TT-7) ER model as primary technique and mainly presented in a view of a tool Panorama. **Scope:** (Sc-1) Feasibility, analysis, logical and physical design are covered in detail. Programming and testing are addressed. Implementation and evaluation are only briefly mentioned. (Sc-2) Feasibility, analysis, logical and physical design are covered in detail. Strategy and implementation are addressed. Evaluation is only briefly mentioned. (Sc-3) Strategy, analysis, logical and physical design and testing are covered in detail. Feasibility, programming, implementation and evaluation are addressed. Maintenance is briefly mentioned. (Sc-4) Feasibility, analysis, logical and physical design are covered in detail. Strategy, programming, testing and implementation are addressed. (Sc-5) Analysis and logical design are covered in detail. Physical design is addressed. (Sc-6) Analysis, logical and physical design, testing, implementation and evaluation are covered in detail. Feasibility and programming are addressed. Strategy is only briefly mentioned. (Sc-7) Analysis, logical and physical design, programming, and implementation are covered in detail. Testing is addressed. Feasibility and maintenance are only briefly mentioned. **Outputs:** (Ou-1) Analysis and design specification and working implementation of a system and (Ou-2) Working version of logical model and data vocabulary and working implementation of a system. **Background:** (Bg-1) Commercial and (Bg-2) Academic. **User base:** (Us-1) Various and numerous and (Us-2) Small. **Participants:** (Ba-1) Professional technical developers and (Ba-2) Business users. **Product:** (Pr-1) Documentation, (Pr-2) Large and copious sets of manuals, (Pr-3) Range of documents, books, specifications and a multimedia website and (Pr-4) Tool, project implementation and documentation.

Based on these findings an experiment was conducted to measure the effectiveness of object-oriented (OO) and Panorama approach on two selected domains, as those approaches share the most commonalities and OO approach is one of the most widespread in information system development.

3.2. Experiment

Method

Altogether 28 members were chosen to conduct the experiment. 4 members (**P₁** to **P₄**) from University of Ljubljana, Faculty of Computer and Information Systems, Information Systems laboratory and 24 students (**S₁** to **S₂₄**) that attended undergraduate program at the same University. Evaluators **P₁** to **P₄**

were experts in software development and had extensive practical experience with object-oriented software development, while evaluators S_1 to S_{24} were users without extensive technical knowledge and inexperienced in software development. Panorama approach was presented to all evaluators before conducting the experiment, because none of them had any previous experience.

Before engaging the experiment all evaluators were randomly divided into two groups A and B of equal size and the same ratio among experts and students. This resulted in individual groups containing 2 expert members and 12 students. Each of the group had to evaluate both Object-oriented and Panorama approach, but in turn in different problem domain.

For evaluation purposes two fairly simple use cases C_1 and C_2 were identified. Use case C_1 is concerned about **information system support for cinemas**, whereas information about cinema halls, timetables, movies, genres and customer's preferred seats by each cinema hall has to be available. Several functionalities had to be implemented, including performing analyses by customer to see which movie genre are they interested in, when they go to the cinema etc. Use C_2 deals with **information support for mobile operators billing**. Support for bundled services, customers, packages, bill detail types, conversation rates and subscription fees had to be captured. Master-detail view on the expenses related to customer has to be supported to enable further analyses of rating information.

Each group of participants (A and B) was requested to implement use cases C_1 and C_2 using Object-oriented and Panorama approach. The requirements of cases C_1 and C_2 were described in detail with the natural language. Participants had a goal to produce a working computer program following given use case and accompanying documentation. In the last phase of Software Development lifecycle – Evaluation, all results from participants were independently double-blind evaluated by domain experts according to the original objectives and requirements. If these requests were not met, users were required to repair their solution and if afterwards the solution was still inadequate, the participant (P_1 to P_4 and S_1 to S_{24}) was excluded from the analysis. Due to this issue three participants were excluded from the set of raters which resulted in reduced number of participants from 27 to 24.

With Object-oriented approach PowerDesigner tool was used in analysis and design phase for user requirements analysis and logical design of the system. UML modeling notation was used to capture and define the problem domain (use case diagrams, class diagrams etc.). Eclipse development environment with a Microsoft SQL Server database was used in the programming phase. For Panorama approach Panorama tool was used throughout the software development cycle, following steps defined in section 2.3.

The experiment was then performed in two steps (see Table 3). First group A implemented case study C_1 using Object-oriented approach, while group B implemented case study C_1 using Panorama approach. In second step roles have been reversed and group B implemented C_2 using object-oriented approach and group A implemented C_2 using Panorama approach.

Table 3. Procedure of conducting experiment

	Object-oriented approach	Panorama approach
Step 1: Use case C₁	group A	group B
Step 2: Use case C₂	group B	group A

For each participant several elements were measured, from time required to complete each stage of software development lifecycle, final outputs and artifacts between the stages and at the end whether the requirements were met.

The phases of the Software Development approach were evaluated on a time consumption and output basis. In analysis users were required to get familiar with problem domain and no special outputs were requested from this phase. In logical design users had to capture the requirements in selected notation (UML use case and class diagrams for Object-oriented approach and business vocabulary in Panorama). In physical design transformation of logical into physical model is required when specific environment is selected. The outputs required from this phase include generation of database schema in Object-oriented approach. In programming phase users were requested to implement required functionalities and the output is a working prototype. The following phase of testing deals with implemented prototype and leads to elimination of any development errors. As already mentioned, in evaluation results are compared to original requirements.

Besides measurements, the questionnaire (Table 4) was introduced to give feedback on different approaches used for information system development. The aim was to capture users view on important aspects such as capability of approach to express static and dynamic aspects of the system, the required effort to progress from design to implementation stage, level of user participation etc. All questions were in Liker-type format.

Table 4. Questionnaire for measuring the development process

	Time spent [min]	Outputs
Analysis Includes user requirements analysis.		
Logical design Design independent of physical environment.		
Physical design		
Programming Physical development of the system.		
Testing Planning as well as the testing of systems, programs, and procedures.		
Implementation Planning and implementation of technical, social, and organizational aspects.		
Evaluation Measurement and evaluation of the implemented system and a comparison with the original objectives.		

The aim of the experiment was to investigate following hypotheses presented in Table 5 and after collecting the data the analysis was performed using SPSS toolkit.

Table 5. Hypotheses of the experiment

H₁	Time spent to complete software development of selected use cases is shorter with Panorama than object-oriented approach.
H₂	Panorama puts emphasis on analysis, logical design and testing, while object-oriented approach on physical design and programming.
H₃	Panorama is more appropriate for domains with static (data oriented) rather than dynamic (process oriented) components.
H₄	With Panorama approach in contrast to object-oriented approach, users are more encouraged to participate, required technical knowledge for development is lower, introducing additional functionalities is less demanding and the approach is easier to learn.

Results

A total of 2 case studies (C₁ and C₂), 28 raters (P₁ to P₄ and S₁ to S₂₄) and 2 approaches (Panorama and Object-oriented) were included in the experiment. Aforementioned resources resulted in 56 executions using different approaches on different use cases (2 x 2 x 14).

To test for differences in rater bias ANOVA (ANalysis Of VAriance) model was used, including the calculation of intraclass correlation (ICC) and to describe raters' marginal distributions graphical method of histograms was employed.

With intraclass correlation (ICC) [40-42] rating reliability was assessed by comparing the variability of different ratings of the same subject (questions from the questionnaire and effort distributions on Software Development activities by approach) to the total variation across all ratings and all subjects. ICC is calculated according to the following formula:

$$ICC = \frac{\sigma^2(\text{subjects})}{\sigma^2(\text{subjects}) + \sigma^2(\text{raters})} \quad (1)$$

Because in practice we do not know the true values of $\sigma^2(\text{subjects})$ and $\sigma^2(\text{raters})$, we must instead estimate them from sample data. With calculation of ICC we have to be aware that there exist several types – Case 1, Case 2 and Case 3. For our experiment Case 2 was selected, where the same set of k raters rate each subject. This corresponds to a fully-crossed (rater x subject), 2-way ANOVA design in which both Subject and Rater are separate effects. Because rater is considered a random effect, this means that k raters in the study are considered a random sample from a population of potential raters; therefore this type 2 of ICC estimates the reliability of the larger population of raters.

Hypothesis H₁

The claim of H₁ is that time spent to complete software development of selected use cases is shorter with Panorama than Object-oriented approach. The hypothesis was tested with 2-way ANOVA model with factors *Approach* and *Case study* and dependent variable *Total time*. Before conducting ANOVA test, exploratory data analysis was performed to confirm that data is normally distributed, so first pre-requirement to proceed with ANOVA was met. Levene's test of equality of error variances also indicated that the error variance of the dependent variable is equal across the groups, i.e. the assumption of the ANOVA test has been met.

Table 6. Tests of between-subjects effects on dependent variable Total time

	F-value	Sig.
Corrected model	F (3, 55) = 46,44	✓
Intercept	F (1, 55) = 6.357,42	✓
Approach	F (1, 55) = 137,48	✓
Case study	F (1, 55) = 0,05	✗
Approach x Case study	F (1, 55) = 1,78	✗

The results of 2-way ANOVA test are depicted in Table 6, where is clearly seen that the only significant effect on Total time variable is of the factor Approach, while factor Case study and Approach together with Case study don't have a significant effect on Total time.

Based on this findings we can reject the null hypothesis of mean times spent to complete software development of selected cases are the same for Object-oriented and Panorama approach. This leads to accepting alternative hypothesis of mean times for development being significantly different. Descriptive statistics in Table 7 and graphical presentation of mean time spent to complete software development of use cases clearly show that time required to complete development is shorter with Panorama than Object-oriented approach.

Table 7. Descriptive statistics of total time

Approach	Case study	N	Mean
Object-oriented	C ₁	14	12h 59min
	C ₂	14	13h 25min
	Total	28	13h 22min
Panorama	C ₁	14	9h 59min
	C ₂	14	9h 40min
	Total	28	9h 49min
Total	C ₁	14	11h 29min
	C ₂	14	11h 33min
	Total	28	11h 31min

To conclude, there is a significant difference between the mean time spent to complete software development of selected use cases using Object-

oriented and Panorama approach, with $F(1, 55) = 137,48$ and level of significance $p < 0,05$ and that time is shorter with Panorama than object-oriented approach.

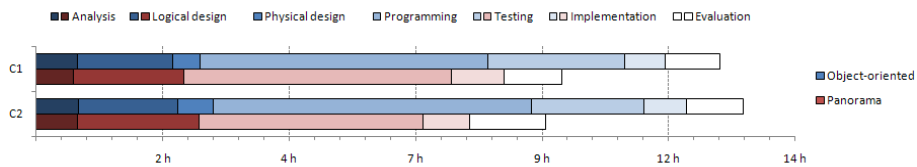


Fig. 4. Mean time spent to complete software development of use cases

With use cases C_1 and C_2 the development with Panorama was 25% faster than with Object-oriented approach, as depicted in Fig. 4.

Hypothesis H_2

Hypothesis H_2 asserts that Panorama puts emphasis on analysis, logical design and testing, while object-oriented approach on physical design and programming.

When examining the data we can see from Fig. 4 that effort distribution by approaches (Object-oriented and Panorama) is not equally divided. The comparison depicts distribution on Software Development activities by approaches, where we can notice that in Panorama approach majority of the time is spent in Analysis, Logical design and Testing, while no time is allocated to Programming and Physical design. In contrast to Panorama is in Object-oriented approach time allocated to Programming and Physical design quite substantial. The Programming and Physical design phase in Panorama is omitted, due to the fact that user defines all the requirements in Analysis and Logical design, while the code at implementation level is automatically generated and supported by Panorama tool.

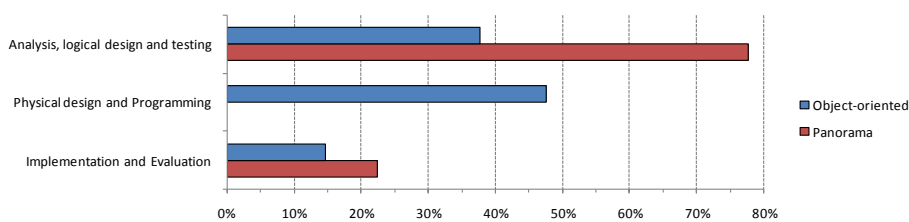


Fig. 5. Mean effort distribution on Software Development activities grouped by significant factors

For better comparison of effort distribution on Software Development activities the visualization depicted in Fig. 5 groups aforementioned phases and approaches. The data is normalized regarding the total time required to

complete use cases C_1 and C_2 and belonging percentage of time of selected phase is than displayed.

To test the significance of the results multiple 1-way ANOVA tests with factor *Approach* and following groups of dependent variables: A) *Analysis, logical design and testing*; B) *Physical design and programming* and C) *Implementation and evaluation* were conducted. Exploratory data analysis confirmed that data is normally distributed and Levene's test also confirmed the assumption of the ANOVA test. The results of ANOVA test and accompanying descriptive statistics are depicted in, where we can confirm for all 3 factors I, II and III that the mean effort distribution between Object-oriented and Panorama approach differs significantly.

The most significant is difference in mean effort distributions of *Physical design and programming* between Object-oriented and Panorama approach with $F(1, 54) = 4.668$ and level of significance $p < 0,05$, which confirms the fact that in Panorama users don't spent time for physical design and programming, because this is done automatically by underlying system. The second most significant difference in mean effort distributions is of *Analysis, logical design and testing* between Object-oriented and Panorama approach with $F(1, 54) = 1.755$ and level of significance $p < 0,05$, which confirms the fact that this phases are significantly more addressed in Panorama than in Object-oriented approach. Mean effort distributions of *Implementation and evaluation* is with $F(1, 54) = 107$ and level of significance $p < 0,05$ also significant but is less notable than with previous factors. Based on these findings we can confirm hypothesis H_2 .

Table 8. ANOVA results and descriptive statistics of combined effort distributions on Software Development activities

Software development phases	Approach	N	Mean	F-value	Sig.
I) Analysis, logical design and testing	Object-oriented	28	37,76%	F (1, 54) = 1.755	✓
	Panorama	28	77,64%		
	Total	56	57,70%		
II) Physical design and programming	Object-oriented	28	47,65%	F (1, 54) = 4.668	✓
	Panorama	28	0,00%		
	Total	56	23,83%		
III) Implementation and evaluation	Object-oriented	28	14,59%	F (1, 54) = 107	✓
	Panorama	28	22,36%		
	Total	56	18,47%		

More detail interpretation of time allocations by software development activities and critical overview can be found in [43].

Questionnaire analysis

The results of the questionnaire are summarized in Table 9 and grouped by selected approach. Each of the 7 questions was rated on a Likart-type scale from 1 to 10.

Table 9. ANOVA results and descriptive statistics of question analysis from questionnaire

	Mean score by approach			F-value	Sig.
	OO	Panorama	Total		
C. What is your experience with the methodology you are assessing? (1 – somewhat familiar, 4 – knows its details, 7 – I've used it, 10 – I've created it)	3,89	2,32	3,11	F (1, 54) = 8,86	✓
D. The notation is capable of expressing models of both static aspects of the system (e.g. structure) and dynamic aspects (e.g. processing)? (1 – strongly disagree, 4 – disagree, 7 – agree, 10 – strongly agree)	8,57	4,32	6,45	F (1, 54) = 187,15	✓
E. Did you find the notation and modeling language easy to learn? (1 – very hard, 4 – hard, 7 – fairly easy, 10 – very easy)	6,07	7,57	6,82	F (1, 54) = 38,04	✓
F. Identify the required a priori technical knowledge for conducting analysis and design stage of software development life cycle! (1 – very detailed software development knowledge, 3 – advanced software development knowledge, 7 – basics of software development, 10 – intuitive approach, very little knowledge required)	6,09	7,29	6,68	F (1, 54) = 20,81	✓
G. Estimate the effort required to progress from design into implementation stage! (1 – manually with user intervention, 3 – semi-automatically with major user interventions, 7 – semi-automatically with minor user interventions, 10 – automatically, without user intervention)	4,82	9,07	6,95	F (1, 54) = 379,69	✓
H. How well is user participation encouraged and supported? (1 – very difficult, only professional developers are involved in software development, 4 – satisfactory, users are involved on their request, 7 – intermediate, with users involvement in early stages of software development, 10 – very well, the user can even build a simple system from scratch to working prototype)	7,04	9,32	8,18	F (1, 54) = 87,63	✓

I. Estimate the effort required introducing additional functionalities in working software product! (1 – very demanding, it requires complete reorganization of software product and involvement of multiple actors, 4 – intermediate, new user requirements are passed to developers, 8 – minor changes can be done immediately with cooperation between users and developers, 10 – very intuitive, user can introduce changes to software product without the support of technical staff)	4,46	8,21	6,34	F (1, 54) = 267,93	✓
---	-------------	-------------	------	------------------------------	---

The first question (C) addressed evaluators' background and their experience with given approach. As expected, evaluators had more experience in Object-oriented approach (mean = 3,89) than Panorama (mean = 2,32). That's because they previously hadn't worked with Panorama but evaluators S_1 to S_{24} had introductory knowledge and P_1 to P_4 had experience in Object-oriented approach. Mean experience with the methodology is with **F (1, 54) = 8,86** and level of significance **p < 0,05**, which confirms that evaluators have different experience in selected approaches.

Hypothesis H₃

The claim of H_3 is that Panorama is more appropriate for domains with static (data oriented) rather than dynamic (process oriented) components. This aspect of development approach was addressed in question D (see Table 9) where users had to indicate whether the approach is capable of expressing models of both static aspects of the system (e.g. structure) and dynamic aspects (e.g. processing). The difference in *mean notation capability of expressing dynamics* with **F (1, 54) = 187,15** and level of significance **p < 0,05**, which confirms that null hypothesis can be rejected and alternative hypothesis is confirmed. Mean notation capability of expressing dynamics in Object-oriented approach is **8,57** and with Panorama **4,32**, that leads to conclusion that Panorama with score of 4,32 on a 1 to 10 scale is not appropriate for problem domains with emphasis on processing but rather for data oriented applications. Object-oriented approach has, as expected, scored very high score, because its notation is capable of expressing statics and dynamics of the system.

Hypothesis H₄

Hypothesis H_4 asserts that with Panorama approach in contrast to Object-oriented, users are more encouraged to participate, required technical knowledge for development is lower, introducing additional functionalities is less demanding and the approach is easier to learn. The aforementioned hypothesis is concerned about questions E, F, G, H and I from the questionnaire depicted in Table 9.

Based on analysis of aforementioned questions from Table 9 (simplicity to learn the approach, low a priori technical knowledge, simplicity of progress from design to implementation, user participation and simplicity of introducing additional functionalities) we can confirm with significance that hypothesis H_4 is valid.

Association and bias of raters

Intraclass correlation (ICC) across raters P_1 to P_4 and S_1 to S_{24} was analyzed. In our experiment this was the case 2 ICC with the same set of raters that rate each subject, which corresponds to a fully-crossed (rater x subject), 2-way ANOVA design in which both subject and rater are separate effects. There were 28 raters (P_1 to P_4 and S_1 to S_{24}) and 30 subjects (8 questions about the development process and 7 effort distributions of development time) included in ICC calculation. In case 2, rater is considered a random affects, which means the raters in the study are considered a random sample from a population of potential raters. The case 2 ICC estimates the reliability of the larger population of raters. ICC was used separately to evaluate expert and student users and to estimate the reliability of a single rating (consistency) and the reliability of a mean of several ratings (absolute agreement). The result of both measurements is depicted in Table 10.

Table 10. Intraclass correlation coefficients

	Consistency		Absolute agreement	
	Single measures ICC(2,1)	Average measures ICC(2,k)	Single measures ICC(2,1)	Average measures ICC(2,k)
Experts (k=4)	0,949	0,987	0,943	0,985
Students (k=24)	0,959	0,998	0,959	0,998
All (k=28)	0,948	0,998	0,946	0,998

The results in Table 10 denote that there exist a statistically significant positive correlation between all raters – experts and students. Both reliability of a single rating and reliability of a mean of several ratings are high. ICC approaches 1 when there is no variance within subjects, indicating total variation in measurements on the Likart scale is due solely to the target variable. ICC is high because any given subject tends to have the same score across the raters. ICC is also interpreted as the ratio of variance explained by the independent variable divided by total variance, where total variance is the explained variance plus variance due to the raters plus residual variance. High ICC is a result of no or little variance due to the raters and no residual variance to explain.

Based on the nature of Case 2 intraclass correlation with random effects that was used, the results obtained can be generalized to other raters.

Discussion

The results show that time spent to complete software development of selected use cases C_1 and C_2 is shorter when using Panorama approach opposed to using Object-oriented approach (see Fig. 4). In Panorama approach the emphasis is on logical design stage, where business vocabulary is constructed and on testing stage, where actual data is inserted into the system and actual testing is performed. On the other hand with Object-oriented approach extra work is conducted in programming and physical design stage.

In average this results in 25% less time required to complete software development life cycle using Panorama approach towards Object-oriented. It has to be noted that this is valid for problem domains with emphasis on data oriented aspects rather than process oriented.

The results of questionnaire analysis (see Table 9) show that the experience of raters is higher in Object-oriented approach due to raters' background and high dissemination of Object-oriented approach in modern software development projects. Panorama approach is not as strong as Object-oriented approach in expressing dynamic aspects (e.g. processing) of the system, while it focuses on static aspects (e.g. structure). Notation and modeling language is slightly easier to learn in Panorama approach as it is mainly focused on data modeling techniques. The same applies to a priori technical knowledge for analysis and design. As progressing from design to implementation stage is concerned Panorama turns out to be preferred approach due to availability of automatic code generation and transformation from logical to physical design. Following JAD approach Panorama also emphasizes user participation and received better score than Object-oriented approach. An important aspect is also the effort for introducing additional functionalities in working software product, whereas Panorama approach scored higher than object-oriented approach because of almost nonexistent programming phase which is automatically implemented in Panorama tool.

4. Conclusions and future work

In this paper a software solution for facilitating associative thinking paradigm, Panorama, was presented. The main goal was to create a "user friendly" tool for information management and software development for people with low IT skills. This was accomplished by introducing visualization in multi-dimensional space with focus on the chosen node (object of observation) and its nearest neighbours (object's context). Only semantically nearest nodes with appropriate attributes are being displayed with a possibility of freely traversing through information space by changing to the new object of observation. The user interface we have developed is very simple and pursues presenting and organizing the information on the screen in a way that user is not overloaded

but still has the ability to expand the context and view the situation in all its extent (following focus + context information visualization techniques).

We can conclude that the conceptual idea of using associative thinking paradigm in computer aided software development was successfully realized in several real world cases. By following the proposed methodology with the Panorama tool, the user is given an opportunity to design the application and start using it, without extensive technical knowledge. Panorama especially minimizes the impact on the last two phases in Software Development Life Cycle (implementation and review), due to existence of all required mechanisms for updating and viewing the content of data vocabulary and controlling the input of data with referential integrity checking. Future trends in information visualization were also addressed. One of them was utilization of information visualization techniques in information systems development process that was realized in Panorama tool.

With the detailed comparison of Panorama approach to existing software development approaches Panorama is positioned as a specific problem-solving approach. It is not intended for process oriented domains, but rather for smaller environments, especially office management, CRM, record management, planning, maintenance etc. The conducted experiment pointed out that there was less time spent to complete software development of selected use cases than using Object-oriented techniques and approaches. One of the findings is also the effort distribution of Software Development activities. In Object-oriented approaches emphasis is on design and programming, while Panorama eliminates programming activity and is focused mainly on design and testing.

This research has several limitations as already pointed out in section 3. The Panorama approach is suitable only for problem domains that emphasis data oriented problem domains while it lacks to support the ones with process oriented aspects.

The results of this research could also assist ontology management community, where Panorama can be used as an ontology editor. Export of Panorama's data vocabulary and known objects space to XML format is already supported, but clearly further work is required to make data available in ontological languages with more expressive power. In our future work we will therefore expand the variety of export formats that Panorama supports. Data from Panorama available in a form of ontology could enable reuse of knowledge in intelligent systems with inference capabilities to derive new knowledge.

References

1. Harbron, T.R., *File Systems: Structures and Algorithms*. 1988, New York, USA: Prentice Hall.
2. Tharp, A.L., *File organization and processing*. 1988, New York, USA: John Wiley & Sons.

3. Caramia, M. and G. Felici, *Mining relevant information on the Web: a clique-based approach*. International Journal of Production Research, 2006. **44**(14): p. 2771-2787.
4. Tell, B.V., Towards Intelligent Management of Information Resources - a Case for Special-Libraries or the National Intelligence. Libri, 1992. **42**(1): p. 20-34.
5. Ortner, E., Enterprise Data Modeling as a Basis for Integrated Information-Processing in Business and Administration. Wirtschaftsinformatik, 1991. **33**(4): p. 269-280.
6. Lavbič, D., O. Vasilecas, and R. Rupnik, *Ontology-based Multi-Agent System to support business users and management*. Technological and Economic development of Economy, 2010. **16**(2): p. 327-347.
7. Ware, C., *Information visualization: Perception for design, 2nd edition*, ed. S. Card, J. Grudin, and J. Nielsen. 2004, San Francisco, USA: Morgan Kaufmann.
8. Atkinson, R.C. and R.M. Shiffrin, *Human memory: A proposed system and its control processes*. The Psychology of Learning and Motivation: Advances in Research and Theory, 1968. **2**: p. 89-195.
9. Waugh, N.C. and D.A. Norman, *Primary Memory*. Psychological Review, 1965. **72**(2): p. 89-104.
10. Obrenović, Ž. and D. Starčević, *Adapting the Unified Software Development Process for User Interface Development*. Computer Science and Information Systems, 2006. **3**(1).
11. Chakrabarti, S., et al., *Using Memex to archive and mine community Web browsing experience*. Computer Networks, 2000. **33**(1-6): p. 669-684.
12. Cole, C., B. Mandelblatt, and J. Stevenson, Visualizing a high recall search strategy output for undergraduates in an exploration stage of researching a term paper. Information Processing & Management, 2002. **38**(1): p. 37-54.
13. Bush, V., *As we may think*. The Atlantic Monthly, 1945. **176**.
14. Ahmed, S. and G. Ashraf, *Model-based user interface engineering with design patterns*. Journal of Systems and Software, 2007. **80**(8): p. 1408-1422.
15. Tergan, S.-O. and T. Keller, *Knowledge and Information Visualization: Searching for Synergies*. Lecture Notes in Computer Science. Vol. 3426. 2005, Heidelberg, Germany: Springer-Verlag.
16. Codd, E.F., *A Relational Model of Data for Large Shared Data Banks*. Communications of the ACM, 1970. **13**(6): p. 377-387.
17. Dietrich, S.W., *An Advanced Course in Database Systems: Beyond Relational Databases*. 2004, New York, USA: Prentice Hall.
18. Kim, W., *Introduction to Object-Oriented Databases*. 1990, Cambridge, USA: MIT Press.
19. Bernstein, P.A., *Repositories and Object Oriented Databases*. ACM SIGMOD Record, 1998. **27**(1): p. 88-96.
20. Jutla, D., P. Bodorik, and Y. Wang, *Developing internet e-commerce benchmarks*. Information Systems, 1999. **24**(6): p. 475-493.
21. Rob, M.A., Dilemma between the structured and object-oriented approaches to systems analysis and design. Journal of Computer Information Systems, 2006. **46**(3): p. 32-42.
22. Avison, D. and G. Fitzgerald, *Information Systems Development: Methodologies, Techniques and Tools, 4th edition*. 2006, Maidenhead, UK: McGraw-Hill.
23. Coad, P. and E. Yourdon, *Object Oriented Analysis, 2nd edition*. 1990, New York, USA: Prentice Hall.
24. Jacobson, I., G. Booch, and J. Rumbaugh, *The Unified Software Development Process*. 1999, Boston, USA: Addison-Wesley.

25. Martin, J., *Rapid Application Development*. 1991, Indianapolis, USA: MacMillan Publishing. 788.
26. Jeffries, R., A. Anderson, and C. Hendrickson, *Extreme Programming Installed*. 2000, Boston, USA: Addison-Wesley.
27. Vidgen, R., et al., *Developing Internet Applications*. 2002, Oxford, UK: McGraw-Hill.
28. Milosavljević, G. and B. Perišić, *A Method and a Tool for Rapid Prototyping of Large-Scale Business Information Systems*. Computer Science and Information Systems, 2004. **1**(2).
29. Schreiber, G., et al., *Knowledge Engineering and Management - The CommonKADS Methodology*. 1999, London, England: The MIT Press: Cambridge, Massachusetts.
30. Griggs, K.A., R.H. Wild, and E.Y. Li, *A Web-based knowledge management environment for consulting and research organizations*. Journal of Computer Information Systems, 2002. **42**(5): p. 110-118.
31. Smaizys, A. and O. Vasilecas, *Business Rules based agile ERP systems development*. Informatica, 2009. **20**(3): p. 439-460.
32. Gonzales-Perez, C. and B. Henderson-Sellers, *Modelling software development methodologies: A conceptual foundation*. Journal of Systems and Software, 2007. **80**(11): p. 1778-1796.
33. Frankel, D.S., *Model Driven Architecture: Applying MDA to Enterprise Computing*. 2003: Wiley.
34. Brinkkemper, S., *Method engineering: Engineering of information systems development methods and tools*. Information and Software Technology, 1996. **38**(4): p. 275-280.
35. Vavpotič, D. and M. Bajec, *An approach for concurrent evaluation of technical and social aspects of software development methodologies*. Information and Software Technology, 2009. **51**(2): p. 528-545.
36. Henderson-Sellers, B. and J. Ralyte, *Situational Method Engineering: State-of-the-Art Review*. Journal of Universal Computer Science, 2010. **16**(3): p. 424-478.
37. Henderson-Sellers, B., *The OPEN framework for enhancing productivity*. IEEE Software, 2000. **17**(2): p. 53-+.
38. ISO/IEC-24744, *Software Engineering - Metamodel for Development Methodologies*. 2007.
39. Lajovic, I. *Panoramic view on databases*. in Proceedings of the Slovenian Informatics conference. 2005. Portorož, Slovenia.
40. Shrout, P.E. and J.L. Fleiss, *Intraclass Correlations: Uses in Assessing Rater Reliability*. Psychological bulletin, 1979. **2**: p. 420-428.
41. Yaffee, R.A. *Enhancement of Reliability Analysis: Application of Intraclass Correlations with SPSS/Windows v.8*. 1998; Available from: <http://www.nyu.edu/its/statistics/Docs/intracls.html>.
42. Hart, T.A., B.S. Chaparro, and C.G. Halcomb, *Evaluating websites for older adults: adherence to 'senior-friendly' guidelines and end-user performance*. Behaviour & Information Technology, 2008. **27**(3): p. 191-199.
43. Kruchten, P., *What do software architects really do?* Journal of Systems and Software, 2008. **81**(12): p. 2413-2416.

Dejan Lavbič is a teaching assistant at University of Ljubljana, Faculty of Computer and Information Science. He graduated in 2004 and received his PhD in 2010 at Faculty of Computer and Information Science at University of Ljubljana. He is author of 5 original scientific papers and more than 10 published scientific conference contributions. His research interests are intelligent agents, Multi-Agent Systems, knowledge management, ontologies, business rules and Semantic Web. His work is mainly oriented toward using Semantic Web technologies in business applications. He has been involved in several research and commercial projects on strategic planning, methodologies for IS development, using intelligent agents and business process automatization and management.

Iztok Lajovic is author of panoramic dataview concept and has developed the Panorama database program. He is managing his own software development company KreS, Kreativni Sistemi, d.o.o. in Ljubljana, Slovenia. His main interests are database design, GUI and intelligent systems.

Marjan Krisper received the M.Sc. degree in information systems engineering from the University of Ljubljana, Ljubljana, Slovenia, in 1977 and the Ph.D. degree in expert systems from the University of Belgrade, Belgrade, Yugoslavia, in 1989. He is an Associate Professor, the Chair of Information Science, and the Head of the Information Systems Laboratory at the University of Ljubljana, Faculty of Computer and Information Science. His research interests include information systems, information systems development methodologies, information systems strategic planning, and electronic business. Dr. Krisper is a member of the Association of Information Systems and a senior member of the Project Management Institute Slovenian Chapter.

Received: November 14, 2009; Accepted: November 10, 2010.

Metrics for Evaluation of Metaprogram Complexity

Robertas Damaševičius¹ and Vytautas Štuikys¹

¹Kaunas University of Technology, Software Engineering Department, Studentų 50,
LT-51368, Kaunas, Lithuania
{robertas.damasevicius, vytautas.stuikys}@ktu.lt

Abstract. The concept of complexity is used in many areas of computer science and software engineering. Software complexity metrics can be used to evaluate and compare quality of software development and maintenance processes and their products. Complexity management and measurement is especially important in novel programming technologies and paradigms, such as aspect-oriented programming, generative programming, and metaprogramming, where complex multi-language and multi-aspect program specifications are developed and used. This paper analyzes complexity management and measurement techniques, and proposes five complexity metrics (Relative Kolmogorov Complexity, Metalanguage Richness, Cyclomatic Complexity, Normalized Difficulty, Cognitive Difficulty) for measuring complexity of metaprograms at information, metalanguage, graph, algorithm, and cognitive dimensions.

Keywords: Metaprogramming, complexity evaluation, metaprogram metric.

1. Introduction

Complexity is a difficult concept to define. Though the term “*complexity*” is used in many of 25 roadmaps for software [1] and can, e.g., be found in relation to software development, software metrics, software engineering for safety, reverse engineering, configuration management, empirical studies of software engineering [2], so far there is no exact understanding of what is meant by complexity with various definitions still being proposed. High complexity of a system usually means that we cannot represent it in a short and comprehensive description. L.C. Briand *et al.* [3] state that complexity (of a modular software system) is a system property that depends on the relationships among elements and is not a property of any isolated element. IEEE Std. 610.12:1990 [4] defines software complexity as “*the degree to which a system or component has a design or implementation that is difficult to understand and verify*”. Therefore, complexity relates both to comprehension complexity as well as to representation complexity.

The other definition deals with *psychological complexity* (also known as *cognitive complexity*) of programs explaining that “*true meaning of software complexity is the difficulty to maintain, change and understand software*” [5]. There are 3 specific types of psychological complexity that affect programmer ability to comprehend software: *problem complexity*, *system design complexity*, and *procedural complexity* [6]. Problem complexity is a function of the problem domain. It is assumed that complex problem spaces are more difficult for a programmer to comprehend than simple problem spaces.

Knowledge-based perception of software complexity is described in [7] as a process of “*translating*” human-seen complexity into numbers. The process starts with an experiment that involves human beings and provides data with embedded knowledge about human perception of complexity. Data processing and analysis of data models lead to discovery of simple rules which represent human perception of software complexity.

From the organizational viewpoint, complexity of a system is defined with respect to *number*, *dissimilitude* and *states’ variety* of system elements and relationships between them [8]. These complexity variables enable distinction between *structural (static)* and *dynamic* complexity. Structural complexity describes the system structure at a defined point in time, and dynamic complexity represents the change of system configuration in time.

How to manage complexity? Many factors influence on better management of complexity. E.g., from the cognitive complexity viewpoint, the major factor is *understandability* [9]. One way to avoid exceeding the cognitive constraints and creating cognitive overload is to reduce the amount of information that needs to be stored in a short-term memory and to decrease the uncertainty of that information [10]. A common method to achieve this would be by creating new and useful *abstractions*. As a program is more than just the informative code during the process of understanding, a programmer’s level of expertise in a given domain, i.e., *domain knowledge*, greatly affects program understanding, as well as programmer’s knowledge. The commonly recognized principles for managing complexity are *reducing the amount of information*, *decomposing a system into modules*, *abstracting or hiding information*, and *providing different levels of abstractions*.

The abstraction level is the level of detail of a software system [11]. In this sense, abstraction is a basic property for understanding the reality and managing complexity of software systems. Abstraction is a gradual increase in the level of representation of a software system, when existing detailed information is replaced with information that emphasizes certain aspects important to the developer while other aspects are hidden. More abstract programming language mechanisms allow to replace complex and repeating low-level operations, and allow to address complex problems with less code and less programming errors.

Software design complexity is also related with design quality. As complexity increases, design quality also tends to decrease. To achieve the levels of quality needed in today’s complex software designs, quality must be *designed in*, not tested in. Thus the *design-for-quality* paradigm is becoming extremely important. In this context, M. Keating [12] proposed a simple

software design partitioning rules as a basis for a quantitative measure of complexity: *the number of modules at any level of hierarchy must be 7 +/- 2.*

The complexity growth forces researchers to seek for the adequate means for better management of complexity. A number of techniques has been identified and followed in software design practice that enforces higher program comprehensibility reuse and eases complexity management. These include various lexical conventions, design style conventions and design process conventions. The primary tasks are understanding of the complexity problem and finding of relevant measures for evaluating software complexity. These issues may have a direct influence on testability, performance, efficiency and other characteristics of software systems to be designed.

Software metrics have always been strongly related to the programming paradigm used by the respective researchers. E.g., McCabe's Cyclomatic Complexity [13] was proposed for measuring the testing efforts of structural programs. For object-oriented programs, complexity metrics are based on special object-oriented (OO) features, such as the number of classes, depth of inheritance tree, number of subclasses, etc. [13]. With the arrival of new higher-level programming paradigms such as aspect-oriented programming, generic programming or metaprogramming, new complexity metrics should be defined, because metrics applied to programs implemented in different paradigms than the one they were developed for may report false results [14].

The aim of this paper is to contribute towards research in software complexity measurement and management by defining complexity metrics specifically for metaprograms. The research is relevant because of the importance of ensuring metaprogram testability and reliability and developing effective metaprogram testing procedures, to which metaprogram complexity measures can contribute similarly to the contribution of software metrics to predict critical information about reliability and maintainability of software systems using automatic analysis of source code.

The outline of the paper is as follows. Section 2 discusses related works on complexity metrics. Section 3 analyzes evaluation of metaprogramming and metaprogram complexity. Section 4 describes the proposed metaprogram complexity metrics. Section 5 presents theoretical validation of the proposed metrics. Section 6 gives two examples of metaprogram complexity calculation. Finally, Section 7 presents conclusions and outlines future work.

2. Related Work on Complexity Metrics

Complexity is the intrinsic attribute of systems and processes through which systems are created. Complexity measures allow reasoning about system structure, understanding system behaviour, comparing and evaluating systems or foreseeing their evolution. System design complexity addresses complexity associated with mapping of a problem space into a given representation. An overall rating of system complexity (*System Complexity*) consists of the sum of the individual module complexities associated with the

module's connections to other modules (*Structural Complexity*) and the amount of work the module performs (*Data Complexity*) [15].

Structural complexity addresses the concept of *coupling*, i.e., the interdependence of modules of source code. It is assumed that the higher coupling between modules is, the more difficult it is for a programmer to comprehend a given module. Data complexity addresses the concept of *cohesion*, i.e., the intradependence of modules. In this case, it is assumed that the higher cohesiveness is, the easier it is for a programmer to comprehend a given module. The structural and data complexity measures are based on the module's fan-in, fan-out, and number of input/output variables. These metrics address system complexity at the system and module levels. Procedural complexity is associated with the complexity of the logical structure of a program assuming that the length of a program in Lines of Code (LOC) or the number of logical constructs such as sequences, decisions, or loops determines complexity of the program.

M. Rauterberg [16] addresses a similar problem, i.e., how to measure the cognitive complexity in human-computer interaction. He proposes to derive *cognitive complexity* (CoC) from *behaviour complexity* (BC), *system complexity* (SC) and *task complexity* (TC) as: $CoC = SC + TC - BC$.

S.D. Sheetz *et al.* [17] address complexity of the OO system at the application, object, method, and variable levels, and at each level propose the measures to account for the cohesion and coupling aspects of the system. Complexity of the OO system at each level is presented as a function of the measurable characteristics such as fan-in, fan-out, number of I/O variables, fan-up, fan-down, and polymorphism. Each measure is defined with adherence to the principles that measures must be intuitive and that they must be applicable to all phases of the OO development lifecycle.

Cyclomatic complexity is one of the more widely-accepted *static* software metrics [13]. It is intended to be independent of the language and language format. The other metrics bring out other facets of complexity, including both structural and computational complexity: *Halstead complexity measures* [18] identify algorithmic complexity, measured by counting operators and operands; *Henry and Kafura metrics* [19] indicate coupling between modules (parameters, global variables, calls); *Bowles metrics* [13] evaluate the module and system complexity, coupling via parameters and global variables; *Troy and Zweben* [13] metrics evaluate modularity or coupling; complexity of structure (maximum depth of a structure chart). Wang's *cognitive complexity measure* [20] indicates the cognitive and psychological complexity of software as a human intelligence artefact. With the arrival of new programming paradigms, new complexity metrics have been proposed for aspect-oriented programming (AspectJ) [21] and generic programming (C++ Standard Template Library) [22].

There were efforts to describe formal properties of complexity metrics that could be used for evaluation and theoretical validation of complexity measures. J. Weyuker [23] introduces a set of syntactic software complexity properties as criteria and examines the strengths and weaknesses of the known complexity measures, which include *statement count*, *cyclomatic*

number, effort measure, and data flow complexity. L.C. Briand *et al.* [3] provide a theoretical framework for relating structural complexity, cognitive complexity and external quality attributes.

3. Complexity of metaprogramming and metaprograms

It is a well-known fact that the same algorithm implemented in different programming paradigms or languages can have very different complexity of description (i.e., description complexity is not a property of an algorithm but rather a property of an implementation language). For example, one study [24] shows that complexity of Quick Sort algorithm implementations measured using Halstead Volume, Program Effort and Program Difficulty metrics [18] is highest for C and lowest for Assembly and Visual Basic language programs.

Metaprogramming [25], as a paradigm for developing programs that create other programs, is a level of complexity above traditional programming paradigms. There are two types of metaprogramming: homogeneous metaprogramming and heterogeneous metaprogramming.

In case of homogeneous metaprogramming, we have two subsets of a domain language: one is dedicated for expressing domain functionality, and the other is used for managing variability at meta-level (generic parameters, templates, etc.). The developer has to know only one programming language syntax, the metaprogram is as readable as a domain program written in the same domain programming language, and the development flow uses the same development toolset. Therefore, the complexity of developing metaprograms using homogeneous metaprogramming technique is only slightly higher than complexity of traditional programming.

In case of heterogeneous metaprogramming, we have two different languages: a domain language itself and a metalanguage, which manipulates with source code of domain language programs. As a result, the cognitive complexity of heterogeneous metaprograms expressed in terms of their readability and understandability is significantly higher, because the developer must know, understand and use the syntactical constructs of two different languages in the same metaspecification. The development flow is significantly more complex: not only two development environments have to be used, but also the testing of metaprograms is a significant and time-consuming problem. Therefore, complexity of developing metaprograms using heterogeneous metaprogramming techniques is considerably higher than complexity of traditional programming.

Complexity measures may be helpful for reasoning about metaprogram structure, understanding the relationships between different parts of metaprograms, comparing and evaluating metaprograms. Here we distinguish between: 1) first-order properties, or *characteristics*, which are derived directly from the metaprogram description itself using simple mathematical actions such as counting, e.g., program size (count of symbols in a file); and 2)

second-order properties, or *metrics*, which cannot be derived directly from artefacts, but are calculated from the first-order properties.

Metaprogram complexity can be evaluated at several dimensions:

1) **Information:** Metaprogram as message (sequence of symbols) containing information with unknown syntax and structure.

2) **Metalanguage:** Metaprogram as annotated domain knowledge. Domain knowledge is expressed using a domain language, whereas domain variability is specified using a metalanguage. Such separation of domain and meta levels is a first step towards the creation of a metaprogram.

3) **Graph:** Metaprogram as a graph of execution paths, where a root is a metaprogram, the nodes are the metalanguage constructs, and the leaves are the domain program instances.

4) **Algorithm:** Metaprogram as a high-level program specification (algorithm), which contains a collection of functional (structural) operations. An operation may have one or more operands specified as metaprogram attributes (parameters).

5) **Cognition:** Metaprogram as a number of different information units available for human cognition. A unit may represent either a metalanguage construct (macro, template, function), its argument or a meta-parameter.

4. Metaprogram Complexity Metrics

We use the following metrics for evaluating complexity at different dimensions of a metaprogram: Relative Kolmogorov Complexity (RKC), Metalanguage Richness (MR), Cyclomatic Complexity (CC), Normalized Difficulty (ND), and Cognitive Difficulty (CD).

4.1. Information dimension: Relative Kolmogorov Complexity

There are several methods to evaluate informational software complexity such as Shannon entropy, computational complexity, network complexity and topological complexity. We use the algorithmic complexity metric also known as *Kolmogorov Complexity* [26]. Kolmogorov complexity is a measure of randomness of strings and other objects based on their information content. Kolmogorov Complexity measures complexity of an object by the length of the smallest program that generates it. Suppose, we have an object x and a description system (e.g., a programming language) φ that maps from a description w to this object. Kolmogorov Complexity $K_\varphi(x)$ of an object x is the size of the shortest program in the description system φ capable of producing x on a universal computer:

$$K_\varphi(x) = \min_w \{ \|w\| : \varphi_w = x \} \quad (1)$$

Different description systems can provide distinct values of $K(x)$, but one can prove that the differences are only up to a fixed additive constant. Intuitively, Kolmogorov Complexity $K_\varphi(x)$ is the minimal size of information required to generate x by an algorithm. Unfortunately, it cannot be computed in the general case and must be approximated. Usually, compression algorithms are used to give an upper bound to Kolmogorov Complexity. Suppose that we have a compression algorithm C_i . Then, a shortest compression of w in the description system φ will give the upper bound to information content in x :

$$K_\varphi(x) \leq C(x) := \min_i \{ \|C_i\|, \varphi_{C_i} = x \} \quad (2)$$

Kolmogorov Complexity has been used earlier (under the name of Generative Software Complexity) to measure the effectiveness of applying program generation techniques to software [27]. Program generators were defined as compressed programs, and the shortest generator is assumed to have maximal generative complexity.

Here we evaluate the complexity of a metaprogram M using the Relative Kolmogorov Complexity (RKC) metric, which can be calculated using a compression algorithm C as follows:

$$RKC = \frac{\|C(M)\|}{\|M\|} \quad (3)$$

where $\|M\|$ is the size of a metaprogram M , and $\|C(M)\|$ is the size of a compressed metaprogram M .

A high value of RKC means that there is a high variability of text content, i.e., high complexity. A low value of RKC means high redundancy, i.e., the abundance of repeating fragments in metaprogram code.

4.2. Metalanguage dimension: Metalanguage Richness

Metaprogram M can be defined as a collection of domain language statements with corresponding annotations (metadata) expressed symbolically: $O = \langle (s, m) \mid s, m \in \Sigma^* \rangle$, where s is a domain language statement,

m is the metadata of s , and Σ^* is a string of symbols from alphabet Σ . For the evaluation of metaprogram complexity at the metalanguage dimension, we use the Metalanguage Richness (MR) metric:

$$MR = \frac{\sum_{m \in M} \|m\|}{\|M\|} \quad (4)$$

where $\|M\|$ is the size (length) of a metaprogram M , and $\|m\|$ is the size (length) of the metalanguage constructs in a metaprogram M .

A higher value of MR means that a metaprogram contains more metadata and its description is more complex.

4.3. Graph dimension: Cyclomatic Complexity

Cyclomatic Complexity (CC) [28] of a program directly measures the number of linearly independent paths through a program's source code from entrance to each exit. For metaprograms, CC is equal to the number of distinct domain program instances that can be generated from a metaprogram.

A metaprogram M can be defined as a function $\Phi(M): P \rightarrow I$ that maps from a set of its parameters P to a set of its domain program instances I . Following this definition, CC of a metaprogram is equal to the cardinality of a set of the distinct domain program instances described by a metaprogram.

$$CC = |\text{cod } \Phi| = |I| \quad (5)$$

Since Φ is an injective function, which associates distinct metaprogram parameter values with distinct domain program instances, the cyclomatic complexity of a metaprogram M can be computed using only the interface description of a metaprogram. For independent parameters, the value of CC can be calculated as a product of the number of allowed parameter values for each parameter of a metaprogram:

$$CC = \prod_{p \in P} |\text{dom } p| \quad (6)$$

A higher value of CC indicates higher complexity of the metaprogram's parameter set (meta-interface).

4.4. Algorithmic complexity: Normalized Difficulty

A functional program specification S is a sequence of functions $S = (f \mid f \in F)$, where $f: (a, a \in A) \rightarrow A$ is a specific function (operator) that may have a sequence of operands as its arguments, and A is a set of function operands. For metaprograms we accept that operations are specified as metalanguage functions, and operands are specified as metaprogram parameters. For the evaluation of metaprogram complexity at the algorithm dimension, we use the Halstead complexity metrics [18]. From a metaprogram we derive the number of distinct operators $n_1 = |F|$, the number of distinct operands $n_2 = |A|$, the total number of operators $N_1 = |S|$, the total number of operands $N_2 = \sum_{f \in S} |A|$.

Halstead Difficulty D indicates the cognitive difficulty of a program:

$$D = \left(\frac{n_1}{2} \right) \left(\frac{N_2}{n_2} \right) \quad (7)$$

The Halstead Volume V measures the size of a program specification:

$$V = N \log_2 n \quad (8)$$

For evaluating metaprogram complexity at the algorithm dimension we propose the Normalized Difficulty (ND) metric, which is a normalized ratio of the cognitive difficulty and size metrics:

$$ND = \frac{n_1 N_2}{(N_1 + N_2)(n_1 + n_2)} \quad (9)$$

The ND metric measures the complexity of a metaprogram as an algorithm. A high value of the ND metric means that metaprogram is highly complex in terms of time and effort required to understand it.

4.5. Cognitive complexity: Cognitive Difficulty

Following the works of G. Miller [29] stating that humans can hold 7 (+/- 2) chunks of information in their short-term memory at one time, and M. Keating [12], who claims that the number of modules at any level of software hierarchy must be 7 +/- 2, for evaluating complexity of metaprograms we propose the Cognitive Difficulty (CD) metric. Cognitive Difficulty is calculated as the maximal number of meta-level units (metaparameters P , metalanguage constructs N_1 , or their respective arguments N_2) in a metaprogram.

$$CD = \max(P, N_1, N_2) \quad (10)$$

The proposed metaprogram complexity metrics are summarized in Table 1.

Table 1. Summary of metaprogram complexity metrics

Metric	Objects of measurement	Meaning for a metaprogram
Relative Kolmogorov Complexity	Object: metaprogram Program: compressed metaprogram	High variability of content
Metalanguage Richness	Data: domain language constructs Metadata: metalanguage constructs	Complexity of description at meta level
Cyclomatic Complexity	Independent paths: number of distinct instances	Complexity of a meta-interface
Normalized Difficulty	Operators: metalanguage functions Operands: metaprogram parameters	Algorithmic complexity of a

		metaprogram
Cognitive Difficulty	Metaprogram parameters, metalanguage functions, metalanguage function arguments	Cognitive understandability of a metaprogram

5. Theoretical validation of complexity metrics

Validation of software metrics is important to ensure that metrics are accepted by the scientific community and used properly. There are two methods of metrics validation: *theoretical* and *empirical* [30]. Theoretical validation ensures that the metric is a proper numerical characterization of software property it claims to measure. Empirical validation relates metrics with some important external attributes of software (such as the number of faults). While both types of validation are necessary, the empirical validation requires much time and many researchers to contribute since many studies need to be performed to gather convincing evidence from many real-world libraries and applications that a metric is valid. The domain of metaprogram complexity research is not mature yet, therefore while there are open metaprogram libraries available (such as Boost [31] in C++) for such research currently there are not sufficient data available publicly on the external characteristics of such metaprograms such as reliability or maintainability.

Therefore, we validate the proposed metaprogram complexity metrics theoretically using Weyuker's properties [23], a set of formal properties that can be used to evaluate any software metrics.

Property 1 (Eq. 11) will be satisfied when we can find two metaprograms of different complexity. All proposed complexity metrics satisfy Property 1.

$$(\exists P)(\exists Q)(|P| \neq |Q|) \quad (11)$$

Property 2 is satisfied when there are finitely many programs of complexity c , where c is a non-negative number. The property is not satisfied for all complexity measures that are size-independent (scaled). Therefore Property 2 is not satisfied for all proposed metaprogram complexity metrics.

Property 3 (Eq. 12) is satisfied if we can find two distinct metaprograms that have equal complexity. The property is satisfied by all proposed metaprogram complexity metrics.

$$(\exists P)(\exists Q)(P \neq Q \wedge |P| = |Q|) \quad (12)$$

Property 4 (Eq. 13) is satisfied if equivalent metaprograms of different complexity can be written. The property is not satisfied by RKC and MR metrics.

$$(\exists P)(\exists Q)(P \equiv Q \wedge |P| \neq |Q|) \quad (13)$$

Property 5 (Eq. 14) is satisfied if after concatenating two metaprograms, the complexity of the merged metaprogram increases beyond individual

complexities of original metaprograms. The property is satisfied by all metrics, except MR (because of averaging).

$$(\forall P)(\forall Q)(|P| \leq |P+Q| \& |Q| \leq |P+Q|) \tag{14}$$

Property 6 (Eq. 15) is satisfied if concatenation of two equal complexity metaprograms with some other metaprogram gives different complexity metaprograms. The property is satisfied by all metrics (because metaprograms can have common metaparameters, but distinct metabodies).

$$(\exists P)(\exists Q)(\exists R)(|P| = |Q| \& |P;R| \neq |Q;R|) \tag{15}$$

Property 7 is satisfied if by permuting the order of statements in a metaprogram, the complexity of a metaprogram changes. The property is not satisfied by all metaprogram complexity metrics except RKC metric.

Property 8 is satisfied if renaming of the symbols and variables of a metaprogram does not change the complexity of a program. The property is satisfied for all metaprogram complexity metrics except RKC metric.

Properties 9a (Eq. 16) and **9b** (Eq. 17) are satisfied when a two (or more) metaprograms are concatenated, the sum of complexities of the original metaprograms is less than the complexity of the bigger metaprogram. The property is satisfied by RKC (because the concatenation provides more opportunities for compression), CC (because adding new metaparameters leads to geometrical increase of metaprogram instance number), CD (because two metaprograms can have the same metaparameters, metalanguage constructs or their arguments) metrics. Properties 9a and 9b are not satisfied by MR metric (because combining two metaprograms will not lead to their increased coupling). Only property 9a is satisfied by ND metric.

$$(\exists P)(\exists Q)(|P| + |Q| < |P;Q|) \tag{16}$$

$$(\forall P)(\forall Q)(|P| + |Q| \leq |P;Q|) \tag{17}$$

Table 2. Summary of metaprogram complexity validation

Complexity metric	Weyuker's property								
	1	2	3	4	5	6	7	8	9
RKC	+	-	+	-	+	+	+	-	+
MR	+	-	+	-	-	+	-	+	-
CC	+	-	+	+	+	+	-	+	+
ND	+	-	+	+	+	+	-	+	+/-
CD	+	-	+	+	+	+	-	+	+

The results of theoretical validation are summarized in Table 2. Note that Weyuker's properties were developed for procedural languages. Hence, there might be possibility that a proposed metaprogram complexity measure may not satisfy all the properties, but still may be valid for metaprogramming

domain as, e.g., some object-oriented metrics that do not satisfy Weyuker's properties are still considered valid for object-oriented programs [32].

6. Example of metaprogram complexity calculation

6.1. Heterogeneous metaprogramming

We demonstrate the complexity calculation of the heterogeneous metaprogram developed for hardware design domain. In hardware design domain, a great number of similar domain entities exist. For example, the most widely used hardware library components are gates (see Fig. 1; in VHDL), which implement a particular logical function. The hardware designer requires many different gate components implementing different functions and having a different number of inputs. All these components are very similar to each other both syntactically and semantically, and thus they constitute a component family.

```
entity gate is
  port ( X1, X2 : in bit; Y : out bit );
end gate;

architecture behave_gate of gate is
begin
  Y <= X1 and X2;
end behave_gate;

entity gate is
  port ( X1, X2, X3 : in bit; Y : out bit );
end gate;

architecture behave_gate of gate is
begin
  Y <= X1 or X2 or X3;
end behave_gate;
```

Fig. 1. Instances of VHDL gate family: a) 2-input AND gate, and b) 3-input OR gate

Next, we develop a metaprogram, which describes a gate component family. For example, the identified generic parameters and their values for the gate component family are as follows:

```
Gate_function = { AND, OR, XOR, NAND, NOR, XNOR }
Gate_inputs = { integer numbers from 2 to 8 }
```

A *gate* metaprogram (see Fig. 2) was developed using Open PROMOL [33] metalanguage. The metaprogram has 2 parameters, 3 metalanguage functions, and its size is 291 B.

```

$
"Enter gate function: " {and, or, xor, nor, nand, xnor} f := and;
"Enter number of inputs:" {2..8} num := 2;
$
entity gate is
  port ( @gen[num,{, }] : in bit; Y : out bit );
end gate;

architecture behave_gate of gate is
  begin
    Y <= @gen[num, { @sub[f] }];
  end behave_gate;

```

Fig. 2. Generic gate described using Open PROMOL metalanguage

We calculate RKC value using a BWT (*Burrows-Wheeler Transform*) compression algorithm, because currently it allows achieving best compression results for text-based information and thus allows to better approximate information content. The size of the *gate* metaprogram is 271 B. The size of the compressed metaprogram will put the upper limit on its information content. After compression we obtain 245 B, therefore RKC value of a *gate* metaprogram is equal to $245/271 = 0.90$.

We calculate MR of the *gate* metaprogram by calculating the size of its metainterface and the length of its metalanguage functions, which is equal to 139 B. Therefore, its MR value is equal to $139/271 = 0.51$.

Cyclomatic Complexity of a metaprogram is a number of different program instances that can be generated from it. The metric can be calculated as the number of distinct metaprogram parameter values. Parameters *f* and *num* are independent. Parameter *f* can have 6 different values, and parameter *num* can have 7 values. The *gate* metaprogram covers a family of $6 \cdot 7 = 42$ different component instances. Therefore, its CC value is 42.

The *gate* metaprogram has 3 metalanguage functions, 2 distinct functions (@gen, @sub), 4 metalanguage function arguments and 3 distinct arguments (num, {,}, {@sub[f]}). Therefore, its ND is equal to: $\frac{2 \cdot 4}{(3+4) \cdot (2+3)} = \frac{8}{35} = 0.23$. From the same values, we calculate that its CD is $\max(2,3,4) = 4$.

The values of the calculated complexity metrics for the *gate* metaprogram are summarized in Table 3.

Based on the metaprogram complexity metric values we can make the following conclusions on complexity of the *gate* metaprogram. The RKC value is high, therefore the metaprogram almost has no repeating fragments, it is coded at a meta-level efficiently and there is hardly room for any additional generalization without introducing new parameters or widening the scope of the metaprogram. The MR value shows that metalanguage constructs cover only about a half of the metaprogram's size, therefore, its understandability and readability is good. Following Frappier *et al.* [34], who introduce the

following boundaries of the CC values based on empirical research and practical implementations of large software systems: simple (1-10), slightly (moderately) complex (11-20), complex (21-50), over-complex and untestable (> 50), we conclude that due to large parameter space of the metaprogram, the exhaustive testability of its instances is complex. The CD value is below lower threshold (< 5) for short-term memorability of chunks of information as formulated by [29], therefore, cognitive complexity of the metaprogram is low.

Table 3. Complexity measures of the *gate* metaprogram

Complexity dimension	Complexity metric	Value
Information	Relative Kolmogorov Complexity (RKC)	0.90
Metalanguage	Metalanguage Richness (MR)	0.51
Graph	Cyclomatic Complexity (CC)	42
Algorithm	Normalized Difficulty (ND)	0.23
Cognitive	Cognitive Difficulty (CD)	4

Finally, we present complexity values calculated for Open PROMOL meta-programs created from Altera's library for OrCAD VHDL components (Table 4). Altera's library is a large collection of specific components, which are supposed to cover the entire circuit design domain (it contains 282 macro-functions and 73 primitives, i.e., 355 VHDL components at all). The components were generalized using Open PROMOL metalanguage to create a generic VHDL component library [35].

Table 4. Complexity of Open PROMOL components of generic VHDL library

No	PROMOL metaprogram	Complexity metric					Complexity
		RKC	MR	CC	ND	CD	
1	Serial multiplier	0.219	0.03	4	0.229	27	Simple
2	Trigger	0.271	0.27	80	0.111	52	Over-complex
3	Gate	0.502	0.49	181	0.026	26	Over-complex
4	Adder	0.478	0.25	4	0.169	20	Simple
5	Register	0.457	0.34	512	0.136	94	Over-complex
6	Multiplexer	0.507	0.36	32	0.051	22	Complex
7	Comparator	0.429	0.31	9	0.123	33	Simple
8	Shift Register	0.392	0.31	18	0.091	121	Moderate
9	Subtractor	0.378	0.20	4	0.072	84	Simple
10	Parallel multiplier	0.328	0.38	96	0.092	126	Over-complex
11	Register File	0.358	0.24	36	0.084	255	Complex
12	Counter	0.323	0.28	30	0.044	172	Complex
13	Multiplier	0.331	0.64	8	0.092	28	Simple
14	Divider	0.527	0.38	30	0.096	48	Complex

We evaluate the results presented in Table 4 as follows. Most complex metaprograms are those, which describe components with largest variability in the domain, thus requiring a larger number of parameters for selection of a specific instance and a larger number of metalanguage functions to represent their variability (see values of CC and CD metrics). Such metaprograms are difficult to test and maintain. Their complexity can be decreased by introducing hierarchical decomposition at the metaprogram level.

6.2. Homogeneous metaprogramming

As an example of complexity measurement of homogeneous metaprograms, we analyze Boost C++ Libraries [31]. Boost is a collection of open source libraries that extend the functionality of C++. To ensure efficiency and flexibility, Boost extensively uses C++ template metaprogramming techniques. In C++, the template mechanism provides a rich facility for computation at compile-time. Here we analyze complexity of template functions in a Boost.Math. This library several contributions in the domain of mathematics such as complex number and special mathematical functions. An example of such template function (a fragment) is presented in Fig. 3.

```

template<class T>
inline T fabs(const std::complex<T>& z)
{
    return ::boost::math::hypot(z.real(), z.imag());
}

```

Fig. 3. An example of template function (fabs)

The complexity measurement results using the proposed metaprogram complexity metrics are presented in Table 5.

Template functions in the Boost.Math library are rather simple. They mostly have CC values either 3, 16 or 19 meaning that each template function has a single template parameter, which can accept either 3 floating point, 16 integer or 19 floating point and integer C++ type values. Only `common_factor_ct` has template function `static_lcm`, whose template parameters are numbers of *long* type rather than types. All template functions also have the same ND value, because all template references are to the same template parameter class and have only one template parameter, therefore the number of distinct metaprogram operators and operands is equal to 1, and ND is equal to 0.25. The value of the CD metric is larger for components, which have a larger number of template references. The values of the RKC and MR metrics are larger for smaller components, which have less domain language (C++ non-template) code. When evaluating testability and maintainability of Boost.Math library components, the CD value could be used using the boundaries proposed by Frappier *et al.* [34] (see last column of Table 5.).

Table 5. Complexity of components of Boost.Math library

No.	Template function	Complexity metric					Complexity
		RKC	MR	CC	ND	CD	
1.	acos	0.229	0.037	3	0.25	34	Moderate
2.	acosh	0.488	0.128	3	0.25	3	Simple
3.	asin	0.218	0.037	3	0.25	34	Moderate
4.	asinh	0.488	0.144	3	0.25	2	Simple
5.	atan	0.423	0.116	3	0.25	10	Simple
6.	atanh	0.231	0.035	3	0.25	23	Complex
7.	bessel	0.149	0.157	3	0.25	127	Over-complex
8.	beta	0.134	0.090	3	0.25	199	Over-complex
9.	binomial	0.373	0.201	16	0.25	19	Moderate
10.	cbrt	0.438	0.144	3	0.25	15	Moderate
11.	common_factor_ct	0.171	0.042	1.9E19	0.25	13	Moderate
12.	common_factor_rt	0.160	0.043	3	0.25	32	Complex
13.	cos_pi	0.410	0.216	3	0.25	14	Moderate
14.	digamma	0.224	0.061	3	0.25	45	Complex
15.	ellint_1	0.242	0.133	3	0.25	40	Complex
16.	ellint_2	0.264	0.167	3	0.25	39	Complex
17.	ellint_3	0.213	0.116	3	0.25	48	Complex
18.	ellint_rc	0.391	0.137	3	0.25	17	Moderate
19.	ellint_rd	0.354	0.121	3	0.25	20	Moderate
20.	ellint_rf	0.363	0.127	3	0.25	22	Complex
21.	ellint_rj	0.320	0.105	3	0.25	24	Complex
22.	erf	0.210	0.055	3	0.25	88	Over-complex
23.	expint	0.234	0.038	3	0.25	128	Over-complex
24.	expm1	0.279	0.172	3	0.25	61	Over-complex
25.	fabs	0.666	0.150	3	0.25	1	Simple
26.	factorials	0.244	0.141	16	0.25	47	Complex
27.	fpclassify	0.146	0.086	19	0.25	98	Over-complex
28.	gamma	0.138	0.140	3	0.25	329	Over-complex
29.	hermite	0.421	0.208	3	0.25	14	Moderate
30.	hypot	0.396	0.211	3	0.25	22	Complex
31.	laguerre	0.260	0.164	3	0.25	30	Complex
32.	lanczos	0.225	0.070	3	0.25	638	Over-complex
33.	legendre	0.237	0.136	3	0.25	43	Complex
34.	log1p	0.204	0.112	3	0.25	91	Over-complex
35.	modf	0.293	0.233	3	0.25	10	Simple
36.	next	0.202	0.085	19	0.25	60	Over-complex
37.	octonion	0.030	0.013	19	0.25	599	Over-complex
38.	pow	0.239	0.196	3	0.25	42	Complex
39.	powm1	0.390	0.254	3	0.25	15	Moderate
40.	quaternion	0.059	0.029	19	0.25	348	Over-complex
41.	round	0.274	0.229	3	0.25	20	Moderate
42.	sign	0.516	0.155	19	0.25	5	Simple
43.	sin_pi	0.471	0.209	3	0.25	9	Simple
44.	sinc	0.215	0.098	3	0.25	36	Complex
45.	sinhc	0.224	0.084	3	0.25	30	Complex
46.	spherical_harmonic	0.210	0.150	9	0.25	50	Complex
47.	sqrt1pm1	0.468	0.238	3	0.25	8	Simple
48.	trunc	0.276	0.225	3	0.25	20	Moderate
49.	zeta	0.257	0.041	3	0.25	61	Over-complex

7. Conclusions and Future Work

Complexity of metaprograms and metaprogramming techniques is an important factor in developing and maintaining generic components and software generators. Complexity of metaprograms can be evaluated at several dimensions (information, metalanguage, graph, algorithm, cognition) using a variety of measures adopted from information theory and software engineering domain. Such metrics can be used to rank metaprograms based on their complexity values, to assess testability and maintainability of metaprograms, and can be used by reusable software library developers for evaluating complexity of their work artefacts. Despite the lack of larger-scale empirical validation, we still expect that metaprogram complexity metrics could be used to indicate poorly written or untestable metaprograms, when the metric values exceed predefined maximal or minimal boundaries.

Future work will focus on the empirical validation of proposed metrics using open metaprogram libraries implemented in different metalanguages.

References

1. Bennett, K.H., Rajlich, V.: Software maintenance and evolution: A roadmap. In: A.C. Finkelstein (ed.), *Future of Software Engineering*. ACM Press, 73-87, 2000.
2. Visscher, B.-F.: *Exploring Complexity in Software Systems*. Ph.D. thesis. Department of Computer Science and Software Engineering. University of Portsmouth, UK, June 2005.
3. Briand, L.C., Morasca, S., Basili, V.R.: Property-Based Software Engineering Measurement. *IEEE Trans. Software Eng.* 22(1): 68-86, 1996.
4. IEEE Computer Society: *IEEE Standard Glossary of Software Engineering Terminology*, IEEE Std. 610.12 – 1990.
5. Zuse, H.: *Software Complexity – Measures and Methods*. DeGruyter Publ., 1991.
6. Card, D.N., Agresti, W.W.: Measuring software design complexity. *The Journal of Systems and Software*, vol. 8, 185-197, 1988.
7. Reformat, M., Musilek, P., Wu, V., Pizzi, N.J.: Human Perception of Software Complexity: Knowledge Discovery from Software Data. In: *Proc. of 16th IEEE Int. Conf. on Tools with Artificial Intelligence (ICTAI 2004)*, Nov. 15-17, 202-206, 2004.
8. Blecker, T., Abdelkafi, N., Kaluza, B., Kreutler, G.: A Framework for Understanding the Interdependencies between Mass Customization and Complexity. In: *Proc. of the 2nd Int. Conf. on Business Economics, Management and Marketing*, Athens/Greece, June 24 - 27, 2004.
9. Misra, S., Akman, I.: A Model for Measuring Cognitive Complexity of Software. In: *Proc. of 12th Int. Conf. on Knowledge-Based Intelligent Information and Engineering Systems (KES 2008)*, Zagreb, Croatia, September 3-5, 2008, Part II. LNCS vol. 5178, pp. 879-886. Springer, 2008.
10. Mayrhauser, A.V., Vans, A.M.: From Code Understanding Needs to Reverse Engineering Tools Capabilities. In: *Proc. of 6th Int. Conference on Computer Aided Software Engineering (CASE'93)*, 19-23 July 1993, 230-239.
11. Damaševičius, R.: On the Quantitative Estimation of Abstraction Level Increase in Metaprograms. *Computer Science and Information Systems (ComSIS)*, 3(1): 53-64, 2006.

11. Keating, M.: Measuring Design Quality by Measuring Design Complexity. In: Proc. of the 1st Int. Symp. on Quality of Electronic Design (ISQED'2000), p. 103, 2000.
12. Software Engineering Institute (SEI). Software Technology Roadmap, 2006. Sipos, A., Pataki, N., Porkolab, Z.: On Multiparadigm Software Complexity Metrics. *Pure Mathematics and Applications*, 17(3-4): 469-482, 2006.
13. Card, D.N., Glass, R.L.: *Measuring Software Design Quality*. Prentice Hall, 1990. Rauterberg, M.: How to Measure Cognitive Complexity in Human-Computer Interaction. In: Proc. of the 13th European Meeting on Cybernetics and Systems Research, Vienna, Austria, 9–12 April 1996, Vol. 2, 815-820.
14. Sheetz, S.D., Tegarden, D.P., Monarchi, D.E.: Measuring Object-Oriented System Complexity. Proc. of. First Workshop of Information Technologies and Systems, 285-307. MIT Sloan School of Management, Cambridge, MA, 1991.
15. Halstead, M.H.: *Elements of Software Science*. New York: Elsevier, 1977. Henry, S.M., Kafura, D.G.: Software Structure Metrics Based on Information Flow. *IEEE Trans. Software Eng.* 7(5): 510-518, 1981.
16. Wang, Y.: On the Cognitive Complexity of Software and its Quantification and Formal Measurement. *International Journal of Software Science and Computational Intelligence* 1(2): 31-53, 2009.
17. Pataki, N., Sipos, A., Porkolab, Z.: Measuring the Complexity of Aspect-Oriented Programs with a Multiparadigm Metric. In: Proc. of European Conf. on Object-Oriented Programming - Quantitative Approaches in Object-Oriented Software Engineering Workshop (ECOOP-QAOOSE), Nantes, France, 1-10, 2006.
18. Pataki, N., Pocza, K., Porkolab, Z.: Towards a Software Metric for Generic Programming Paradigm. In: 16th IEEE Int. Electrotechnical and Computer Science Conference, 24-26 September 2007, Portorož, Slovenia.
19. Weyuker E.J.: Evaluating Software Complexity Measures. *IEEE Transactions on Software Engineering*, vol. 14, no. 9, 1357-1365, September, 1988.
20. Olabiyisi, S.O., Ayeni, R.O., Omidiora, E.O.: Comparative Study of Implementation Languages on Software Complexity Measures of Quick Sort Algorithm. *Journal of Engineering and Applied Sciences* 3 (1): 118-122, 2008.
21. Damaševičius, R., Štūkys, V.: Taxonomy of the Fundamental Concepts of Metaprogramming. *Information Technology and Control*, 37(2), 124-132, 2008.
22. Li, M., Vitanyi, P.: *An Introduction to Kolmogorov Complexity and Its Applications*. Springer Verlag, 1997.
23. Heering, J.: Quantification of structural information: on a question raised by Brooks. *ACM SIGSOFT Software Engineering Notes* 28(3): 6, 2003.
24. McCabe, T.J.: A Complexity Measure. *IEEE Transactions on Software Engineering*, vol. se-2, no. 4, 308-320, 1976.
25. Miller, G.: The Magic Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *The Psychological Review*, March, 1956.
26. El Emam, K.: *A Methodology for Validating Software Product Metrics*. National Research Council of Canada, Ottawa, Ontario, Canada NCR/ERC-1076, 2000.
27. Abrahams, D., Gurtovoy, A.: *C++ Template Metaprogramming: Concepts, Tools, and Techniques from Boost and Beyond*. Addison-Wesley Professional, 2004.
28. Misra, S., Akman, I.: Applicability of Weyuker's Properties on OO Metrics: Some Misunderstandings. *Computer Science and Information Systems (ComSIS)*, Vol. 5(1), 17-24, June 2008.
29. Štūkys, V., Damaševičius, R., Ziberkas, G.: Open PROMOL: an experimental language for target program modification. In: A. Mignotte, E. Vilar and L. Horobin (eds.), *System on chip design languages: extended papers: best of FDL'01 and HDLCon'01*. Netherlands: Kluwer Academic Publishers, 235-246, 2002.

30. Frappier, M., Matwin, S., Mili, A.: Software Metrics for Predicting Maintainability. Software Metrics Study: Tech. Memo. 2. Canadian Space Agency, 1994.
31. Damaševičius, R.: Scripting Language Open PROMOL: Extension, Environment and Application. MSc. Thesis. Kaunas University of Technology, Lithuania, 2001.

Robertas Damaševičius received his MSc (2001) and PhD (2005) degrees in informatics from Kaunas University of Technology (KTU), Kaunas, Lithuania. Currently he is an associated professor at Software Engineering Department, KTU. He teaches several computer science, programming and software engineering courses. He is also the member of Design Process Automation Group at Software Engineering Department. His research interests include hardware design, design automation, metaprogramming, software generation and program transformation, as well as domain analysis methods. He is an author of more than 50 scientific papers in the area.

Vytautas Štuikys is a professor at Software Engineering Department of KTU, Lithuania. He received the PhD and *doctor habilitatis* titles from KTU in 1970 and 2002, respectively. He is a lecturer and a researcher as well as a leader of the Design Process Automation Group. His research interests include software and hardware design methodologies, software reuse, component-based programming, metaprogramming and program generation, CAD systems and soft IP design. He has published more than 100 papers in the area. He is an author of several books and a monograph.

Received: March 15, 2009; Accepted: December 04, 2009.

An Evolutionary Solution for Multimodal Shortest Path Problem in Metropolises

Rahim A. Abbaspour and Farhad Samadzadegan

Surveying Department, University College of Engineering,
University of Tehran, Tehran, Iran
{abaspour, samadz}@ut.ac.ir

Abstract. This paper addresses the problem of time-dependent shortest multimodal path in complex and large urban areas. This problem is one of the important and practical problems in several fields such as transportation, and recently attracts the research focus due to developments in new application areas. An adapted evolutionary algorithm, in which chromosomes with variable lengths and particularly defined evolutionary stages were used, was employed to solve the problem. The proposed solution was tested over the dataset of city of Tehran. The evaluation consists of computing shortest multimodal path between 250 randomly selected pairs of origins and destination points with different distances. It was assumed that three modes of walking, bus, and subway are used to travel between points. Moreover, some tests were conducted over the dataset to illustrate the robustness of method. The experimental results and related indices such as convergence plot show that the proposed algorithm can find optimum path according to applied constraints.

Keywords: multimodal shortest path, genetic algorithm, metropolis.

1. Introduction

Finding optimum path between two designated points is one of the classic and fundamental problems in the network analysis [6, 13]. According to literature, this problem and its variations have been extensively investigated in different fields. One of the most interesting application domains is transportation in which the problem in various forms including classic, constrained, dynamic, and multi-criteria can be recognized [29]. Moreover, recent developments in intelligent transportation system (ITS), Location-based Services (LBS) and route advisory systems have led many researchers to concentrate further on different aspects of problem [2, 6, 9, 15, 22, 35, 36].

Multimodal transportation systems are public, ordinary networks in urban areas, particularly in metropolises where the citizens may utilize the combinations of several modes of transportation such as personal car, taxi, subway, tram, bus, and walking. Using these networks brings real benefits for

citizens by saving their time and cost, and also greatly assists sustainable development of metropolises with reduction of traffic jams and air pollutions.

Multimodal shortest path problem is concerned with finding a path from a specific origin to a specific destination in a given multimodal network while minimizing total associated cost. The complexity of finding multimodal route is obviously much higher than monomodal one. In the multimodal networks several modes of transportation operate concurrently under the changing conditions. Since using a multimodal network requires information about timetables of vehicles, it is necessary to have some information about departure time of travel to solve the problem. Due to this nature of these networks, they can be considered as dynamic networks in which travel time associated to each arc changes with time [21, 37].

A novel formulation and evolutionary-based solution is proposed in this paper to compute single-objective shortest path on multimodal networks, taking both travel and switching time into account. It is assumed that the multimodal network consists of three modes of walking, bus, and subway and the arcs of network have time-dependent weights. In the remainder of this paper, in section 2, the related researches are reviewed. Section 3 gives a detailed description and formulation of the multimodal shortest path problem, and moreover, states proposed solution to the problem in detail. In Section 4, the solution is examined as a case study, and the results are evaluated to show the proficiency of proposed method. Finally, section 5 concludes with the extensions of the proposed solution.

2. Review of Related Research

There have been many research on computing multimodal shortest paths. A considerable portion of them is about finding solutions for static multimodal shortest path problem [12, 27, 28, 34]. Most of these algorithms approximate waiting time of transfer according to headway, or using generating of artificial waiting time arcs [11]. But, recent researches have changed the direction towards working on dynamic solutions.

Battista *et al.* [7] introduced a heuristic approach called *path composition approach* for finding reasonable routing rules on bimodal networks based on what a user probably chooses. They used these composition rules to concatenate small number of good sub-paths to create the shortest path.

Modesti and Sciomachen presented an approach based on the classical shortest path problem for finding multi-objective shortest paths in urban multimodal transportation networks with objectives of minimizing the overall cost, time, and users' discommodity associated with the required paths. They used an *ad hoc* utility function to assign the weights to arcs according to their cost and time while simultaneously considered the preferences of the users related to all of the possible transportation modalities [26].

The temporal intermodal optimum path algorithm, which was proposed by Ziliaskopoulos and Wardell according to the principles of dynamic

programming, took the delays at modes, and arc switching points into account. Their Time-Dependent Intermodal Least Time Path (TDILTP) algorithm was based on a general framework in which the routes from all origins, departure time, and modes to a destination node are computed [37]. The TDLITP algorithm operates in a label-correcting manner, assuming that dynamic travel time of arcs and actual schedules of transit lines are known. Chang *et al.* extended the TDILTP algorithm to calculate Time-Dependent Intermodal Minimum Cost Paths (TDIMCP) from all origins and departure time to a destination node, based on time-dependent, fixed travel and transfer costs [11]. Despite the time-invariant problems, this algorithm considered time as an index, which determines the feasibility of a path, while a separate cost function is optimized.

Lozano and Storchi proposed a multimodal shortest path algorithm to find the shortest viable path, in which paths with illogical sequences of used modes were eliminated [23]. They used label correcting techniques and an *ad hoc* modification of the Chronological Algorithm (Chrono-SPT) which was proposed by Pallottino and Scutella [29] to solve the problem. They extended their algorithm to calculate the viable hyperpath [24].

Abdelghany and Mahmassani proposed an algorithm to calculate intermodal paths based on a multi-objective shortest path algorithm, where the set of non-dominated paths was computed and an optimum path is selected among them based on a generalized cost function[1].

Another research was conducted by Sherali *et al.* in the Route Planner Module of Transportation Analysis Simulation System (TRANSIMS) [33]. They proposed a dynamic programming-based approach for the time-dependent, label-constrained shortest path problem by adapting existing partitioned shortest path algorithmic schemes.

Bielli *et al.* proposed a framework to address both algorithmic approaches proposed for solving the multimodal shortest path problem and a transportation network modeling using geographic information systems (GIS) [10]. They developed a modified version of k-shortest path algorithm to define an efficient solution for multimodal shortest path with time constraints.

A new graph structure, namely *transfer graph*, was introduced by Ayed *et al.* to model the multimodal networks [4]. A transfer graph is described by a set of sub-graphs called components. They proposed an algorithm to deal with multi-objective route guidance problem in the time-dependent multimodal network. The advantage of their algorithm was the capabilities to find the multimodal route when all involved networks are kept separated, which imply to be accessed separately.

Qu and Chen proposed a hybrid multi-criteria decision making method combining the fuzzy Analytic Hierarchy Process (AHP) and Artificial Neural Network (ANN) to find the multimodal, multi-criteria paths [31]. They used fuzzy AHP to find the suitable initial input-hidden weights to improve the efficiency of ANN. The improved ANN with error back-propagation is applied to study the relationships between criteria and the alternatives performance. They used this approach to find the best way from a certain origin to a specific

destination through a multimodal transportation network according to six main criteria, which was extracted among 15 criteria.

In addition to these researches, some practical systems have been developed, and most of them are in the form of multimodal route planners, which include modules to find the shortest multimodal path for users. Some of these systems are [8, 16, 20, 25, 30, 32].

The major weakness of these researches is that they were not carried out on real dataset of any metropolises that include complex transportation networks. In these cities, search spaces are too large and highly complex. Thus, the problem will be too complex to be solved by traditional methods, and efficient optimization strategies are required to deal with this difficulty.

3. Proposed Solution for Time-dependent Shortest Path

An evolutionary-based framework has been developed in this paper to find the optimum solution for the multimodal shortest path problem (Fig. 1). The framework consists of two main parts. One part is the multimodal shortest path module, in which some parameters such as origin and destination, start time, and transportation modes are introduced to the engine. The engine of this module, which works based on an adapted evolutionary algorithm, computes the optimum path according to the input parameters and a geodatabase. It is possible for user to introduce his/her contextual information such as age and health level to engine of this module to have more suitable solutions. The other part of the framework is geodatabase. This part includes all necessary information both in spatial and attribute format, which broadly are divided into two groups of pathways and SST.

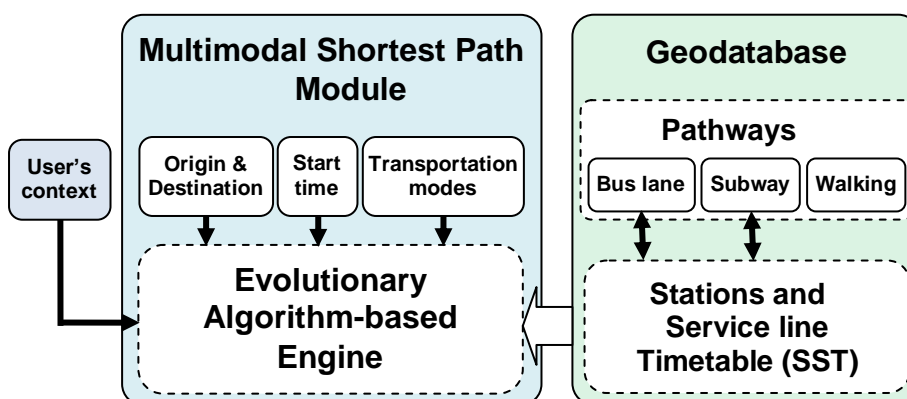


Fig. 1. Proposed framework for multimodal shortest path finder

The remainder of this section consists of three parts to explain the proposed solution in details. An introduction to evolutionary algorithms is described in section 3.1. Then, the basic formulation of the problem is

discussed in section 3.1, and proposed solution for finding shortest paths in multimodal networks are described in sections 3.3.

3.1. Evolutionary Algorithm

The evolutionary algorithm is a class of optimization methods that simulate the process of natural evolution [17, 18, 19]. Evolutionary computing comprises genetic algorithm (GA), genetic programming, evolutionary programming, evolutionary strategy, and classifier systems [5]. This algorithm is also a member of a group of methods, known as meta-heuristics. This set of techniques includes simulated annealing method, tabu search method, ant colony algorithm, bee algorithm, particle swarm optimization, artificial immune systems, and distributed reinforcement learning, and has been proposed to solve the difficult possible optimization problems [14].

Genetic algorithms (GAs) were invented by John Holland in the 1960s [19] and were developed by Holland, his students, and colleagues at the University of Michigan in the 1960s and the 1970s [17]. The general algorithm of this approach is as follows:

```
Algorithm 1: Evolutionary Algorithm
  initialize population
  evaluate population
  do while (termination-criteria is not satisfied)
    select parents for reproduction
    perform recombination and mutation
    evaluate
  loop
```

GA is different from other heuristic methods in several ways. The most important difference is that GA works on a population of possible solutions, while the other heuristic methods use a single solution in their iterations. The general acceptance is that GA is particularly suited to multidimensional global search problems where the search space potentially contains multiple local minima [18]. The basic GA does not require extensive knowledge of the search space, such as likely solution bounds or functional derivatives. Moreover, GA has a number of other advantages, some of them are (i) the concept is easy to understand, (ii) modular, separate from application, (iii) supports multi-objective optimization, (iv) easy to exploit previous or alternate solutions, (v) there is always an answer, (vi) answer gets better with time, (vii) ability to scan a vast solution set quickly, (viii) bad initial population do not influence the end solution negatively, (ix) are useful and efficient when the search space is large, complex, or poorly understood. According to these considerable advantages of GA, this approach may be the first candidate when someone wishes to solve an optimization problem. Further information on genetic algorithms can be found in [3, 18].

3.2. Basic Formulation of Problem

In contrast with finding monomodal paths, computing the shortest multimodal path may encounter some difficulties. The algorithm should take into account all service lines and transportation modes that pass one station according to their temporal schedules. Furthermore, it is necessary to model the waiting time when the services/modes are changed. The differences between a monomodal and multimodal path are illustrated through an example. As depicted in Fig. 2, it is assumed that there are three service lines and a number of bus stops. Each service line, SL_i , is represented by the sequence of stops. For example, $SL_3 = (\dots, 301, 102, 103, 202, 203, 401, \dots)$ shows the service line with ID number 3 that passes through the stations mentioned. Each line has a departure timetable and it is possible to build a timetable for each of the stations according to the temporal distance of preceding stations and the line number. If the non-temporal monomodal path from O to D is represented by $Path(O,D) = (O, 101, 102, 103, 202, 203, 204, 205, D)$, then this sequence may coincide with different multimodal paths such as the following:

- Walking from O to S_{101} , waiting for the bus at S_{101} , going to S_{103} by SL_1 , walking from S_{103} to S_{202} , waiting for the bus at S_{202} , going by SL_3 to S_{205} , and finally walking to D .
- Walking from O to S_{101} , waiting for the bus at S_{101} , going to S_{102} by SL_1 , waiting to change to another bus, going by SL_2 to S_{203} , again waiting for a bus change, then going by SL_3 to S_{205} , and finally walking to D .
- Walking from O to S_{102} , waiting for the bus at S_{102} , going to S_{203} by SL_2 , waiting to change the bus, going by SL_3 to S_{205} , and finally walking to D .

Obviously, each of these paths may have a different corresponding total cost.

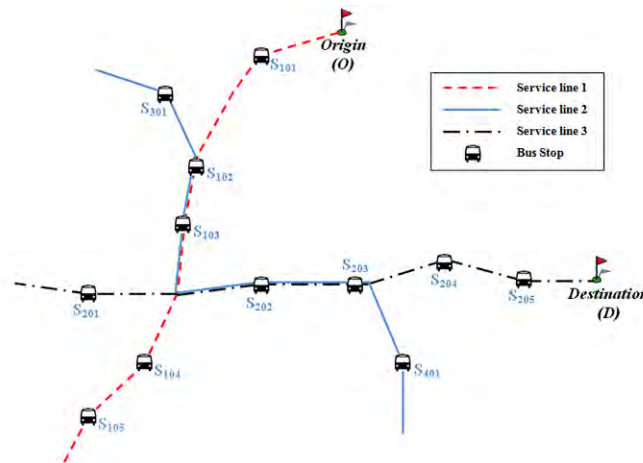


Fig. 2. Sample service lines in a multimodal network

Let the multimodal transportation network be assumed as a directed labeled multigraph of $G(N, A, L_N, L_A, M_N, M_A, \psi, \omega)$. $N = N_B \cup N_S$ is the set of nodes consisting of bus stops (N_B) and subway stations (N_S). A is a multiset of arcs comprises different passes of service lines and walking paths. L_N is a set of node labels showing associated IDs of stops (stations), and L_A is a set of arc label showing the service line IDs and departure orders for buses and subways. $M_N: N \rightarrow L_N$ is the node labeling function associates a label from L_N with each node. $M_A: A \rightarrow L_A$ is the arc labeling function that associates a label from L_A with each arc. $\psi: A \rightarrow N \times N$ is the incidence function which associates a pair of nodes from N with each edge in A . $\omega: A \times \mathbb{N} \cup \{0\} \rightarrow \mathbb{R}^+$ is the weight function, which associates a weight with each arc in A . The weights of arcs represent the time duration required to travel between two nodes with specific departure time.

The cost of each multimodal path consists of two main parts of traveling and waiting time. Traveling time shows the duration of traveling, and waiting time represents the periods when someone changes the service line(s) of transportation. The time it takes to switch between bus and subway mode consists of two parts of walking and waiting. Walking, which is simulated as the connecting arc, is added to travel time. Consequently, as the following formulation demonstrates, the objective function of finding a multimodal shortest path between an origin (O) and a destination (D) consists of two general parts of minimizing the weights of used arcs (first summations) and minimizing the waiting time (second summations) when the modes are changed.

$$\min \sum_{l \in L_A} \sum_{i=0}^D \sum_{j=0}^D x_{ij}^l \omega_{ij}^l + \sum_{\substack{l, l' \in L_A \\ l \neq l'}} \sum_{\substack{i=0 \\ i \neq j, k}}^D \sum_{\substack{j=0 \\ j \neq i, k}}^D \sum_{\substack{k=0 \\ k \neq i, j}}^D (\tau_{jk}^{l'} - (\tau_{ij}^l + \omega_{ij}^l)) \quad (1)$$

Subject to:

$$\sum_{\substack{l \in L_A \\ j=0 \\ j \neq i}}^D x_{ij}^l - \sum_{\substack{l \in L_A \\ j=0 \\ j \neq i}}^D x_{ji}^l = \begin{cases} -1 & i = D \\ 0 & i \neq D, O \\ 1 & i = O \end{cases} \quad (2)$$

$$x_{ij}^l \in \{0, 1\} \quad (3)$$

In this optimization function, x_{ij}^l is a binary variable associated with each arc (i, j) with label l and equal to 1 if and only if the corresponding arc is used in the solution. ω_{ij}^l and τ_{ij}^l show the travel time (weights) and departure time of arc (i, j) with label l , respectively. Constraint (2) guarantees that the result is a path and there is no loop.

A *stations-service lines timetable (SST)* is used to store the essential information about the transportation network. This table is a part of the geodatabase, and its structure is shown in Table 1. The SST consists of four fields of *service line ID*, *from-station ID*, *to-station ID*, and *departure time*. Each row of this table shows the time when the bus/subway with *service line ID* passes *from-station* heading towards *to-station*. *Service line ID* column indicates the mode, order, and ID of a service in the form of a number. Obviously, different service lines may share the same stops/stations. The information in this table is also used to extract adjacency relationships between stations, and to find the neighbors of a specific station. Another advantage of this table is the possibility of modeling all effective factors on temporal distances between two nodes, such as congestion, as time differences.

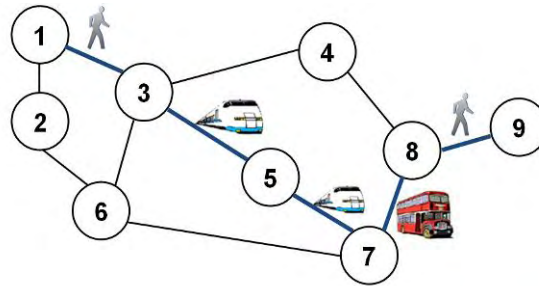
Table 1. Stations and Service lines Timetable (SST)

Service line ID	From-station ID	To-station ID	Departure time
1001	101	102	06:30
1001	102	103	06:40
1002	102	103	06:42
...
2097	1520	1547	18:45
...

3.3. The Engine of Proposed Multimodal Shortest Path Finder Module

The engine of proposed module, which is based on an adaptive evolutionary algorithm, works in five steps.

Coding of the chromosomes and initialization are the first step. Because the number of nodes for a path is not predefined in this problem, the chromosome with variable length is used to show a path in the network. The values of odd genes show the labels (IDs) of nodes. The values of even genes represent the transportation modes between two successive nodes where number 1 and service line IDs are used for walking and the other modes, respectively. The position of a node represents the order of that node in a path. Fig. 3 shows an example of the encoding. The randomly generated genes (nodes) are appended sequentially to construct a chromosome (path). When the population reached the population size (num_{pop}), the cost of each chromosome is calculated using the SST. Velocity of walking, which actually depends on the context of the user, is assumed approximately an average amount of 4 Km/h. Because large initial population provides the algorithm with a comprehensive sampling of search space; thus, the initial population size is assumed larger than the population size in the other iterations (num_{ipop}).



Chromosome (path):

1	1	3	2	5	2	7	5	8	1	9
---	---	---	---	---	---	---	---	---	---	---

Fig. 3. A path (weighted lines) in a network and its related encoded chromosome

The length of a chromosome depends on the number of intermediate nodes that each path passes. To generate the initial population of the chromosomes, this process is repeated until the predefined size of population is achieved. Algorithm 2 (*Initialization*) shows the procedure.

```

Algorithm 2: Initialization (O, D, G, SST, numpop)
// O is the origin and D is the destination point
// Ai = {adjacent nodes of node i extracted from SST}
// Scan = {scanned nodes}
// chrom is a (population-size × n) array where the
// generated chromosomes are stored
// modei,j = {possible transportation modes between
// two successive nodes i and j}
counter ← 1
Do While (counter ≤ numpop)
    previous ← 0
    Do While (current ≠ D)
        chrom(counter,1) = 0
        current ← Random (Aprevious)
        mode ← Random (modeprevious,current)
        add current to Scan
        append mode and current to chrom(counter)
        previous ← current
    Loop
    remove the contents of Scan
    counter ← counter+1
Loop
Return (chrom)
    
```

In the next step, i.e. *natural selection*, the fittest chromosomes survive, and the remaining ones are discarded from the population. The selection rate (C_{rate}) which is usually arbitrary, is used to show the fraction of surviving chromosomes (num_{keep}). It is common to keep half of the population in each

iteration, i.e. $C_{rate}=50\%$. This rate is used in this procedure. The kept chromosomes form the mating pool.

$$\text{num}_{\text{keep}}=C_{\text{rate}}\times\text{num}_{\text{pop}} \quad (4)$$

The third step is *selection* in which two chromosomes are selected from the mating pool to produce two new offspring. There are several selection methods, including random pairing, weighted random (roulette wheel) pairing, and tournament selection [18]. The roulette wheel pairing is used in this study. In this method, the probability assigned to a chromosome is inversely proportional to its cost [18].

Mating is the fourth step in which offspring are created from parents selected in the previous step. It is common to produce two offspring from mating two parents using randomly selected crossover point(s) on the parents' chromosomes. In the proposed algorithm, a combined method is used, in which both single-point and two-point crossovers are used. The procedure continues until $\text{num}_{\text{pop}}-\text{num}_{\text{keep}}$ offspring are born to replace the discarded chromosomes. Furthermore, an elitism strategy is also adopted. Elitism guarantees that the best individuals in a population survive into the next generation. In our solution, two chromosomes are kept as elite. After this step, the number of chromosomes becomes again equal to population size.

Because a chromosome shows a path as the sequences of stations connecting transportation modes, changing the value of any genes during crossover may cause the chromosome to show invalid paths. To cope with this issue, a rectification procedure was utilized to ensure that newly generated chromosomes have no loops, and show valid paths. This procedure removes some genes, which are inside a loop, to rebuild a valid path.

To avoid locally optimal solutions and prevent the algorithm from converging quickly before sampling the entire cost surface, the random *mutation* is used in the next step. In proposed solution, the mutation is applied only to odd genes. The even genes (transportation modes) are modified according to odd genes. The modification procedure is also applied after mutation. Then, the last check is carried out for avoiding loops along the path. The mutation procedure is illustrated in Fig. 4.

Both crossovers and mutations are applied to chromosomes with predefined probability levels. The probability function has uniform distribution. The costs associated with offspring and mutated chromosomes are again computed and assigned.

The whole process, i.e. step 2 to step 5, is iterated until the temporal differences between twenty best cost paths reach zero in successive iterations. However, if the algorithm does not stop according to specified criterion, the process is set to terminate after 500 iterations.

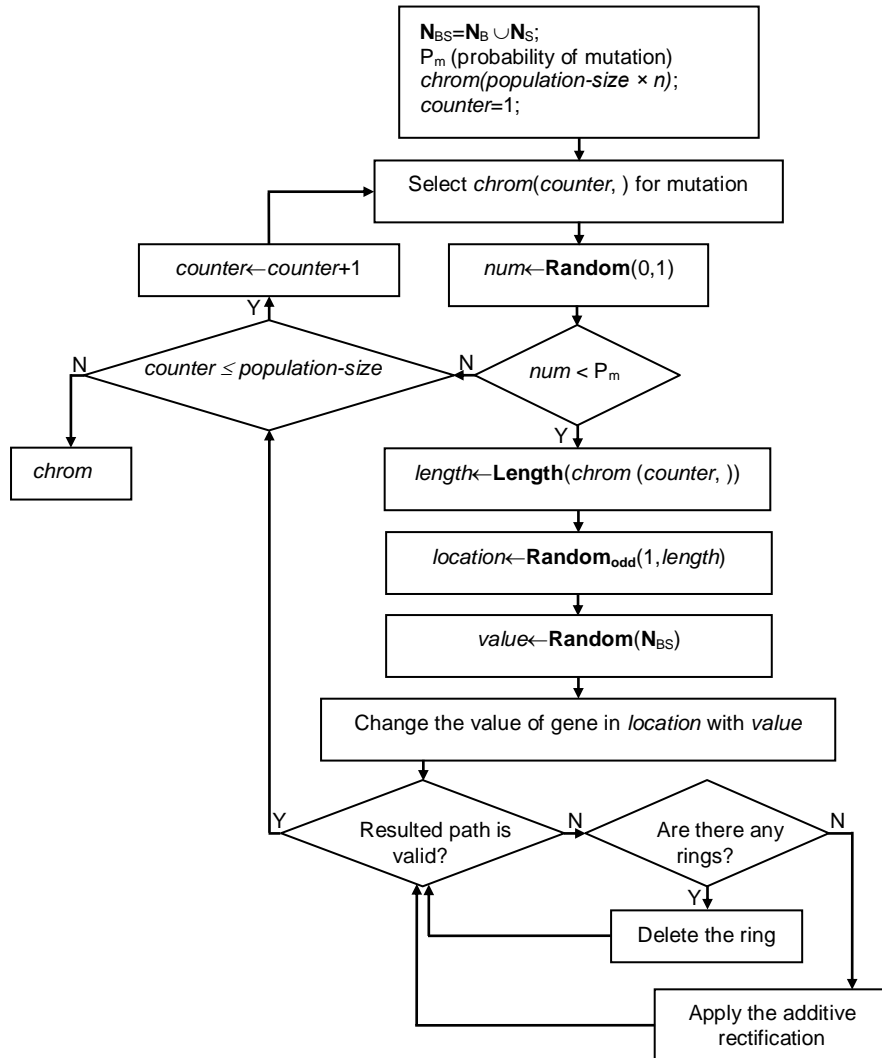


Fig. 4. Flowchart of the mutation procedure

4. Experimental Results

The proposed framework and formulation is evaluated by conducting some experiments using data from the city of Tehran, which is the capital city of Iran. Tehran is one of the metropolises of Iran with area of about 700 square kilometers and composed of 22 districts. Public transportation in this city

employs five main modes: taxi, van, minibus, bus, and subway. Among these, subway, due to its high speed, and bus, due to its extensive coverage, are of greatest interest for travelers and commuters. Table 2 presents some statistics of these two modes.

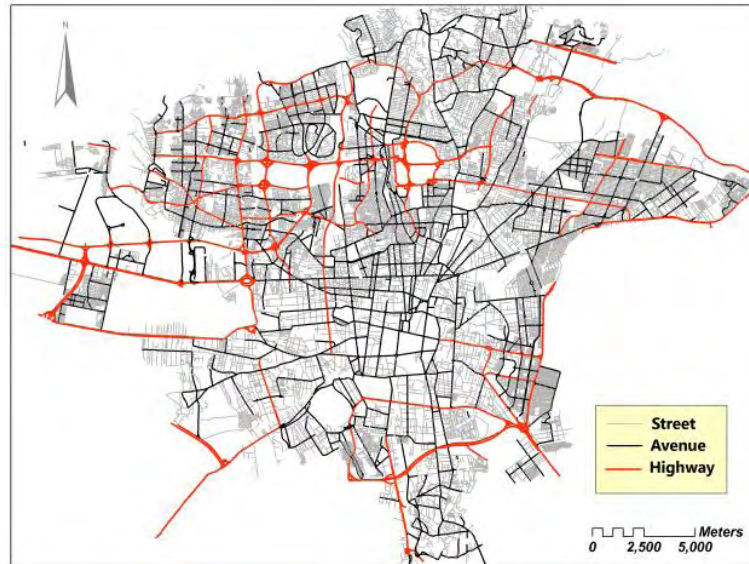
Table 2. Information about bus and subway transportation networks in Tehran

Property	Subway	Bus
Number of service lines	4	301
Number of stops (stations)	45	4763
Length of line (Km)	91.6	2761.3

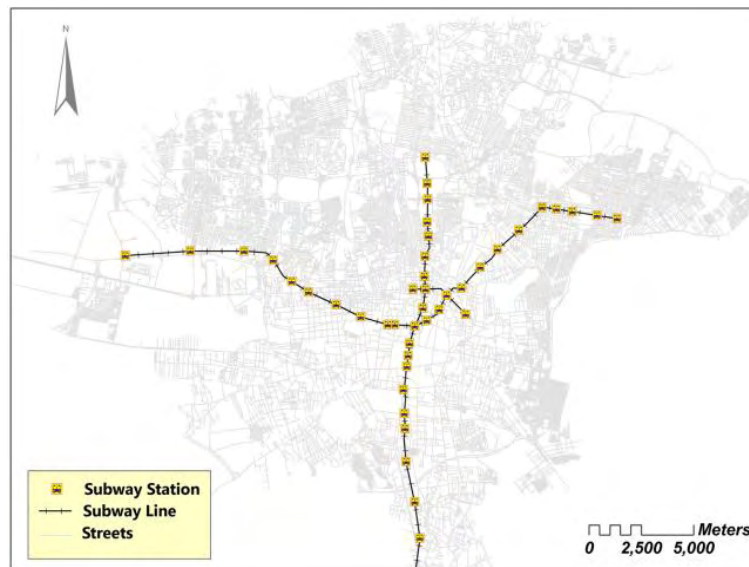
As shown in Fig. 5, the data set, which is the main body of geodatabase, consists of pathways (i.e. streets, avenues, and highways used by buses), subway system (consisting of lines and stations), and bus stops. The dataset is prepared as needed, i.e. the topological structure is rebuilt and the required tables are created.

To evaluate the multimodal shortest path algorithm, 500 points (250 source-destination pairs) with different number of nodes and start time were selected. They were used as the input origin and destination of the algorithm.

The proposed algorithm starts by generating a large population of chromosomes as the initial population. For the initial population it is inferred that $num_{ipop}=250$ is a good choice. Then, the cost of each chromosome is calculated based on the fitness (objective) function, and chromosomes are sorted in descending order according to the assigned cost. In the next step, $num_{ikeep}=num_{pop}=100$ best chromosomes, which had lower costs, are kept for the successive iteration and the others are discarded. It is realized that $num_{pop}=100$ is a suitable value according to our experience. In the next iterations, $C_{rate}=50\%$ is selected. The next steps were natural selection, mutation and checking termination criteria. It was found that the best range for crossover and mutation probabilities are $[0.55, 0.85]$ and $[0.07, 0.18]$ respectively for this dataset. No significant differences in the results were observed within these ranges. Hence, the fixed values of 0.7 for crossover probability and 0.1 for mutation probability are adopted. Four different cases are discussed to illustrate some samples of the results obtained. The first case is a considerably longer path and is shown over the whole extent of the dataset, while the other three cases represent the local paths.



(a)



(b)

Fig. 5. Dataset of evaluation of proposed methodology (a) pathways and (b) subway

The first case is representative of those paths that are considerably long. The results show that the paths in this class are generally multimodal. Fig. 6 shows one of the resulted paths on the map. As shown in Fig. 6, the result of finding a shortest path between a source and destination is a multimodal path. It begins by walking from the source to the nearest bus stop, using the bus to reach the subway station, walking from the bus stop to the subway station, using two different lines of the subway (walking to change lines), walking from the subway station to the nearest bus stop, using the bus to stop nearest to the destination, and finally walking to the destination.

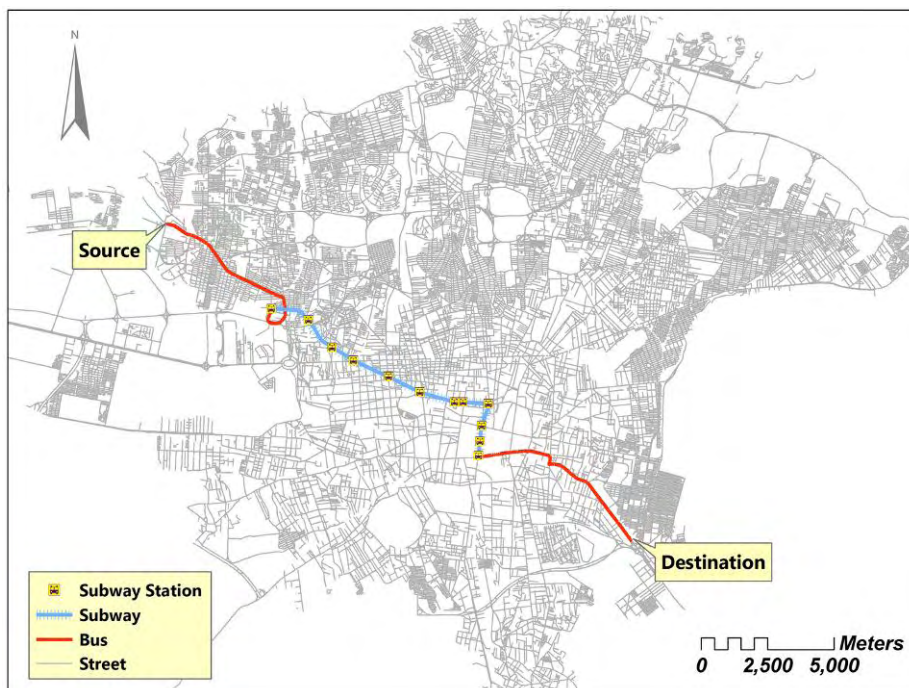
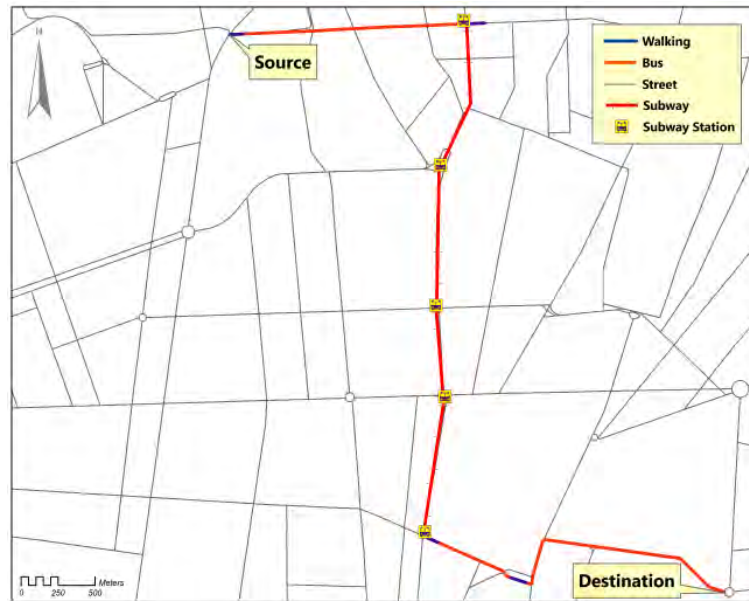


Fig. 6. Result for case 1

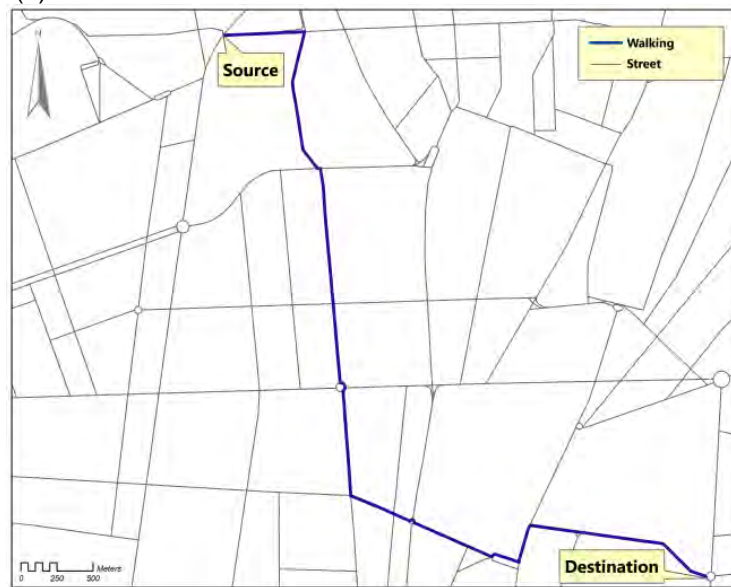
The convergence curve plot, which shows the cost of the minimum cost path as a function of iterations, is adopted to represent the results of iterations for this case (Fig. 8.a). As shown by the convergence plot, there is a rapid decrease of the fitness values in the first few generations. The curve also indicates that the process successfully reaches convergence.

In the second case, the result for the multimodal shortest path is also a combination of all three modes (Fig. 7.a). Different compositions of shortest paths are generated using individual modes to compare the results with monomodal transportation. It can be seen that traveling from the selected source to the destination by using only bus or subway is impossible. The result for shortest path by walking between the source and destination is

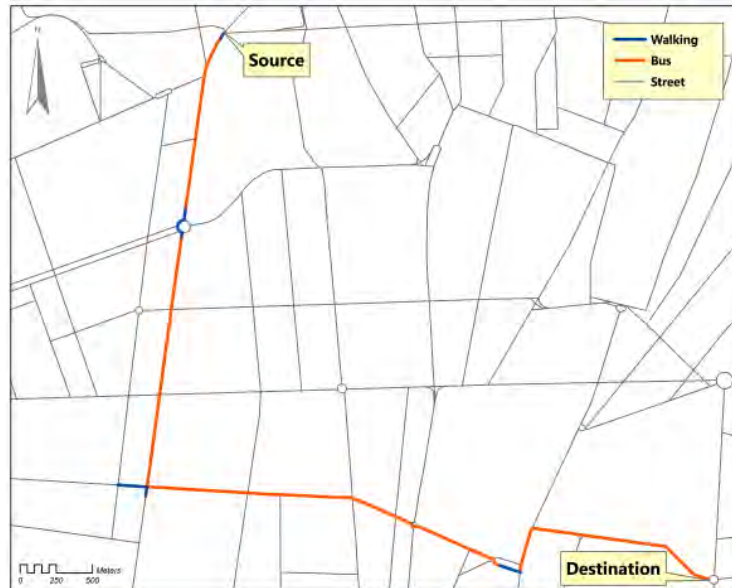
depicted in Fig. 7.b. Finding the possible shortest path using a combination of two or three modes is also examined. There are no paths composed of a combination of bus and subway modes. For the other two possible situations, i.e. walking-bus and walking-subway, the results are depicted in Fig. 7.c and 7.d, respectively.



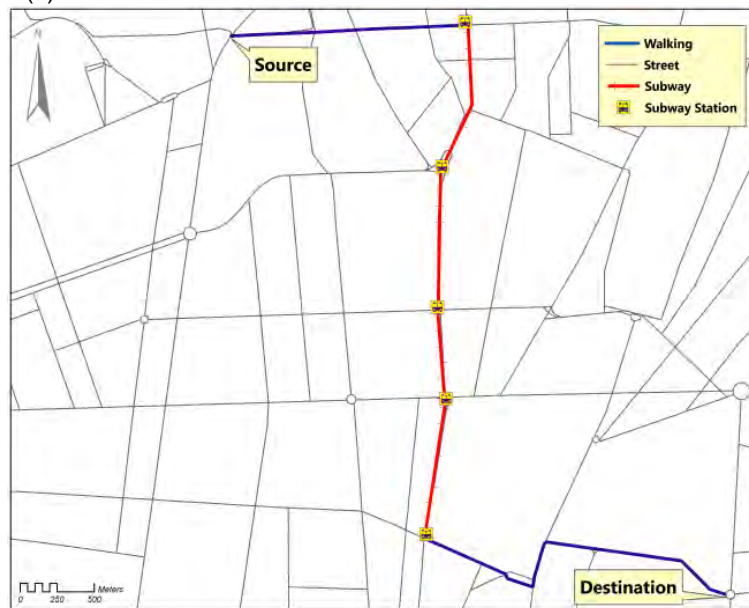
(a)



(b)



(c)



(d)

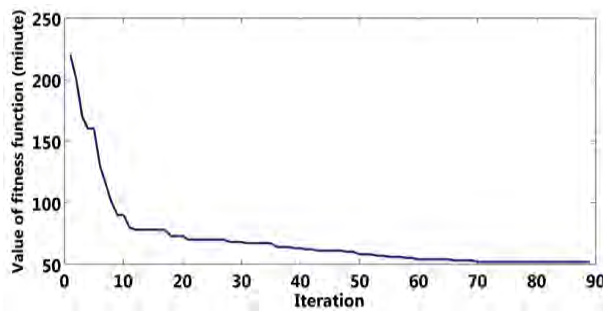
Fig. 7. Results for case 2

The minimum time of the resulting shortest paths using the different combinations of modes are given in Table 3. According to this table, the minimum cost belongs to multimodal combination.

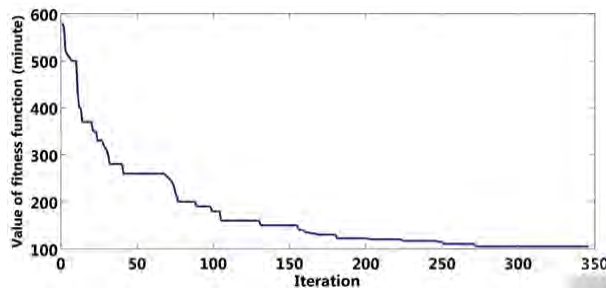
Table 3. Cost for different combinations of modes for a given source and destination

No. of Mode(s)	Combination	Cost of Shortest Path (Minutes)
Monomodal	Walking	125
	Bus	-
	Subway	-
Dual modal	Walking and Bus	83
	Walking and Subway	69
	Bus and Subway	-
Multimodal	Walking, Bus and Subway	52

The convergence curve plot is also adopted for this case to show the results of iteration, and the convergence of the process (Fig. 8.b). The pattern and general trend of convergence in this case is similar to the first case. This behavior is observed in all cases evaluated.



(a)

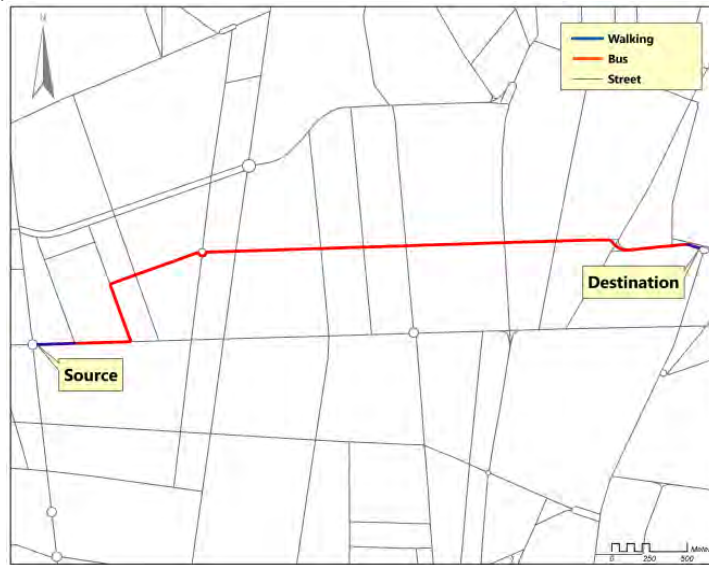


(b)

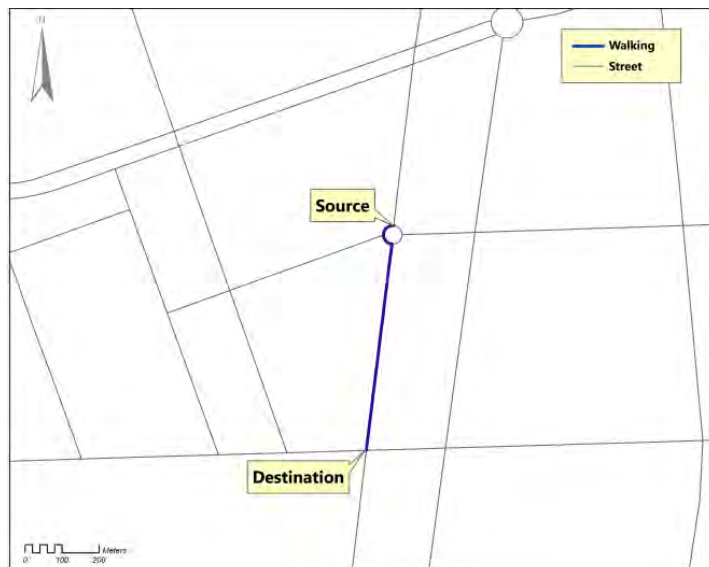
Fig. 8. Convergence plot of shortest paths of first case (a), and second case (b)

In the third case, the result for multimodal shortest path is a combination of walking and bus modes (Fig. 9.a). It is perceived that this result has the least

cost compared with other combinations. In the fourth case, the result for multimodal shortest path is just walking between the source and destination (Fig. 9.b).



(a)



(b)

Fig. 9. Results for case 3 (a) and case 4 (b)

To measure the quality of proposed method, the failure ratio is utilized. This ratio shows the times that an evolutionary algorithm fails to find the global optimum in respect of whole runs. This index was calculated for each of 250 paths, and shows the frequency of paths which fails to be optimum among 30 runs for each of the cases.

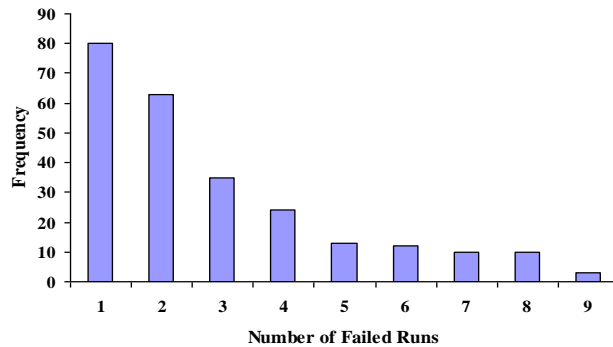


Fig. 10. Failure rates of calculated paths

Fig. 10 shows this index for the evaluated data set. The horizontal axis of this diagram shows the numbers of failed runs among the 30 runs, and vertical axis illustrates the frequency of number of failures among all cases. It was perceived that the average failure rate of proposed path finding algorithm is about 2.77. Therefore, the quality of solution and path optimality is about 90.75%.

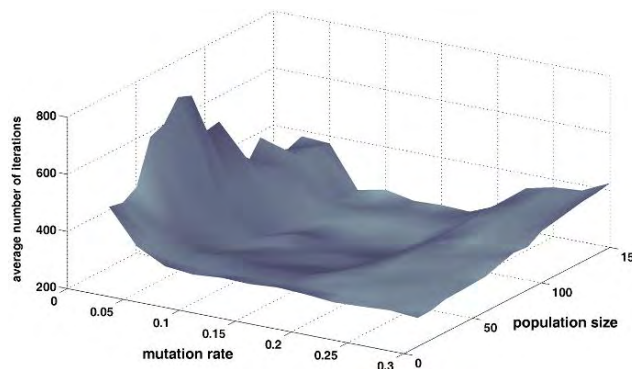


Fig. 11. The correlation between population sizes and mutation rates with average number of iterations

To evaluate the correlation between population size and optimal solution, and to find the suitable population size and mutation rate needed to obtain the acceptable solutions, the algorithm were conducted over 100 different paths for 15 different population sizes and 12 different mutation rates. Each combination was run 20 independent times. Population size varied from 20 to

150 in increments of 10. Mutation rate varied from 0 to 0.3 in increments of 0.025. A three-dimensional plot of the average number of iterations for the different values of population size and mutation rate is depicted in Fig. 11. As shown in this figure, the algorithm works best with population size about 100 and mutation rates interval about 0.1. It was found from this experiment that when the population size increases, the optimal mutation rate decreases.

5. Conclusion

In this paper, the possibility of using the genetic algorithm to solve the dynamic shortest path problem in urban multimodal transportation networks is investigated. The proposed approach has been tested on a dataset of a part of Tehran. 250 pairs of points, selected randomly as the source and destination to evaluate the algorithm, and to tune some parameters of it. The results were divided into three main groups where path consists of one, two or three modes of transportation. These show that the multimodal path is not essentially the shortest one in all cases. Moreover it is concluded that proposed algorithm has high degree of robustness since it covers monomodal solutions as the special case of multimodal paths. The high average amount of success rate of algorithm shows the high performance of it.

Several topics remain for future research. Since some of the decisions made by citizens to select their ways to transport between two points are not just the function of time; thus, extension of proposed algorithm to address the multi-criteria path finding problem is important. The second activity may be related to consider the real-time implementation of algorithm which in turn needs some modifications to improve the speed and management of on-line input parameters. To develop a useful and practical path finding module for a user, considering his/her contextual information is critical. These parameters modify the selected path to adapt the user situations as much as possible.

References

1. Abdelghany, K.F., Mahmassani, H.S.: Dynamic Trip Assignment-simulation Model for Intermodal Transportation Network. *Transportation Research Record*, Vol. 1771, 52-60. (2001)
2. Aifadopoulou, G., Ziliaskopoulos, A., Chrisohoou, E.: Multiobjective Optimum Path Algorithm for Passenger Pretrip Planning in Multimodal Transportation Networks. *Transportation Research Record*, Vol. 2032, 26-34. (2007)
3. Ashlock, D.: *Evolutionary Computation for Modeling and Optimization*. Springer Science+Business Media, New York, USA. (2006)
4. Ayed, H., Khadraoui, D., Habbas, Z., Bouvry, P. & Merche, J. F.: Transfer Graph Approach for Multimodal Transport Problems. In: Hoai, L. T., Bouvry, P., Tao, P.D. (eds.): *Modelling, Computation and Optimization in Information Systems and Management Sciences*. Communications in Computer and Information Science, Vol. 14. Springer, Berlin Heidelberg, 538-547. (2008)

5. Bäck, T., Hammel, U., & Schwefel, H. P.: Evolutionary Computation: Comments on the History and Current State. *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, 3–17. (1997)
6. Barrett, C.L., Bisset, K., Jacob, R., Konjevod, G., Marathe, M.V.: Classical and contemporary shortest path problems in road networks: Implementation and experimental analysis of the TRANSIMS router. In: Möhring, R.H., Raman, R. (eds.) *Lecture Notes in Computer Science*, Vol. 2461. Springer Heidelberg 126–138. (2002)
7. Battista, M.G., Lucertini, M., Simeone, B.: Path Composition and Multiple Choices in a Bimodal Transportation Network. In *Proceedings of the 7th World Conference on Transport Research Society*. Sydney, Australia. (1995)
8. BayernInfo: Trip Information for all Modes of Transport (2009). Available: <http://www.bayerninfo.de> (current July 2009)
9. Bérubé, J., Potvin, J., Vaucher, J.: Time-dependent Shortest Paths through Fixed Sequence of Nodes: Application to a Travel Planning Problem. *Computers & Operations Research*, Vol. 33, No. 6, 1838–1856. (2006)
10. Bielli, M.; Boulmakoul, A. Mouncif, H.: Object Modeling and Path Computation for Multimodal Travel Systems. *European Journal of Operational Research*, Vol. 175, No. 3, 1705–1730. (2006)
11. Chang E., Floros E., Ziliaskopoulos, A.: An Intermodal Time-dependent Minimum Cost Path Algorithm with an Application to Hazmat Routing. In: Zeimpekis, V., Tarantilis, C.D., Giaglis, G.M., Minis, I. (eds.): *Dynamic Fleet Management, Concepts, Systems, Algorithms & Case Studies*. Operations Research/Computer Science Interfaces Series, Vol. 38. Springer US, 113- 132. (2007)
12. Crainic, T., Rousseau, J.M.: Multicommodity, Multimode Freight Transportation: A General Modeling and Algorithmic Framework for the Service Network Design Problem. *Transportation Research Part B*, Vol. 20, No. 3 25-242. (1986)
13. Deo, N. and Pang C.: Shortest Path Algorithms: Taxonomy and Annotation. *Networks*, Vol. 14, No. 2, 275-323. (1984)
14. Dr'eo, J., P'etrowski, A., Siarry, P., Taillard, E.: *Metaheuristics for Hard Optimization*. Springer-Verlag, Berlin, Germany. (2006)
15. Feillet, D., Dejax, P., Gendreau, M., Gueguen, C.: An Exact Algorithm for the Elementary Shortest Path Problem with Resource Constraints: Application to Some Vehicle Routing Problems. *Networks*, Vol. 44, No. 3 ,216–229. (2004)
16. Fragouli M. and Delis A.: Navigation and Multimodal Transportation with Easytransport. *Intelligent Systems, IEEE*, Vol. 20, No.2, 54-61. (2005)
17. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA, USA. (1989)
18. Haupt, R.L., Haupt S.E.: *Practical Genetic Algorithms*. John Wiley & Sons, Hoboken, NJ, USA. (2004)
19. Holland, J. H.: *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, USA. (1975)
20. JPL: Journey Planner for London (2009). Available: <http://journeyplanner.tfl.gov.uk/> (current July 2009)
21. Kaufmann, D.E., Smith, R.L.: Fastest Paths in Time-dependent Networks for Intelligent Vehicle, Highway Systems Application. *IVHS Journal*, Vol. 1, No. 1, 1-11. (1993)
22. Kramer, R., Modsching, M., ten Hagen, K.: A City Guide Agent Creating and Adapting Individual Sightseeing Tours Based on Field Trial Results. *International Journal of Computational Intelligence Research*, Vol. 2, No. 2,191–206. (2006)
23. Lozano, A. and Storchi, G.: Shortest Viable Path in Multimodal Networks. *Transportation Research Part A*, Vol. 35, No. 3, 225-241. (2001)

24. Lozano, A. and Storchi, G.: Shortest Viable Hyperpath in Multimodal Networks. *Transportation Research Part B*, Vol. 36, No. 10, 853-874. (2002)
25. Meng, F.H., Lao, Y., Wai, L.H., Chuin, L.H.: A Multi-Criteria, Multi-Modal Passenger Route Advisory System. In *Proceedings of IES-CTR international symposium*, Singapore. (1999)
26. Modesti, P., Sciomachen, A.: A Utility Measure for Finding Multiobjective Shortest Paths in Urban Multimodal Transportation Networks. *European Journal of Operational Research*, Vol. 111, No. 3, 495–508. (1998)
27. Nguyen, S., Morello E., Pallotino, S.: Discrete Time Dynamic Estimation Model for Passenger Origin/Destination Matrices on Transit Networks. *Transportation Research Part B*, 22 (4), 251-260. (1988)
28. Nguyen, S., Pallotino, S., Malucelli, F.: A Modeling Framework for the Passenger Assignment on a Transport Network with Timetables. *Transportation Science*, Vol. 35, No. 3, 238-249. (2001)
29. Pallotino, S., Scutellà, M.G.: Shortest Path Algorithms in Transportation Models: Classical and Innovative Aspects. In: Marcotte, P., Nguyen, S. (eds.): *Equilibrium and Advanced Transportation Modelling*. Kluwer Academic, the Netherlands, 245-281. (1998)
30. Pun-Cheng, L.S.C., Mok, E.C.M., Shea, G.Y.K., Yan, W.Y.: EASYGO – A Public Transport Query and Guiding LBS. In: Gartner, G., Cartwright, W., Peterson, M.P. (eds.): *Location Based Services and TeleCartography*. Springer Science+Business Media, Berlin, 545-556. (2007)
31. Qu, L., Chen, Y.F.: A Hybrid MCDM Method for Route Selection of Multimodal Transportation Network. In: Sun, F., Zhang, J., Tan, Y., Cao, J., Yu, W. (eds.): *Advances in Neural Networks. Lecture Notes in Computer Science*, Vol. 5263. Springer, Berlin / Heidelberg, 374-383. (2008)
32. SCOTTY, Multi-Modal Door-to-Door Route Planner for Austria (2009). Available: http://fahrplan.oebb.at/bin/query.exe/dn?L=vs_addr (current July 2009)
33. Sherali, H.D., Jeenanunta, C., Hobeika, A.G.: Time-dependent, Label-Constrained Shortest Path Problems with Applications. *Transportation Science*, Vol. 37, No. 3, 278–293. (2003)
34. Spiess, H., Florian, M.: Optimal Strategies: A New Assignment Model for Transit Networks. *Transportation Research Part B*, Vol. 23, No. 2, 83-102. (1989)
35. Souffriau, W., Vansteenwegen, P., Vertommen, J., Berghe, G. V., Oudheusden, D. V.: A Personalized Tourist Trip Design Algorithm for Mobile Tourist Guides. *Applied Artificial Intelligence*, Vol. 22, No.10, 964-985. (2008)
36. Vansteenwegen, P., Oudheusden, D. V.: The Mobile Tourist Guide: An OR Opportunity. *OR Insight*, Vol. 20, No. 3, 21-27. (2007)
37. Ziliaskopoulos A.K., Wardell W.: An Intermodal Optimum Path Algorithm for Multimodal Networks with Dynamic Arc Travel Times and Switching Delays. *European Journal of Operational Research*, Vol. 125, No. 3, 486-502. (2000)

Rahim Aliabbaspour gained his doctoral in GIS in 2010 from university of Tehran. His main research interests are geospatial information science, artificial intelligence, and operational research. His emails address is abaspour@ut.ac.ir.

Farhad Samadzadegan is an associate professor of Geomatics at university of Tehran. He is currently the head of the department of surveying. His research interests cover the topics in computer vision, artificial intelligence, and meta-heuristics for optimization. You can contact him via samadz@ut.ac.ir.

Received: July 10, 2009; Accepted: July 16, 2010.

Trojan horses in mobile devices

Daniel Fuentes¹, Juan A. Álvarez¹, Juan A. Ortega¹, Luis Gonzalez-Abril²,
and Francisco Velasco²

¹ Departamento de Lenguajes y Sistemas Informáticos
Universidad de Sevilla
Avda. Reina Mercedes s/n, 41012, Seville, Spain
{dfuentes, jaalvarez, jortega}@us.es

² Departamento de Economía Aplicada I
Universidad de Sevilla
Avda. Ramón y Cajal 1, 41018, Seville, Spain
{luisgon,velasco}@us.es

Abstract. This paper focuses on the behavior of Trojan horses in mobile devices. This malicious software tries to steal information from a mobile device while the user is unaware. We describe the communication links through a Trojan horse installed into a mobile device. To demonstrate the effects of a Trojan horse infection we present a practical example on a PDA. Via SMS, the malicious user can access a user's contacts information through the previous installation of the Trojan horse. The results show that this process means a loss of information and a quantified cost to the attacked user too. This paper proposes different solutions to avoid this malware and its effects.

Keywords: Mobile security, Mobile Infections, Trojan horses.

1. Introduction

According to a study by the ITU (International Telecommunication Union) at the end of 2008, there were 4.000 millions of mobile phones in the world. A lot of people in the world are permanently communicated and can exploit the power of their terminals for advanced operations by the increased capabilities of these devices: via GPS navigation, Internet communication, photography, video, etc. According to Gartner Inc., in 2009 Smartphone sales surpassed 40 million units, a 27 per cent increase from the same period last year, representing the fastest-growing segment of the mobile-devices market. In addition, currently 33 millions of users already use mobile devices for shopping and are expected to be 103.8 millions in 2010. The increasing user base will influence collaboration and communication of enterprises more than ever: it is expected that more than 80% of the knowledge workers will receive and create information on notebooks and Smartphones by 2012 [1]. Communication, entertainment, exercising and travel are merely a few

Daniel Fuentes, Juan A. Álvarez, Juan A. Ortega, Luis Gonzalez-Abril, and Francisco Velasco

lifestyle improvements made possible with mobile technology. Moreover, the growing number of services and benefits, are becoming more essential in our daily life because they provide not only the basic voice communication service, but also contain other forms of communication such as instant messaging, multimedia messaging, Bluetooth, NFC (Near Field Communication) or e-mail. And all of these services can be an infection vector.

In this paper we explain a test program designed to study the behavior of a Trojan horse. Trojans, as well as viruses and worms, are known to create a backdoor that gives malicious users access to a system, possibly allowing confidential or personal information to be compromised. Due the growth of the personal information stored in mobile devices (location, mail, SMS, photos, etc.), we focus this work on one of its possible infections, the Trojans horses. We have demonstrated how Trojan horses can easily carry out the theft of the data contained in another user's phone book. Communication will be via SMS with predefined structures in which the attacker can send commands to the attacked PDA while the other user is aware of anything. The Trojan horse prototype for PDA will install into the attacked device and it will return user data from his contacts to the malicious user. The Trojan horse will be camouflaged inside an image (although it could also be implemented into an audio or video file).

The structure of this paper is as follows. In section 2 we present some related works. Section 3 describes different possible attacks on mobile devices and infection vectors. In particular, section 3.1 focuses on Skulls Trojan horse attacks whose main objective is, in addition to its own propagation, the theft of private user information while the victim is unaware. Section 4 describes the behavior of the Trojan horse in mobile devices through a practical experience where this infection is introduced on a mobile phone to obtain the user's contacts information stored in the terminal. In 4.3 we explain the results of the experiment and in 4.4 we propose different solutions to avoid the effects of the infection. Finally, in Section 5, we summarize our conclusions and present some lines of future work.

2. Background and related work

In the last years, many types of malware for mobile devices have appeared [2]. They can attack a device in many different ways, as we can see in Figure 1, and degrade mobile functions, delete or steal personal data, increase the victim's phone bill or disable the device completely. Each service that allows the user to connect to another device can be an infection for a virus intrusion or other threats [3-4].

SMS, MMS and Bluetooth [5] are together the most common ways for a possible infection. The small size of SMS (only 160 bytes, 168 characters) is the main disadvantage and the reason why there has not yet been a large-scale infection via SMS. However, MMS is one of the most used routes of

infections. The maximum size of the MMS is imposed by the service provider and depends on the device. For example, in India it's common for mobile videos to be 100kb and however in Sweden, the network Telia restricts MMS size to 300kB. All 3G compatible phones can receive/send 300kb MMS and most older phones may only allow 100kb, whilst even older phones may only allow 50kb. Today, the most extended size is 300KB which seems an appropriate size to accommodate the malware. Bluetooth technology develops different levels of security based on the identification of the devices involved but, in spite of that, the number of vulnerabilities via Bluetooth has increased considerably. One of the most dangerous vectors is e-mail [6], since there are no size restrictions and they can spread more easily to other tools, mainly PCs. Other ways, like USB connections, allow an infection to move from one device to another. Finally, WI-FI networks provide interoperable wireless access but sometimes the network origin and reliability is unknown.

Mobile malware detection [7-8] is a new area and almost all solutions that currently exist for mobile devices were originally created for PCs and they do not approach the key challenges of the mobile environment such as limited processing power as important issues. Recently, products like Flexilis [9] or Aircanner [10], which are dedicated exclusively to mobile devices security, protect mobile devices against threats including viruses, malwares or spam.



Fig. 1. Virus malware routes in mobile devices

2.1. Trojan horses in mobile devices

What happens if your mobile phone or PDA is lost or stolen? The device may contain confidential data and legal liabilities could arise if it contains confidential information such as medical records. We have already seen that

Daniel Fuentes, Juan A. Álvarez, Juan A. Ortega, Luis Gonzalez-Abril, and Francisco Velasco

there are many ways in which a mobile device can be attacked, but this article focuses on attacks with specific Trojans horses whose primary goal is, in addition to its own propagation [11], getting information in a maliciously way while the attacked user is unaware of the theft.

Nowadays, when users download a song, video, image or video game in his terminal, the data necessary for the download has been seen in an advertisement on TV, Internet website, magazine or newspapers. These downloads are usually done by sending SMS messages to telephone numbers that do not guarantee the safety of what the user will receive in his mobile phone. We could think that since we are paying for the download service, this may not prove to be an infection vector and, however, it is not too difficult to send a Trojan horse in a file of this type. As we will see in the demonstration, the Trojan horse will be camouflaged inside an image, but it can be introduced in any file so that when the infection arrives to the terminal, its malicious software will be installed.

One clear example of a Trojan horse is the Mosquito Trojan 2.0 [12], which accompanied the pirated version of the game for mobile devices with the same name. The Trojan did not affect the functionality of the device, but sent SMS messages to premium services (1 €/SMS approximately) while the user was playing with an illegal copy of the game. In fact, it is very probable that there are still websites where a user can download the game and, although there are two warnings before installing it, some users may be tempted to install it. Despite everything, this Trojan horse disappeared when the mobile game was deleted.

3. Development

There are programs like Windows Mobile Pro-X FlexiSPY [13] which performs mobile espionage. This application allows you to control all sent and received SMS messages and all call records and their duration, listen to telephone conversations, remote control software functions using SMS, or download directly into the device without a PC or cables. Moreover, if the device has a GPS function, it can be used as a crawler to get the coordinates to locate the device. It works with all versions of Windows Mobile 2003, except with Pocket PC, and it costs approximately 350 US\$.

We have carried out a simulation of a Trojan horse infection. This software allows a malicious user to steal all the contacts information. Then, we describe the Trojan horse implementation and the results, quantify the damage and take measures to prevent it.

3.1. Implementing a Trojan horse malware

Windows Mobile 6 Professional Software Development Kit was installed on the Microsoft Visual Studio 2008 programming environment to implement the

Trojan horse prototype. The attacked user will be played by the Visual Studio Simulator and when the program begins, it will emulate the installation of a malicious software running in the background while the user will only see a picture.

Moreover, the Microsoft Tool Cellular Emulator v1.43 was used to enable the malicious user to send and receive SMS messages and make calls (including other services) to the Visual Studio's emulator. Thus, the simulation of information exchange via SMS between mobile devices has been done in an efficient manner without using the services of any company. The main objective is to obtain private data from the attacked phone. In this application that information will consist of the contacts' names and telephone numbers.

The main Windows Mobile 6 SDK classes used for the demonstration were:

OutlookSession: It allows, among other functions, to access and modify data in the Contacts. In this case, it uses the nickname and the phone number.

MessageInterceptor: Personalized message receiver. It implements the channel that allows the infection to remain pending for an incoming SMS. The purpose of the *MessageInterceptor* class will be a key factor in the implementation of the Trojan horse and the proposed solution (as we shall see below), because it contains the event which receives SMS messages (*MessageReceived()*).

SmsMessage: It implements the creation and sending of SMS.

3.2. Information flow

The operation on the user's attacked device is shown in the pseudocode:

1. The attacked user receives the picture where the Trojan horse infection is packed. The file can be transferred from the Internet through MMS or via Bluetooth to the terminal.
2. Once the virus reaches the mobile phone, it automatically installs itself.
3. The malicious program awaits orders. The attacker's instructions are introduced by means of a SMS with a default structure.
4. If the received SMS is in the correct format, in this case with the head *@spy@*, the content processing begins. Otherwise, the message goes to the user's inbox.
5. Then, the Trojan horse checks the label-value pairs. The parser recognizes the pairs *<sms>telephone_number*.
6. The program automatically sends the Contacts data in the Phonebook to each phone *<sms>telephone_number* pair which appears in the SMS received (stage 3). In the demo, it was sent via SMS to each contact

Daniel Fuentes, Juan A. Álvarez, Juan A. Ortega, Luis Gonzalez-Abril, and Francisco Velasco

name and phone number with the format *contact_name:telephone_number*, but different data could also be sent.

7. Once every SMS is sent, the Trojan horse infection awaits new orders. The pseudocode of the Trojan horse behavior on the attacked device is:

```
image_download();
trojan_horse_installation();
execution_in_background();
if (sms_received ())
    if(is_malicious_user(SMS_recieved))
        while(is_all_information_contacts_sent())
            SMS_send(information_contacts,
                    SMS_recieved .telephone_number);
        end_while
    else
        send_SMS_to_inbox();
    end_if
end_if
```

Now, the Trojan horse is installed in the victim's mobile phone. In turn, the Trojan horse behavior on the attacker device is described in detailed in the next process.

1. The malicious user sends an order to the Trojan horse by a SMS to the user under attack in an appropriate format, which in the test application is *@spy@<sms>telephone_number<sms>telephone_number<sms>...*
2. Awaits the response of the Trojan horse.
3. Begins to receive SMS messages to the structure *contact_name:telephone_number-contact_name:telephone_number-...*The messages are processed through a second parser in which the malicious user decides how to process information.

The process is completed when the user decides to send an SMS with new orders that lead to begin the process of reattacking. In pseudocode, the Trojan horse behavior on the attacker device is:

```
send_SMS();
while(no_response_recieved())
    if(SMS_recieved())
        processing_contacts_information();
    end_if
end_while
```

3.3. Results of the experiment

Obviously, the whole process takes place without the user's awareness of the attack because the Trojan horse remains running in the background. In

addition, messages sent from the device to the attacked phone do not arrive to the mailboxes, so they do not arouse suspicion. Furthermore, sending SMS messages entails an economic cost. For a possible estimation, we will assume that the average length of the contact's name or nickname is ten characters approximately and we know that mobile phone numbers consist of nine characters. If we add the spaces, a contact's information consumes twelve characters. An average user may have one hundred contacts stored in his phone book and an SMS costs 0.15€ on average in Spain (according to Viviane Reding, European Commissioner for Information Society). Therefore, whenever a malicious user requests data from the contacts in the agenda of the attacked device it will cost the attacked user 8.62€ approximately and the malicious user can begin this process whenever he wants.

In tests carried out on a HTC 3300 PDA, it has been found that when the device is in any way attacked (after the Trojan horse was installed) it is not aware of the entry or exit of information via SMS, which the Trojan horse uses in the process. Moreover, the device does not save copies of those SMS messages in inbox or outbox. However, when the user receives a message with the malicious Trojan horse the screen light turns on but still nothing happens.

3.4. Proposed solutions

A Trojan horse that has easily allowed the obtaining of information from the Contacts of an infected device has been implemented. It has been also found that the theft of information goes completely unnoticed by the user of the attacked terminal.

Therefore, one possible solution is a service that uses the Event Listeners (for SMS, MMS, GPS, Bluetooth...), which are provided by the specific programming language to detect access to a list of potentially dangerous resources in the event the terminal is the target of an attack. In the case of our demonstration activities, the *MessageReceived()* event is used to receive SMS. This event is provided by Microsoft Windows Mobile 6 SDK (*MessageInterceptor()* method) to detect any access via SMS that occurs in our system (in PDAs the test program reports no information for in/out SMS), and thus the system can alert the user that a new message has been received, regardless of its content or origin.

When a new SMS arrives, the user will be alerted of the arrival by the PDA, providing additional information as to the identity of the sender or the first characters of the message. In turn, the user can accept it (go to inbox) or delete it, as appropriate.

This is undoubtedly a simple and flexible solution that can contribute to a more comprehensive control of entry and exit information from our device. However, the user will make the final decision, helped by his experience and the content of the SMS, to allow sending or receiving a message.

Another solution would be including a signature in outgoing SMSs, but this procedure shows several disadvantages, for example, the origin of the SMS

Daniel Fuentes, Juan A. Álvarez, Juan A. Ortega, Luis Gonzalez-Abril, and Francisco Velasco

is not one hundred per cent ensured; in addition, it does not work with incoming messages and reduces the free space for writing messages. However, with this method, only signed messages could be sent and the problem caused by the infection could be solved.

Moreover, another solution for incoming messages would be the implementation of a new parser to check these messages. First, the sender's phone number is checked and if it does not belong to a known contact, the contents would be checked again by the parser. The main problem, however, is the parser design and which characters are to be included.

4. Conclusions and future work

Currently most mobile users, unlike PC users, do not feel the need to install on their mobile terminals an antivirus program or other devices to protect them from potential infections. However, due to the growth of services, capabilities and stored data of these devices, it is almost indispensable to take any measure against a possible attack.

In this article we have discussed the attacks on mobile terminals by Trojan horses, through which new vulnerabilities can be installed on the attacked device, capturing and sending some of the private information for future misuse, taking into account that throughout the process the user is unaware of the theft and the device functionality is not damaged at all.

We have developed a demonstration to show the theft of part of the contact information contained in the mobile device phonebook. This experiment was conducted on a simulator provided by Microsoft Windows Mobile 6 SDK and then ran on a PDA. The solution proposed in the article provides a further control on the flow of information sent and received from the device.

As future work, we plan to study the behavior of Trojan horses when they attack other terminal services such as Bluetooth, GPS or e-mail. And not just Trojan horses, but other types of vulnerabilities such as worms or viruses that could affect the operational ability of the device (software or hardware).

References

1. Beurer-Zuellig, B. and Meckel, M., "Smartphones Enabling Mobile Collaboration", in Proceedings of the 41st Hawaii International Conference on System Sciences, pp. 49-49, 2008.
2. Gostev, A., "Mobile Malware Evolution: An Overview", [Online]. Available: <http://www.viruslist.com/en/analysis?pubid=204792080>, Sept. 2009
3. Van Ruitenbeek, E., Courtney, T., Sanders, W.H. and Stevens, F., "Quantifying the Effectiveness of Mobile Phone Virus Response Mechanisms". In Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, pp. 790-800, 2007.

4. Ganesh, A. J., Massoulié, L., and Towsley, D., "The effect of network topology on the spread of epidemics". In Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies. vol. 2, pp. 1455-1466, 2005.
5. Bose, A., and Shin, K. "On mobile viruses exploiting messaging and bluetooth services". In International Conf. on Security and Privacy in Comm. Networks (SecureComm'06)", pp. 1-10, Sept. 2006.
6. Newman, M., Forrest, S., and Balthrop, J., "Email networks and the spread of computer viruses", Phys. Rev.E 66, 3, Sept. 2002.11.
7. Pradip, D., Liu, Y. and Das, S.K., "An Epidemic Theoretic Framework for Vulnerability Analysis of Broadcast Protocols in Wireless Sensor Networks", in IEEE Transactions on Mobile Computing, vol. 3, pp. 413-425, 2009.
8. Bose, A., Hu, X., Kang, G., Shin, K.G. and Park, T." Behavioral detection of malware on mobile handsets", in Proceedings of the 6th international conference on Mobile systems, applications, and services, pp. 225-238, 2008.
9. Flexibilis, <http://www.flexibilis.com>
10. Airscanner, <http://www.airscanner.com>
11. Fleizach, C. B., "Can You Infect Me Now?. A Treatise on the Propagation of Malware in a Cellular Phone Network". Tech. Rep. CS2007-0894, UCSD, June 2007.
12. Flexi SPY, <http://www.flexispy.com/>
13. Kim, S. h. and Leem, C. S., "Security Threats and Their Countermeasures of Mobile Portable Computing Devices in Ubiquitous Computing Environments". In Proceedings of the International Conference Computational Science and Its Applications (ICCSA 2005), pp. 79-85, 2005.

Daniel Fuentes Brenes received his bachelor degree in Computer Science in March 2008. Since June 2008, he is a Lecturer in the Department of Languages and Computer Systems at the Seville University. His main research interests are mobile computing and machine learning.

Juan Antonio Álvarez García received his bachelor degree in Computer Science in 2003. He is a Lecturer since 2003 in the Department of Languages and Computer Systems at the Seville University. His main research interests are ubiquitous, mobile and urban computing. He is also interested in healthcare systems.

Juan Antonio Ortega Ramírez obtained the Ph.D. degree in Computer Science in 2000 at the Seville University in Spain. He is a Lecturer since 1992 in the Department of Languages and Computer Systems at the Seville University. His research interests are: the temporal series and the global information systems and specifically the domotic and assistencial systems. He is Head of the Centre of Computer Scientific in Andalusia (Spain).

Luis González Abril is a Lecturer in the Dep. Of Applied Economy I at the University of Seville (Spain). He obtained his graduate in Mathematics in 1986 from the University of Sevilla and his Ph. D. degree in Economy in 2002 from the University of Seville. His researchs are: Similarities, Support

Daniel Fuentes, Juan A. Álvarez, Juan A. Ortega, Luis Gonzalez-Abril, and Francisco Velasco

vector machines, machine learning and bifurcations of dynamical systems applied to economic problems. e-mail: luisgon@us.es.

Francisco Velasco Morente is a Lecturer in the Dep. Of Applied Economy I at the University of Seville (Spain). He obtained his graduate in Mathematics in 1979 from the University of Sevilla and his Ph. D. degree in Mathematics in 1991 from the University of Seville. His researchs are: similarities, Support vector machine, bifurcations of continuous and discrete dynamical systems and optimal control problems, both applied to economic problems.

Received: March 30, 2009; Accepted: August 09, 2010.

A Tabular Steganography Scheme for Graphical Password Authentication

Tsung-Hung Lin¹, Cheng-Chi Lee^{2,4}, Chwei-Shyong Tsai³,
and Shin-Dong Guo⁴

¹Department of Computer Science and Information Engineering,
National Chin-Yi University of Technology, TaichungTaiwan, R.O.C.
duke@ncut.edu.tw

²Department of Library and Information Science,
Fu Jen Catholic University, Taipei, Taiwan, R.O.C.

Correspondence: clee@mail.fju.edu.tw

³Department of Management Information Systems,
National Chung Hsing University, Taichung, Taiwan, R.O.C.
tsaics@nchu.edu.tw

⁴Department of Photonics & Communication Engineering,
Asia University, Taichung, Taiwan, R.O.C.

Abstract. Authentication, authorization and auditing are the most important issues of security on data communication. In particular, authentication is the life of every individual essential closest friend. The user authentication security is dependent on the strength of user password. A secure password is usually random, strange, very long and difficult to remember. For most users, remember these irregular passwords are very difficult. To easily remember and security are two sides of one coin. In this paper, we propose a new graphical password authentication protocol to solve this problem. Graphical password authentication technology is the use of click on the image to replace input some characters. The graphical user interface can help user easy to create and remember their secure passwords. However, in the graphical password system based on images can provide an alternative password, but too many images will be a large database to store issue. All the information can be steganography to achieve our scheme to solve the problem of database storage. Furthermore, tabular steganography technique can achieve our scheme to solve the information eavesdropping problem during data transmission. Our modified graphical password system can help user easily and friendly to memorize their password and without loss of any security of authentication. User's chosen input will be hidden into image using steganography technology, and will be transferred to server security without any hacker problem. And then, our authentication server only needs to store only a secret key for decryption instead of large password database.

Keywords: graphical password authentication; security; teganography; protocol.

1. Introduction

Today's network environment is full of dangerous attackers, hackers, crackers, and spammers. Authentication, authorization and auditing are the most important issues of security on data communication. In particular, the user authentication is indispensable to every person living closest friends, but also a key area of security research. Authentication system is based on the passwords which use the alphanumeric for authentication and identifier. Many authentication techniques including biometrics and smart card are the possible and useable applications. The authentication security can be increased, but sometime the passwords seem to remain dominant due to the drawbacks of reliability, security, or cost of other techniques [3][11][13][15]. Biometric techniques make use of physiological or behavioral characteristics to be their identities. Authentication may be secure, but it still needs to resolve the security usability and balance for general usage [13]. And, smart card also needs to type passwords for user authentication. In this paper, we propose a new graphical password authentication protocol to solve this problem. Graphical password technique is one of methods which may provide more secure and more efficiency system for authentication. A set of secure passwords needs to be long enough and random [4], but that will be a problem for human to remember. Everyone will forget their settings everyday if they didn't use again [12]. The research results showed that, when users forget their password, they can only remember part of the correctness [8]. Usable and easy memorization is the main research issues of graphical password authentication. In this paper, we modify and redesign the Passpoints scheme to improve the efficiency of password authentication. Passpoints is a new, more secure and more flexible graphical password that proposed by Wiedenbeck et al. [23]. In addition, the stored passwords database is another common problem for graphical password systems. In our system, we use steganography technique to hide the user graphical password points on the image. Then we transfer the encryption key to server. Server only needs to store a secret key to the database. Server can use the secret key to decrypt user graphical password points from any images which store in the database. By the way, our tabular steganography technique can not only provide more security and also reduce the storage of database information. Furthermore, tabular steganography technique can achieve our scheme to solve the information eavesdropping problem during data transmission.

2. Related Works

Graphical password authentication protocol is first described in Blonder [7]. Blonder proposes a new idea, he lets users use mouse or stylus by themselves to click the picture on presetting correct regions for replacing the traditional text input password. After that, graphical password systems are popular and several different issues are developed [2][9][10][14][20][23].

Graphical password authentication is an image-based authentication technique which can be divided into two categories [2][27]. We describe these two categories, recognition-based authentication technique and recall-based authentication technique as follows.

2.1. Recognition Based

Recognition-based technique is a kind of graphical password authentication techniques which need to choose those correct pictures from many pictures. Dhamija and Perrig [18] propose a scheme which can use Hash Visualization technique [1] to support recognition-based graphical password authentication. Kotadia [16] proposes a new recognition-based graphical password authentication that is a multi-image technique with one step for authentication. In Dhamija and Perrig's system, the system shows some random generated images and the users are asked to select a certain number of images for the authentication in the graphical interface [18], as shown in Fig. 1. Before use, the user needs to set in advance through the authentication sever to identify images. However, the average log-in time is longer than input alphanumeric password, but the result showed that 90% of all participants succeeded and the text-based password with PINS only 70% in the result [27]. The scheme proposed by Dhamija and Perrig was not really secure because the passwords need to store in database and that is easy to see.

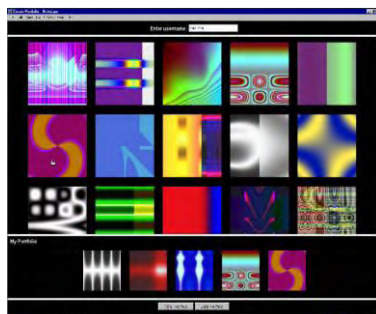


Fig. 1. Random images select used by Dhamija and Perrig [18]

After that, Akula and Devisetty's proposed algorithm [20] is similar to Dhamija and Perrig's, the different is that it uses hash function SHA-1 rather than Hash Visualization technique. Their scheme can be more secure and require less memory than Dhamija and Perrig's [18]. "Passface" is still a multi-image technique developed by Real User Corporation [19]. Users will be asked to choose four face images to be the password. In authentication step, the users see a grid of nine faces, as illustrated in Fig. 2, and only one face was correct image.



Fig. 2. An example of Passface which shows nine faces in a grid (source: www.realuser.com)

User should choose the correct one from these nine faces and this step will repeat four times. “Passface” is based on assumption that people can recognize human faces easier than other pictures. Nevertheless, Valentine [25][26] has shown that “Passface” protocol is very memorable over long interruption. This is easier to remember, but not secure, users click four times in a nine grid has only equal third of the login failure rate of alphanumeric-based password [21].

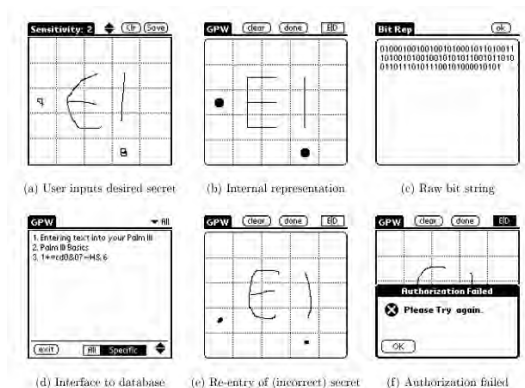


Fig. 3. Draw-a-Secret (DAS) graphical password system

2.2. Recall Based

Another image-based authentication technique is recall-based authentication technique. “Draw-a-Secret” (DAS) is a famous recall based technique which proposed by Jermyn, et al. [11]. DAS technique allows users to draw their passwords on the interface, as shown in Fig. 3. A user will be asked to draw a

simple picture and the picture will be store for authentication. During authentication, the user is asked to draw their unique password. User should draw the same grids in the same sequence, and then the authentication will be success. DAS technique can let users draw by themselves, but need more time then alphanumeric-based password [6], and if the users draw in the middle of two sequences, the system will be error. It is difficult that users should draw their password correct and always be same with the first time. Passlogix [17] has developed a graphical password system which is based on a designated image. In this scheme, user should click different items on an image in order to be authenticated, as illustrated in Fig. 4. User should recall the various items that combined their password and choose these correct items. This scheme is not flexible, but certainty provides more security. Passpoints [23] is a kind of single-images based graphical password scheme and this scheme provide more password space to support more security than textual passwords technique. Passpoints scheme also needs one picture to be the interface and allows users to click anywhere in this designated picture to be the user password. Passpoints provide large password space [10] to ensure the secure and provide more flexible interface for user. In addition, Birget et al. [10] proposed a new scheme that is based on the discretization method. Passpoints is our main subject and we will introduce and improve in next subsection.

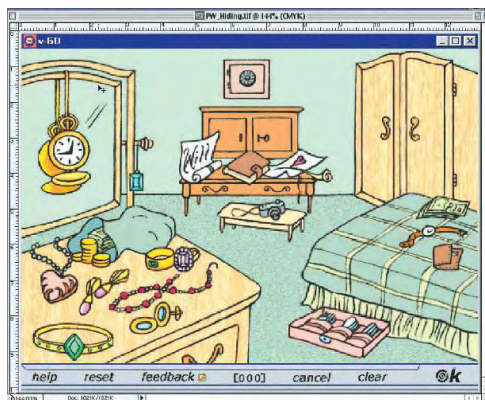


Fig. 4. A recall-based graphical password system

2.3. Passpoints

Passpoints [23] is a graphical password scheme similar to Blonder's scheme [7]. Passpoints allows any images to be used and users can click on this designated image to complete their password authentication. A rich and colorful image can be divided into hundreds or more squares. The system of Passpoints includes both graphical and alphanumeric interfaces, but we only

discussed the graphical interface. The graphical interface includes a rich picture and several buttons, as shown in Fig. 5. The size of the picture is 451 x 331 pixels and the grid square around a click point is set to 20 x 20 pixels. The special buttons of graphical interface is the “See My Password” button, and it allows users to view their password when they are clicking for ensuring the correct password. Users choose sequence of these memorable points to be the password and it can be hashed that means more security of this graphical password system.

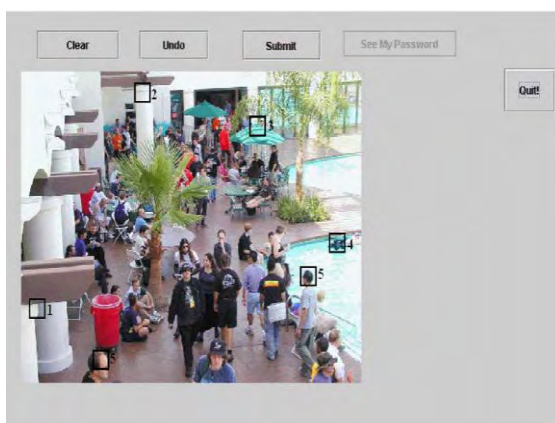


Fig.5. An example of Passpoints graphical password input interface

2.4. Security of Passpoints

The graphical password authentication security is based on the “password space”. We carefully compared the Passpoints graphical password with the alphanumeric password. In Table 1, we show that the password space between graphical password and alphanumeric password. For example, a 64 character alphabet which includes 10 digits, 26 lower-case letters, 26 upper-case letters, underscore, and dot. If a user chooses alphanumeric password length eight over 64 character alphabets, then the entire number of guessing possible password is $64^8 = 2.8 \times 10^{14}$. In graphical password system, there is an image size $451 \times 331 = 149,281$ and grid square size $20 \times 20 = 400$, there are about $149281 / 400 = 373$ grid squares. If a user chooses 5 click points for the password and the entire number of possible is $373^5 = 7.2 \times 10^{12}$. The image size 451×331 is a small password space, but with a big image and more click points will increase the password space, we show the comparison in Table 1. “Passpoints” is secure and have enough password space, but needs to save more pictures for user to use. It will make a huge burden for database storage. In our modify scheme, we will use the “Steganography”

technique to reduce the burden of database storage and get our final goal—**without any password table.**

Table 1. Comparison of password space between graphical and alphanumeric password by Susan, et al. [22]

	Image size	Grid square size (pixels)	Alphabet size/ No. squares	Length/No. click points	Password space size
Alphanumeric	N/A	N/A	64	8	2.8×10^{14}
Alphanumeric	N/A	N/A	72	8	7.2×10^{14}
Alphanumeric	N/A	N/A	96	8	7.2×10^{15}
Graphical	451x331	20x20	373	5	7.2×10^{12}
Graphical	1024x752	20x20	1925	5	2.6×10^{16}
Graphical	1024x752	14x14	3928	5	9.3×10^{17}
Graphical(1/2 screen used)	1024x752	14x14	1964	5	2.9×10^{16}

3. Our Graphical Password Authentication Scheme

3.1. Our Scheme

In general authentication process, the password will be hashed and stored for mapping to the correct identifier. It means the authentication database needs to store entire hashed security password. The more the users have registered to the system, the more databases will be stored for huge data. To reduce the amount of data storage is one of our main objectives. We hope that the database be able to more space to do a lot more things. We also hope that the number of users regardless of the size of the database remains the same efficiency.

The notations used in proposed scheme are summarized as follows:

A denotes the user A.

ID denotes the user identity of user A.

GPW denotes the encoded graphical password of user A.

S denotes the server identity.

N denotes a sequence number starting from 1 in the first register.

x denotes the secret key of S used to encrypt and generating a unique storage unique key storage key for each user.

ISN denotes the image serial number which user chosen to be the click image.

|| denotes the concatenation operation.
 \oplus denotes the bitwise XOR operation.
 $h()$ represents a cryptographic hash function.

3.2. Registration Phase

In the registration phase, user will go through a security channel to register this system. We need users to set their *IDs*, login images and graphical password points, then server will use the secret key x to encrypt the graphical password and hide in the image which users choose. When users want to register this system, system will show the images for users to choose, and then compute the hiding values. In our scheme, we use the steganography technique to hide the encryption value in the images which are chosen by users. The encryption scheme can use AES or DES schemes. The steganography technique can also use any present scheme like vector quantization [24] scheme or others. The registration phase is processed in security environment, and the workflow is described as follows.

Step R1. $A \rightarrow S$: User A sends registration request, *ID* and *ISN* to server S.

Step R2. Server shows the registration interface to user A. User A sets the necessary registration information, *ID*, *ISN*, and *GPW* then computes the verifier $V = h^2(S||GPW||ISN||N)$.

Step R3. $A \rightarrow S$: V

Step R4. Server computes the key $K_A = h(ID||h(x)||ISN)$, and computes the hiding verifier $hv^{(A)} = V \oplus K_A$.

Step R5. Server will request user to practice about five times for user who can remember his or her graphical password.

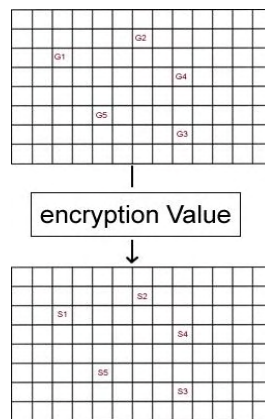


Fig. 6. Our mapping scheme use steganography technique

For example, an image will divide into many grids and we use information hiding technology to store secret key x and password in this image, as illustrated in Fig. 6. In Fig. 6, we also show the grids to explain the information hiding. We set a key x for secret key and use click five points which definition to G1, G2, G3, G4 and G5, respectively. We use secret key x and the graphical password points to do steganography, then hide the encryption value in five secret points S1, S2, S3, S4 and S5 in the image which is chosen by user. The hiding points are embedded in the same place as the graphical password points.

3.3. Login Phase

When a user wants to login the system, the user needs to input his or her ID and choose the correct image and click correct graphical password points. Then server will take the hiding value to authenticate user. In the login phase we have two symbols are ra and rs , ra is the random number which is decided by the arrangement of nine grids and rs is the random number chosen by server. If the users forget their chosen graphic points (locations) or images they uses, then they should bring up their petition and process the system's identity verification procedure. If the users have passed the identity verification, then they will redo the registration phase to set their new passwords. The flow is similarly to the processing of currently password forgotten processing phase. No one can help you to get your forgetting password. Because of everyone should carefully keep their passwords by themselves.

The workflows of login phase are described as follows.

Step L1. $A \rightarrow S: ID, ra$

Step L2. Server computes $K_A = h(ID||h(x)||ISN)$, and then reduces the $hV^{(a)}$ from the image.

Server computes $V = hV^{(a)} \oplus K_A$ and $h(V \oplus ra)$.

Step L3. $S \rightarrow A: rs, h(V \oplus ra)$

Step L4. A computes $V = h^2(S||GPW||ISN||N)$, then computes $h(V \oplus ra)$.

If $h(V \oplus ra)$ equals to the received one, then the user A authenticates server S .

Then A computes

$d_1 = V \oplus h(S||GPW||ISN||N)$,

$d_2 = h(S||GPW||ISN||N) \oplus h^2(S||GPW||ISN||N+1)$,

$d_3 = h(h^2(S||GPW||ISN||N+1)||rs)$.

Step L5. $A \rightarrow S: d_1, d_2, d_3$

Step L6. Server S uses the hiding verifier to compute $u_1 = d_1 \oplus V$, then computes $h(u_1)$. Then S verifies it whether equal to V , or terminates this session.

S computes $u_2 = d_2 \oplus u_1$ and will get $h^2(S||GPW||SM||N+1)$.
 Finally, if $h(u_2||rs)$ equals to d_3 then the server authenticates A.

The "Passpoints" system needs to store user ID, images and password table in the database. In our scheme, we use steganography technique to hide the user graphical password points in the image and our server only needs to store a secret key x . Server can use the secret key x to decrypt users' graphical password points from any images which are stored in the database.

3.4. Analysis of Our Scheme

The Security of Our Scheme

We will discuss some attacks that may occur on our scheme and describe how to resist these attacks and analyze the guessing attack. Although the graphical password scheme can resist the guessing attack, but our scheme can improve it and spare larger password space. The graphical password system assigns one user with an image and that will let the database store too much information. In our scheme, an image is chosen from a group and many users can share the same image because the secret key x is the same and the recognition scheme can provide more security for our system.

Resistance to Replay Attack

Suppose that $N = n$ and the hacker or adversary has captured all the past authentication messages of user A which $\{d_1(i), d_2(i), d_3(i)\}$ for $i = 1, 2, 3, \dots, n-1$. The correct hiding verifier of A is $h^2(S||GPW||ISN||n)$, and the hacker or the adversary cannot login into the system by using $\{d_1(i), d_2(i), d_3(i)\}$, where $1 < i < n-1$. The each value is as follows:

$$\begin{aligned} d_1(i) &= h^2(S||GPW||ISN||i) \oplus h(S||GPW||ISN||i) \\ d_2(i) &= h(S||GPW||ISN||i) \oplus h^2(S||GPW||ISN||i+1) \\ d_3(i) &= h(h^2(S||GPW||ISN||i+1)||rs) \end{aligned}$$

If the hacker or adversary replaces the $d_2(n)$ and $d_3(n)$ with $d_2(i)$ and $d_3(i)$, where $i = 1, 2, 3, \dots, n-1$ during A login phase. Server will be aware of this crafty attack, because $h((d_2(i) \oplus (d_1(n) \oplus h^2(S||GPW||ISN||n))))||rs(n))$ will be not equal to $d_3(i)$ which is $h(h^2(S||GPW||ISN||n+1)||r(i))$. Another case is that the hack or adversary could cheated and fooled into the server S to replace the verifier $h^2(S||GPW||ISN||n)$ with $h^2(S||GPW||ISN||i)$ of A, where $1 < i < n-1$. But the hacker cannot impersonate user A to login S because $rs(n)$

$\neq rs(i)$, and $h((d_2(i) \oplus (d_1(n) \oplus h^2(S||GPW||ISN||n))) \oplus rs(n))$ is not equal to $d_3(i)$. On the other hand, the hacker or adversary do not know the $h^2(S||GPW||ISN||n)$, and certainly he cannot generate it. It is impossible to send correct $h(h^2(S||GPW||ISN||n) \oplus ra)$ to user A in the step L3 when finish receiving the ra sent from user A in step L1. Therefore, the hack or adversary cannot be successful to emulate server to cheat user A by such replay attacks we had decried before. Our scheme can certainly resist the replay attack.

Resistance to Guessing Attack and Improve Password Space

In the input phase, we show nine images for the user to choose and each image size 451 x 331 has 7.2×10^{12} numbers possibility to guess [5], and now the numbers of possibility were $9 \times 7.2 \times 10^{12}$. We just discuss about the guessing attack and the security is based on the password space. The password space in our scheme can be increased and our scheme can provide more security and efficiency in graphical password system. Although the graphical password system can resist the guessing attack but in our scheme, we can provide larger password space than Passpoints and the nine grids scheme also can help us to resist more attacks.

Resistance to Forge Attack

We use the nine grids arrangement to be the random number ra for user and server S selects random number rs to user. These random numbers will be different from every login processes. If a hacker can obtain all the information which is sent on the internet, the hacker did not know the graphical password. Hacker cannot compute the correct authentication data which is $h(S||GPW||ISN||n)$ and also cannot to produce the correct keys $\{d_1, d_2, d_3\}$. The ra used by user to authenticate the server and the rs used by server to authenticate user, only the legal user and server can get the correct values.

Resistance to Password-File Compromise Attack

Usually, the authentication server saves the keys N and ISN in the password-file. In our scheme, we need not to save password table, because the keys N and ISN are small and we use them to hide the $hV^{(a)}$ into the image. We suppose the password file of server has been compromised by hacker or adversary. If hacker has obtain the ISN , N and the $hV^{(a)}$ of user A. The $hV^{(a)}$ is the hiding verifier of user A and the value is $h^2(S||GPW||ISN||n) \oplus K_A$, the ISN and N are not secret. But the hacker cannot compute correct $h^2(S||GPW||ISN||n)$ because he do not know the K_A which is $h(A||h(x)||ISN)$. The secret key x can resist the password-file attack. Only server S can use secret key x to take out the hiding value from images. And, only server S can contrast them when the hiding value has been changed. If the secret key x

has been not secret anymore, the server should change another secret key and all the users should choose their new graphical passwords and register to S again.

Table 2. The attacks and security features on graphical password

Graphical Password Scheme	Techniques		Possible Methods					Attack			Security on Password		Features on Graphical Password	
	Recognition	Recall	Brute force	Dictionary	Guessing	Spyware	Shoulder surfing	Social	Large password space	Randomly assign	Hash function	Image variation	Decoy images	Repeat verification
Jansen et al.	V		V	X	V	X	V	X	X	V		V		
Passfaces	V		V	V	V	X	V	X	X	V		V		
Triagle	V		V	X	V	X	X	X	V	V		V		
Movable Frame	V		V	X	V	X	X	X	V	V		V		
Intersection	V		V	X	V	X	X	X	V	V		V		
Pict-O-Lock	V		V	X	X	V	X	X	V	V		V	V	V
Déjà Vu	V		V	X	V	X	V	X	X	V	V	V		
Blonder		V	V	X	V	X	V	X	V					
VisKey SFR		V	V	X	V	X	V	X	X					
Passlogix v-Go		V	V	X	V	X	V	X	X					
PassPoints		V	V	X	V	X	V	X	V		V			
DAS		V	X	V	V	X	V	X	V		V			
V=Yes X=No Blank=not mentioned														

Resistance to Insider Attack

The insider attack is named that user A uses the same graphical password to access several servers for convenience. If the insider hacker of server has obtained the graphical password GPW , then he can impersonate user A to access other servers. Actually, it is not easy and the insider of server cannot directly obtain the GPW , because the user cannot reveal GPW from server in the registration and login phases. The insider server also cannot know the value of GPW from user A. Another supposing that is the server wants to impersonate user A to log into another servers. If a vicious server knows the hiding verifier $h^2(S||GPW||ISN||n)$, and then try to impersonate A to log into another server, say S^* . Then, the $h^2(S||GPW||ISN||n)$ will not be equal to $h^2(S^*||GPW^*||ISN^*||n^*)$ even if $GPW^* = GPW$, $ISN^* = ISN$ and $n^* = n$. The insider server cannot successfully impersonate user A to login S^* . Our scheme can certainly resist the insider attack.

Resistance to Denial-of-Service Attack

Recently, denial-of-service (DOS) attack becomes the important issue on data communication security. To achieve prevention from the denial-of-server attack, server S has to make sure that u_2 computes correctly. The d_3 can protect the integrity of d_1 and d_2 . And, the d_3 can be used to calculate the u_2 , and any unauthorized changes of d_1 , d_2 and d_3 will be found in the server. The hacker or adversary cannot disable any user's account and the proposed scheme can really resist the denial-of-server attack. Our scheme can resist many kinds of attacks and improve the password space of the Passpoints system. In Table 2, Hafiz et al. [28] show the features and possible attacks of graphical password systems, we use this table to prove our improvement and our features. The Passpoints will be attacked by guessing but the large password space can resist it. We use nine grids to decrease password space and will be more effective to resist attacks.

Improve Database Storage

Recognition based graphical password techniques has a big problem that is to store many images for users. The password database needs to store ID, password table, images and other information which server need. In the premise, server needs enough space to store many sorts of information, especially the images. It will cause equipment burden. In our scheme, we successfully overcome this problem that database does not have to store the large data instead the secret key x . No matter how many users use the system, server just needs to pass the secret key x and key x is only for server using. Although in our scheme also need to store some images but when the user member increases, our scheme will still be a very effectual scheme.

3.5. Security Policy

In graphical password system, shoulder-surfing is a big problem but many researches can solve this problem. Shoulder-surfing means a person who stands on the back of user and wants to see the password when user inputs their passwords. We introduce some schemes and our policy to prevent shoulder-surfing problem. First scheme is developed by Sobrado and Birget [14] and which system displays a number of pass-objects among many different objects. User needs to recognize pass-objects to be authenticated. Second scheme is developed by Man, et al. [22], user needs to select a number of pictures as pass-object, but each pass-object has several variants and each variant is unique code. Third scheme is developed by Hong, et al. [5], and this scheme still uses pass-object, but allows user to assign their own codes to pass-object variant. Shoulder-surfing is a problem, but we propose a simple scheme to prevent this problem. In traditional authentication phase, User uses mouse to click their correct passwords on the interface which uses the left-button of mouse. Our improved scheme is that we can use the right-button of mouse to confuse the person who wants to see the password input. Any person wants to see the password but will not know which clicks are correct. We can set our system to accept only left-button, but let right-button of mouse to click and display. User can use left-button or right-button of mouse, but only the designated-button will be accept and authenticated.

3.6. Usability of Graphical Password

The usability is very important of graphical password and number of existing graphical password schemes available on the internet. We discuss about the usability and explain why the "Passpoints" can really use in the future. In Table 3, we show the usability features of graphical password which is made by Hafiz, et al. [28] and there are 12 schemes in the table to compare with their usability. The Passpoints scheme is the best of these 12 schemes in compare with each usability feature.

The Passpoints scheme is the only scheme that can be considered as an efficient scheme. The input reliability and accuracy are one of the features which can provide the usability and users can easily remember their graphical password. Even we use nine grids but do not spend many times and we believe that our modified Passpoints still can keep efficient. For user to use our system, user clicking one of nine pictures is just like to choose one of the correct points in Passpoints. Although this is easy but can really improve the security and also can keep efficient. If the users forget their setting points or images, they can redo the setting phase by the certified phase that same as the traditional alphanumeric password certified phase.

Table 3. The usability of graphical password

Graphical Password Scheme	Techniques		Usability Features on Graphical Password											
	Recognition	Recall	Memorability							Efficiency	Input reliability &	Easy and fun to use	Grid based	Drawing password
			Meaningfulness	Human faces	Organized by theme	User assign image	Leon based	Abstract image	Navigating image					
Jansen et al.	V		V		V						X	V	V	
Passfaces	V			V	V						X	V	V	V
Triagle	V						V				X		V	
Movable Frame	V						V				X		V	
Intersection	V						V				X		V	
Pict-O-Lock	V						V				X	V		V
Déjà Vu	V							V			X		V	V
Blonder		V	V			V				V	X			
VisKey SFR		V	V			V				V	X	V		
Passlogix v-Go		V	V		V				V		X		V	
PassPoints		V	V			V				V	V	V	V	V
DAS		V	V									X		V

V=Yes X=No Blank=not mentioned

4. Conclusion

In the graphical password systems, we need the image big enough to ensure the security. Because of the large enough images can be cut into many enough sub-blocks to meet the users to set their passwords. In a small screen device, the problem is how to provide a large enough password space on a small image. The research of Passpoints suggested that user might be handled by magnification of any area of the chosen image [23]. Our research suggests that user can move the image in small screen by mouse. In the screen, the system will show the given area and when user hold the left button of mouse then user can move the image in this area that means we can put a big image in a small screen device like PDA. The limitation of graphical password system has some important issues. First, the image should be colorful and rich enough, the image should be a big enough to provide large enough password space to keep the security. And when input password may click in the middle of two grids, the fault tolerance can be set to solve this problem. Second issue is efficiency; users to use the mouse to enter a password may be slower than the keyboard. However, graphical password still has value and possibility instead of alphanumeric password. The most important issue is the human memory. People should spend more time learning and practice the graphical password but user's thinking and feeling this kind of graphical password will be much easier than alphanumeric password [23]. Finally, this paper reports a new scheme of graphical password and combines with another technology to improve database of server. In cryptography, security is based on the strength of password. Most passwords belong to alphanumeric password which is developed from symmetric cryptographic algorithm to asymmetric cryptographic algorithm. That shows the importance that saving password is in password-table. We provide a new scheme of graphical password and prove that our scheme can solve the problem of database storage. All information can be steganography to achieve our scheme and can solve the problem of database storage. Furthermore, the information eavesdropping problem during data transmission can be overcome by our tabular steganography technique. Overall, our modified graphical password system can help user easy and friendly to memory their password and without loss of any security of authentication. User's chosen input will hide into image using steganography technology, and transfer to server security without any hacker problem. And then, our authentication server only needs to store a secret key X for decryption instead of large password database.

Acknowledgment. The authors would like to express their appreciation to the anonymous referees for their valuable suggestions and comments. This research was partially supported by the National Science Council, Taiwan, R.O.C., under contract no.: NSC 99-2221-E-030-022.

References

1. A. Perrig and D. Song, "Hash Visualization: A New Technique to Improve Real-World Security," Proceedings of the 1999 International Workshop on Cryptographic Techniques and E-Commerce. (1999)
2. A. Paivio, T. B. Rogers, P. C. Smythe, "Why are pictures easier to recall than words?", *Psychonomic Science*, Vol.11, No.4, pp.137–138. (1999)
3. C. C. Lee, M. S. Hwang, W. P. Yang, "A Flexible Remote User Authentication Scheme Using Smart Cards", *ACM Operating Systems Review*, Vol. 36, No. 3, PP. 46-52. (2002)
4. C. S. Tsai, C. C. Lee, M. S. Hwang, "Password Authentication Scheme: Current Status and Key Issues", *International Journal of Network Security*, Vol.3, No.2, pp.101-115. (2006)
5. D. Hong, S. Man, B. Hawes, and M. Mathews, "A password scheme strongly resistant to spyware", Proceedings of International conference on security and management. Las Vegas, NV. (2004)
6. D. Weinshall, S. Kirkpatrick, "Passwords you'll never forget, but can't recall", Proceedings of CHI 2004 ACM Press, New York, pp. 1399-1402. (2004)
7. G. E. Blonder, "Graphical password", United States Patent 5559961. (1996)
8. H. P. Bahrick, "Semantic memory content in permastore: fifty years of memory for Spanish learned in school", *Journal of Verbal Learning and Verbal Behavior*, Vol. 14, pp. 1-24. (1984)
9. I. Jermyn, A. Mayer, F. Monrose, M. K. Reiter, and A. D. Rubin, "The Design and Analysis of Graphical Passwords", Proceedings of the 8th USENIX Security Symposium. (1999)
10. J.C. Birget, D. Hong, and N. Memon. "Robust Discretization, with an Application to Graphical Passwords". *IEEE Transactions on Information Forensics and Security*, Vol. 1, pp. 395-399. (2006)
11. J. Scholtz, J. Johnson, "Interacting with identification technology: can it make us more secure?", Proceedings of the CHI 2002 Extended Abstracts, ACM Press New York, pp. 564–565. (2002)
12. J. T. Wixted, "The psychology and neuroscience of forgetting", *Annual Review of Psychology*, Vol.55, pp. 235–269. (2004)
13. L. Coventry, A. De Angeli, G. I. Johnson, "Usability and biometric verification", ATM interface, CHI 2003 Proceedings, pp. 153–160. (2003)
14. L. Sobrado and J. C. Birget, "Graphical passwords", *The Rutgers Scholar, An Electronic Bulletin for Undergraduate Research*, vol. 4. (2002)
15. M. A. Sasse, S. Brostoff, D. Weirich, "Transforming the 'weakest link'-a human/computer interaction approach to usable and effective security", *BT Technical Journal* Vol.19, pp.122–131. (2001)
16. M. Kotadia, "Microsoft: Write down your passwords", *ZDNet Australia*, May 23. (2005)
17. Passlogix, "www.passlogix.com", last accessed in June. (2005)

18. R. Dhamija and A. Perrig, "Deja Vu: A User Study Using Images for Authentication", Proceedings of 9th USENIX Security Symposium. (2000)
19. RealUser, "www.realuser.com", last accessed in June. (2005)
20. S. Akula, V. Devisetty, "Image Based Registration and Authentication System", Proceedings of Midwest Instruction and Computing Symposium. (2004)
21. S. Brostoff and M. A. Sasse, "Are Passfaces more usable than passwords: a field trial investigation", People and Computers XIV - Usability or Else: Proceedings of HCI. Sunderland, UK: Springer-Verlag. (2000)
22. S. Man, D. Hong, and M. Mathews, "A shoulder-surfing resistant graphical password scheme - WIW", Proceedings of International conference on security and management, Las Vegas, pp. 105-111. (2003)
23. S. Wiedenbeck, J. Waters, J. C. Birget, A. Brodskiy, and N. Memon, "Passpoints: Design and longitudinal evaluation of a graphical password system", International Journal of Human-Computer Studies, Vol.63, pp.102-127. (2005)
24. T. -S. Chen, C. -C. Chang, and M. -S. Hwang, "Virtual image cryptosystem based upon vector quantization", IEEE Transactions on Image Processing, Vol. 7, No. 10, pp. 1485-1488. (1998)
25. T. Valentine, "An evaluation of the Passface: personal authentication system", Technical Report, Goldsmiths College, University of London. (1998)
26. T. Valentine, "Memory for Passfaces after a Long Delay", Technical Report, Goldsmiths College, University of London. (1999)
27. X. Suo, Y. Zhu, G. Scott. Owen, "Graphical Password: A Survey", Proceedings of the 21th ACSAC IEEE COMPUTER SOCIETY. (2005)
28. M. D. Hafiz, A. H. Abdullah, N. Ithnin, H. K. Mammi, "Towards Identifying Usability and Security Features of Graphical Password in Knowledge Based Authentication Technique", Proceedings of second Asia International Conference on Modelling & Simulation, IEEE Computer Society. (2008)

Tsung-Hung Lin received the B.S. degree in computer science from Tamkang University, Taiwan, Republic of China, in June 1988 and the M.S. and Ph.D. degrees in Computer Science and Information Engineering from the National Chung Cheng University, Taiwan, Republic of China. He joined the faculty of Department of Information Management, Hsing Wu College, Taiwan, Republic of China, as an associate professor in August 2005. Since 2007, he has been an associate professor at the Department of Computer Science and Information Engineering at National Chin-Yi University of Technology, Taiwan, Republic of China. His research interests include wireless communication and mobile computing, the next-generation mobile network and system, wireless personal area network, wireless sensor network, Steganography, and multimedia security.

Cheng-Chi Lee received the B.S. and M.S. in Information Management from Chaoyang University of Technology (CYUT), Taichung, Taiwan, in 1999 and in 2001. He researched in Computer and Information Science from National Chiao Tung University (NCTU), Taiwan, Republic of China, from 2001 to 2003. He received the Ph.D. in Computer Science from National Chung Hsing University (NCHU), Taiwan, in 2007. He was a Lecturer of Computer

and Communication, Asia University, from 2004 to 2007. From 2007, he was an assistant professor of Photonics and Communication Engineering, Asia University. From 2009, he is an Editorial Board member of International Journal of Network Security and International Journal of Secure Digital Information Age. From 2010, he is now an assistant professor of Library and Information Science, Fu Jen Catholic University. His current research interests include information security, cryptography, and mobile communications. Dr. Lee had published over 60+ articles on the above research fields in international journals.

Chwei-Shyong Tsai was born in Changhua, Taiwan, Republic of China, on September 3, 1962. He received the B.S. degree in Applied Mathematics in 1984 from National Chung Hsing University, Taichung, Taiwan. He received the M.S. degree in Computer Science and Electronic Engineering in 1986 from National Center University, Chungli, Taiwan. He received the Ph.D. degree in Computer Science and Information Engineering in 2002 from National Chung Cheng University, Chiayi, Taiwan. From August 2002, he was an associate professor of the Department of Information Management at National Taichung Institute of Technology, Taichung, Taiwan. Since August 2004, he has been an associate professor of the Department of Management Information Systems at National Chung Hsing University, Taichung, Taiwan. Since August 2007, he has been an professor of the Department of Management Information Systems at National Chung Hsing University, Taichung, Taiwan. His research interests include image watermarking, image authentication, information hiding, bio-information and computer networks.

Shin-Dong Guo received the B.S. and M.S. in Photonics and Communication Engineering, Asia University, in 2006 and 2008. His current research interests include information security, cryptography, and mobile communications.

Received: December 23, 2008; Accepted: August 20, 2010.

A Multi-attribute Auction Model by Dominance-based Rough Sets Approach

Rong Zhang^{1,2}, Bin Liu¹, and Sifeng Liu²

¹College of Information and Management Science¹
Henan Agricultural University
450002, Zhengzhou, P.R.China
zr.mis@163.com, liubhnau@163.com

²College of Economics and Management
Nanjing University of Aeronautics and Astronautics
210016, Nanjing, P.R.China
sfliu@nuaa.edu.cn

Abstract. As an alternative to the price-based traditional auction model, the multi-attribute auction model is an integrated model requiring the simultaneous trade of different types of attributes as the sellers and buyers deal. As a result, the design and modeling of the auction mechanism have become very difficult. This paper proposes a multi-attribute auction model using the dominance-based rough sets approach (DRSA). The multi-attribute decision method by DRSA can directly mine out the preference relations between the attributes of alternatives so that relevant auction mechanisms can be designed. This model uses a natural reasoning procedure similar to that of decision makers. Finally, a numerical example demonstrates the simplicity, efficiency, and feasibility of the proposed auction model.

Keywords: multi-attribute decision, rough sets, auction, DRSA.

1. Introduction

An auction is an ancient pricing mechanism that began with Chaldeans(fl. c.340-c.270 B.C.) auctioning their wives, ancient Egyptians auctioning their mining rights, and ancient Romans auctioning their slaves, booties, and debtor's belongings. Research on auction theories began very late, and quantitative studies on auction problems did not commence until the 1960s. Further, although the quantitative study of auction problems is a very complex

* Contact author, Dr. Bin Liu, email:liubhnau@163.com.

process, many interesting conclusions have been reached. For instance, researchers found that an auction is a perfect price discovering mechanism when both the supplier and the buyer are having difficulties in naming a price. The American FCC(Federal Communications Commission) spectrum auction, known as the greatest auction in history, has become the foundation of spectrum auctions in many regions. It has resolved the disjointedness of the auction theory and auction practice, while proving the practicability of the game theory. Furthermore, the auction theory can be used to explain many complex non-economic phenomena (Klemperer 2006).

A traditional auction is the public sale of a product based on price bidding, wherein the highest bidder acquires the product. The seller wants to sell at a greater range of prices, while the buyer wants to buy at a smaller range of prices. Traditional auction problems have been analyzed by the game theory approach for a long time. New auction models have emerged because of developments in the society, especially the advent of information technology. For example, the procurement competition model, which is often used by the US Department of Defense, is unlike most traditional auction models and involves many performance/quality dimension data aside from price, such as promised technical characteristics, delivery date, and managerial performance (Che 1993). The evaluation of bids in a multi-attribute auction involves the application of an elaborate scoring system designed by the buyer: each individual component of a bid is evaluated and assigned a score, the scores are summed to yield a total score, and the firm achieving the highest total score wins the contract.

The use of the Internet as a platform has allowed the inclusion of many novel and useful features in auctions. One of these features is the incorporation of multiple attributes, which is based on the realization that there is much more to value than price (Teich et al. 2006). By the B2B mode, multi-attribute online auction is a new type of auction on the Internet that has risen with the development of e-commerce in recent years. Through the dominance of information in the Internet, online auctions can help both the buyer and the seller in expanding their range for finding individuals to cooperate with and in ensuring that an approved price for both sides is settled on. Recent trends have required that a product cannot be auctioned simply by its price. Instead, the multi-attribute auction, which is based on the multiple attributes of a product, such as its price, quality, time of delivery, and management performance of the supplier, has become de rigeur. At present, multi-attribute online auctions have been applied in public biddings of government projects and in the procurement of enterprises. Thus, multi-attribute trading mechanisms transcend traditional, price-only mechanisms by facilitating a negotiation over a set of predefined attributes representing various non-price aspects of the product (Engel and Wellman 2007).

In this paper, a novel multi-attribute online auction model is developed based on the dominance-based rough set approach (DRSA). Similar to the natural reasoning learning method, the new method can reuse the agent's preferences to greatly increase bidding efficiency.

The rest of this paper is organized as follows. Section 2 is the literature review. Section 3 will advance the multi-attribute decision model by dominance-based rough sets approach. Section 4 will develop a new-type multi-attribute auction model by dominance-based rough sets approach, Section 5 is an example, and the last section provides concluding remarks.

2. Related Literature

Che (1993), the first to study the multi-attribute auction problem, designed a score rule for integrating two attributes and the auction mechanism, and proved the two-dimensional revenue equivalence theorem. Branco (1997) extended Che's study by taking into account auction problems correlated with the firm's cost, and concluded that the optimal effect cannot be directly obtained through the auction process but by a two-stage mechanism. Bichler (2000) carried out experimental analyses of the similar utility function and found that multi-attribute auctions yield utility scores that are significantly higher than those of single-attribute auctions. Based on two attributes and existing auction algorithms, Teich et al. (2000a, 2006b) described a Java-based Web auction system. They designed a "suggested price" mechanism and introduced its algorithms and theoretical foundations. Mishra and Veeramani (2002) identified multi-attribute auction problems in the outsourcing industry. They developed a simple and feasible auction mechanism proven to lead to competitive equilibrium while estimating the private cost functions of suppliers and the private valuation functions of the parent-firms based on the attribute values to be known. Karimi et al. (2007) stated that multi-attribute procurement problems have different characteristics under the supply chain framework. The exact value of the production cost is kept private and is difficult to be known by producers and customers. They designed an auction mechanism that integrated two attributes (time and price) into the scoring rule. Engel and Wellman (2007) acknowledged that trader's preferences cannot be ignored and that the full additivity of attributes cannot be assumed. Therefore, they introduced an iterative auction mechanism that maintained the prices in local clusters related to attributes rather than the full space of joint configurations. Jin et al (2006a, 2006b) introduced the MAV(Multi-Attribute auction model pioneered by Vickrey) and MAE(Multi-Attribute auction model pioneered by Esther David's) models for multi-attribute auctions and compared them with existing models to demonstrate the advantages of the new models. However, they did not constrain the number of attributes, and they assumed that both the cost functions of the seller and the valuation functions of the buyer are known. Another study (Wang et al. 2006) used the incentive mechanism to design a franchise bidding mechanism for a regional district distribution service to maximize the expected social welfare. However, the format of the optimal bidding mechanism is so complicated that a two-stage practical application is needed to implement it, namely, to bid first and then to negotiate about the quality.

Rong Zhang, Bin Liu, and Sifeng Liu

Most extant studies often integrate multidimensional attributes to two-dimensional attributes, including price and quality, by first predigesting multiple attributes (except for price) into a quality attribute. Then the characteristics of the commodity are mapped from multidimensional to one-dimensional real number space by the valuation function, score rule, cost function, or social welfare function. Finally, the multi-attribute auction format is studied using the game theory and object optimization. However, the auction models developed from two-dimensional attributes are already complex and difficult to implement in practical applications. In management applications, the cost and social welfare functions are private for the seller and the buyer, so that the mapping from multidimensional space to single dimension space is extremely focal and difficult. Most studies examine the multi-attribute auction from the view of mechanism design, generally predigesting this kind of problem with the assumption that the valuation function is known and failing to consider it as the focus of the study.

The design of scoring rules for multiple attributes and the identification of the influence of different scoring rules on auction performance are among the most important study directions of multi-attribute auction problems. Based on the multi-attribute decision theory, multi-attribute auction studies should not constrain the number and types of attributes (whether qualitative or quantitative) of the commodity. A study (Wu, 2007) proposed the use of the multi-attribute decision making of the weighted aggregative valuation method to choose the winning bidder and introduced full bidding rules. During multiple rounds of bidding, suppliers and buyers are permitted to continually adjust the weights of attributes and focus on the importance of the assignment of weights. Xie and Li (2005) evaluated bidding alternatives in multi-attribute online auction decisions with the fuzzy aggregative valuation method. They also made attribute reductions and set the weights of attributes using the rough set theory. Other than these two methods, the rest of studies proposed that multiple attributes should be aggregated first before evaluating bidding based on aggregative valuation.

Other studies also paid more attention to agent preferences. In multiple rounds of bidding and negotiation under limited time, buyers or sellers may follow their own preferences. If the preferences of the selected winner in the previous round can be extracted, they can be used to predict the optimal bidding in the next round, thus increasing substantially the efficiency of the multi-attribute auction. At present, the rough set method is preferred for extracting decision preferences. Furthermore, each attribute in a multi-attribute auction has ordinal properties in addition to classifiable characteristics. Hence, based on the ordinal properties of attributes, the dominance-based rough set theory can be used to solve multi-attribute auction problems. There have been many successful applications of the dominance-based rough set theory to multi-attribute decision fields.

3. Multi-attribute decision model by DRSA

3.1. Information system and the composition of the Pairwise Comparison Table (PCT)

A multi-attribute decision problem, which has a finite set of alternatives $B = \{x_1, x_2, \dots, x_n\}$, can be treated as an information system. Generally, an information system is denoted as $S = (U, A, V, f)$, where U is a domain of discourse, so that the set of alternatives is $B \subset U$; A is the set of attributes; $V = \bigcup_{a \in A} V_a$, where V_a is the actual range of attribute a ; and f is the information function, and $f: U \times A \rightarrow V$ makes $f(x, a) \in V_a$ for $\forall x \in U, a \in A$.

For $\forall a \in A, T_a$, denoted on set B , is a finite set of duality relations in light of attribute a , such that $\forall (x, y) \in B \times B$ uniquely ensures a dual relation $\tau \in T_a$. PCT is a defined data sheet for $S_{PCT} = (B \times B, A, T_a, g)$, where $B \times B$ is the set of pairwise comparisons; $T_A = \bigcup_{a \in A} T_a$; and $g: (B \times B) \times A \rightarrow T_A$ is an information function that yields $\forall (x, y) \in B \times B, a \in A, g[(x \times y), a] \in T_a$.

Assuming that the synthetic preference information of a small part of alternatives E ($E \subset B$) is known, and then PCT is a decision-making table, namely, $D_{PCT} = (E \times E, C \cup \{d\}, T_c \cup T_{(d)}, g)$, where E is a set of reference objects, and $E \times E$ is its set of pairwise comparisons, attribute set A consists of condition attribute set C and decision-making attribute $\{d\}$ of synthetic pair wise comparison, and $C \cap \{d\} = \Phi, C \cup \{d\} = A; T_c = \bigcup_{a \in C} T_a, T_a = \{P_a^h, h \in H_a\}, p_a^h$ showing preference grades produced as alternatives in pairs for attribute a is compared, and H_a is a special set describing preference grades for attribute a .

For $\forall (x, y) \in E \times E, T_d = \begin{cases} S & x \succ y \\ S^c & \text{otherwise} \end{cases}$,

where ' \succ ' shows that x is at least as good as y ; g is an information function and shows that $g[(x \times y), a \cup d] \in T_a \cup T_d$ for $\forall (x, y) \in E \times E, \forall a \in A$.

3.2. Determining the preference intension of multiple-grade dominance for attributes

Greco et al. (1999) put forward SPCT analysis by single grade dominance relation and assumed that the preference of all criteria has the same intension. This model is convenient for approximating and drawing decision rules, but is not precise enough. Taking different intensions of preference dominance relation into account for each criterion can yield a more accurate and real PCT (Wang et al. 2006).

The set of attributes $A = \{a_1, a_2, \dots, a_n\}$ is where each attribute has its preference. Set A is divided into the set of condition attributes C and decision attribute $\{d\}$, such that $C \cap \{d\} = \Phi$, $C \cup \{d\} = A$, where the decision attribute is the synthetic valuation result of alternatives, reflecting the integrated preference relation of alternatives. Further, the decision attribute is quantitative and can be an index whose value is a cardinal number reflecting the integrated valuation of alternatives, or an ordinal number reflecting the ranking result of alternatives. A condition attribute that is a cardinal number shows a specific occurrence of the attribute for the alternative and should take the value in the actual range. Another form of condition attribute is the qualitative index of linguistic that describes the type.

The condition attributes of alternatives all have some preference relations, and different attributes have different preference intensions. Assume that the pair wise comparison of alternatives for each attribute can be shown by gradation preference P_a^h , namely, for $\forall a \in A$, $(x, y) \in E \times E$, and $h \in H_a$, $x P_a^h y$ shows that alternative x in light of attribute a dominates alternative y by the intension of h ; when $h > 0$, it shows that alternative x dominates alternative y , when $h = 0$, it shows that alternative x does not vary against alternative y , and when $h < 0$, it shows that alternative x is dominated by alternative y . H_a is the set of preference intensions generated as alternatives that are compared in pairs, and is confirmed through the following process:

For information system $S = (U, A, V, f)$, first ensure an increasing preference function concerning attribute $a \in A$, $r_a: E \rightarrow R$. When an attribute is a numerical value, for the alternative $x \in E$, there is $r_a(x) = f(x, a)$; when an attribute is a linguistic description, the set of comments is V_a , there is $r_a(x) = k$, and $k \in \{1, 2, \dots, |V_a|\}$. For example, if attribute a is a price, the values of the attributes for alternatives x_1 , x_2 , and x_3 , respectively, are 20000, 18000, and 21000, namely, $f(x_1, a) = 20000$, $f(x_2, a) = 18000$, and $f(x_3, a) = 21000$, and then $r_a(x_1) = f(x_1, a) = 20000$, $r_a(x_2) = f(x_2, a) = 18000$, and $r_a(x_3) = f(x_3, a) = 21000$; if attribute a is the quality of a commodity,

$V_a = \{good, medium, bad\}$, $|V_a|=3$, namely, there are three comments in the set of comments, and the attribute values of alternatives x_1, x_2, x_3 are good, medium, and bad, respectively. Specifically, $f(x_1, a) = 'medium'$, $f(x_2, a) = 'bad'$, and $f(x_3, a) = 'good'$. r_a is an increasing preference function, so the value of a function should increase with increasing preference, such that $r_a(x_1) = 2$, $r_a(x_2) = 1$, and $r_a(x_3) = 3$.

Afterwards, the preference intension function is denoted as

$$h_a(r_a(x), r_a(y)) = \text{sgn}(r_a(x) - r_a(y)) \cdot \begin{cases} 0 & 0 \leq |r_a(x) - r_a(y)| \leq \Delta_{a1} \\ 1 & \Delta_{a1} < |r_a(x) - r_a(y)| \leq \Delta_{a2} \\ \dots\dots \\ k & |r_a(x) - r_a(y)| > \Delta_{ak} \end{cases}$$

where Δ_{ak} is the threshold of the attribute properly given by the decision maker before the multi-attribute decision-making method. The Delphi method and analytic hierarchy process are the usual methods to decide on the threshold. The threshold can also be chosen according to industrial standards and national regulation. In the above example, since the attribute is price, and the threshold is $\Delta_{a1} = 2000$, then the preference grades of the pairwise comparisons of alternative are $h_a(r_a(x_1), r_a(x_1)) = 0$, $h_a(r_a(x_1), r_a(x_2)) = 1$, $h_a(r_a(x_1), r_a(x_3)) = 0$, and $h_a(r_a(x_2), r_a(x_1)) = -1$; therefore, $x_1 p_a^0 x_1$, $x_1 p_a^1 x_2$, $x_1 p_a^0 x_3$, and $x_2 p_a^{-1} x_1$ respectively show that according to attribute a , alternative x_1 is similar to alternative x_1 , alternative x_1 dominates alternative x_2 with the intension of 1, alternative x_1 is similar to alternative x_3 , and alternative x_2 does not dominate alternative x_1 with the intension of 1.

When the attribute is the quality of the commodity, and the thresholds are $\Delta_{a1} = 1$, $\Delta_{a2} = 2$, then the preference grades of the pairwise comparison of the alternative are $h_a(r_a(x_1), r_a(x_1)) = 0$, $h_a(r_a(x_1), r_a(x_2)) = 1$, $h_a(r_a(x_1), r_a(x_3)) = -1$, and $h_a(r_a(x_2), r_a(x_3)) = -2$; therefore, $x_1 p_a^0 x_1$, $x_1 p_a^1 x_2$, $x_1 p_a^{-1} x_3$, and $x_2 p_a^{-2} x_3$ respectively show that alternative x_1 is similar to alternative x_1 , alternative x_1 dominates alternative x_2 with the intension of 1, alternative x_1 does not dominate alternative x_3 with the intension of 1, and alternative x_2 does not dominate alternative x_3 with the intension of 2 according to attribute a .

The above attributes discussed are the types of profit. As one attribute is the type of cost, it can be translated into the type of profit and then denoted as

preference intension grades, or denoted as the preference intension function as follows:

$$h_a(r_a(x), r_a(y)) = (-1) \cdot \text{sgn}(r_a(x) - r_a(y)) \cdot \begin{cases} 0 & 0 \leq |r_a(x) - r_a(y)| \leq \Delta_{a1} \\ 1 & \Delta_{a1} < |r_a(x) - r_a(y)| \leq \Delta_{a2} \\ \dots\dots \\ k & |r_a(x) - r_a(y)| > \Delta_{ak} \end{cases}.$$

3.3. Dominance relation based on PCT

For $\forall(x, y), (w, z) \in E \times E$, in light of $a \in A$, $(x, y)D_a(w, z)$ shows that (x, y) dominates (w, z) , or the intension that x dominates to y is at least the same as that when w dominates z . Then $\forall(x, y), (w, z) \in E \times E$, $a \in A$, $h_a, k_a \in H_a$, $xP_a^h y$, and $wP_a^k z$ will yield $(x, y)D_a(w, z) \Leftrightarrow h_a \geq k_a$.

Based on D_a dominance relation, positive dominance $D_A^+(x, y)$ and negative dominance $D_A^-(x, y)$ can be introduced:

$$D_A^+(x, y) = \{(w, z) \in E \times E \mid (w, z)D_a(x, y)\},$$

$$D_A^-(x, y) = \{(w, z) \in E \times E \mid (x, y)D_a(w, z)\}.$$

The dual-dimension relation S and S^C defined on E about decision attribute $\{d\}$ can be considered. Regarding attribute set A , the upper approximation and the lower approximation of S are respectively defined by

$$\underline{C}(S) = \{(x, y) \in E \times E \mid D_A^+(x, y) \subseteq S\},$$

$$\overline{C}(S) = \{(x, y) \in E \times E \mid D_A^-(x, y) \cap S \neq \Phi\} = \bigcup_{(x, y) \in S} D_A^+(x, y),$$

while the upper approximation and the lower approximation of S^C are respectively defined as

$$\underline{C}(S^C) = \{(x, y) \in E \times E \mid D_A^-(x, y) \subseteq S^C\},$$

$$\overline{C}(S^C) = \{(x, y) \in E \times E \mid D_A^+(x, y) \cap S^C \neq \Phi\} = \bigcup_{(x, y) \in S^C} D_A^-(x, y).$$

The following complementary properties come into existence: $\underline{C}(S) = E \times E - \overline{C}(S^C)$, $\overline{C}(S) = E \times E - \underline{C}(S^C)$, $\underline{C}(S^C) = E \times E - \overline{C}(S)$, and $\overline{C}(S^C) = E \times E - \underline{C}(S)$.

The boundaries of S and S^C are respectively denoted as follows:

$$bn(S) = \overline{P}(S) - \underline{P}(S), \quad bn(S^C) = \overline{P}(S^C) - \underline{P}(S^C), \quad \text{and} \quad bn(S) = bn(S^C).$$

3.4. Acquiring the decision rules for multi-grade dominance

The upper accumulating preference and the lower accumulating preference are denoted as

$$\forall(x, y) \in E \times E, a \in A, h, k \in H_a$$

Such that when $k \geq h$ and $xP_a^k y$, then $xP_a^{\geq h} y$; when $k \leq h$, and $xP_a^k y$, then $xP_a^{\leq h} y$.

In terms of rough set approximation, the following three kinds of decision rules are derived from the appointed PCT:

$$P = \{a_1, a_2, \dots, a_p\} \subseteq A, (h_{a_1}, h_{a_2}, \dots, h_{a_p}) \in H_{a_1} \times H_{a_2} \times \dots \times H_{a_p},$$

D_{\geq} Decision rules: if $xP_{a_1}^{\geq h_{a_1}} y$ and $xP_{a_2}^{\geq h_{a_2}} y$,, and $xP_{a_p}^{\geq h_{a_p}} y$, then $xS y$, and supported by objects in pairs of $\underline{C}(S)$.

D_{\leq} Decision rules: if $xP_{a_1}^{\leq h_{a_1}} y$ and $xP_{a_2}^{\leq h_{a_2}} y$,, and $xP_{a_p}^{\leq h_{a_p}} y$, then $xS^c y$, and supported by objects in pairs of $\underline{C}(S^c)$.

$D_{\geq \leq}$ Decision rules: if $xP_{a_1}^{\geq h_{a_1}} y$ and $xP_{a_2}^{\geq h_{a_2}} y$,, and $xP_{a_k}^{\geq h_{a_k}} y$, and $xP_{a(k+1)}^{\leq h_{a(k+1)}} y$ and $xP_{a(k+2)}^{\leq h_{a(k+2)}} y$,, and $xP_{a_p}^{\leq h_{a_p}} y$, then $xS y$ or $xS^c y$, and supported by objects in pairs of $bn_C(S)$.

The decision rules above acquired on E are used for the whole set of alternatives B . Net flow value $S(x)$ is computed for each alternative $x \in B$ and can be used to rank and choose alternatives, where

$$S(x) = S^{++}(x) - S^{+-}(x) + S^{-+}(x) - S^{--}(x),$$

$$S^{++}(x) = \text{card}(\{y \in B \mid \text{at least one decision rule exists to supports } xS y\});$$

$$S^{+-}(x) = \text{card}(\{y \in B \mid \text{at least one decision rule exists to supports } yS x\});$$

$$S^{-+}(x) = \text{card}(\{y \in B \mid \text{at least one decision rule exists to supports } xS^c y\});$$

$$S^{--}(x) = \text{card}(\{y \in B \mid \text{at least one decision rule exists to supports } yS^c x\}).$$

The alternative $x^* \in B$ satisfying $S(x^*) = \max_{x \in B} S(x)$ is the most optimal alternative.

4. Multi-attribute auction model by DRSA

4.1. Decision model

First, the qualification of suppliers for bidding is evaluated with the multi-attribute decision-making method. All suppliers that submitted bidding reports

are ranked, and the former m suppliers can be chosen to qualify for the later bidding. The attribute set of the required commodity is $C = \{a_1, a_2, \dots, a_n\}$. The actual range of attribute a is denoted as V_a , and $V = \bigcup_{a \in C} V_a$. Thus, the whole bidding process is seen as the information system $S = (U, C, V, f)$, where U is the domain of discourse. The bidding alternative set B consists of the alternatives bided by m bidders each round, $B \subset U$; and f is an information function where $f: U \times C \rightarrow V$.

The attributes of the procured commodity are taken as condition attribute set C , assuming that there are no attributive redundancies. Decision attribute $\{d\}$ is obtained from the ranking result of m suppliers. This yields the comprehensive preference information of E , a small part of alternatives in the full information system. After confirming the preference grade set H_a of each condition attribute, PCT can be structured as $D_{PCT} = (E \times E, C \cup \{d\}, T_c \cup T_{\{d\}}, g)$, and this datasheet is a decision table. Moreover, the decision table derived through this method is rational and effective because it is based on the information of each bidder's bidding alternative, and it integrated the characters of commodity required by the buyer. The decision rules are derived in terms of DRSA from a small part of PCT for which the comprehensive preference information is known, and then are used on the bidders' alternatives in each round to obtain the decision value of pair wise comparison. Finally, the ranking of alternatives and the selection of the optimal alternative is implemented with scoring functions. The bidding involves single goods and multiple attributes. Assume that there are R rounds of bidding, or that the bidding is constrained by time T . Before the r -th ($r \in \{1, 2, \dots, R\}$) round begins, the procurer declares the bidding result of the previous round. Assume that the optimal bidding alternative of the r -th round is $b^{*(r)} = (a_1^{*(r)}, a_2^{*(r)}, \dots, a_n^{*(r)})$, the worst being $b^{-(r)} = (a_1^{-(r)}, a_2^{-(r)}, \dots, a_n^{-(r)})$. Based on the suppliers' actual bidding on $r-1$ round, the procurer declares the reference alternative $b_0^r = (a_{01}^r, a_{02}^r, \dots, a_{0n}^r)$ and $b_0^r = b^{*(r-1)}$, which reflects his/her own bidding preference, with the worst alternative and preference grade thresholds Δ_{ak}^r for each condition attribute $a \in C$. Going along with the bidding, the procurer can modify Δ_{ak}^r according to the actual bid of each supplier.

Assume that there are m^r bidders on r round bidding, and $m^r \leq m$. B^r is the bidding alternative set composed of m^r bidders' alternatives on r round. b_k^r is the bidding value of bidder k on the r -th round, and $b_k^r = (a_{k1}^r, a_{k2}^r, \dots, a_{kn}^r)$, $S(b_k^r)$ is the net flow value, then

$$S(b_k^r) = S^{++}(b_k^r) - S^{+-}(b_k^r) + S^{-+}(b_k^r) - S^{--}(b_k^r),$$

Where

$$S^{++}(b_k^r) = \text{card}(\{b_j^r \in B^r \mid \text{at least one decision rule exists to supports } b_k^r S b_j^r\});$$

$$S^{+-}(b_k^r) = \text{card}(\{b_j^r \in B^r \mid \text{at least one decision rule exists to supports } b_j^r S b_k^r\});$$

$$S^{-+}(b_k^r) = \text{card}(\{b_j^r \in B^r \mid \text{at least one decision rule exists to supports } b_j^r S^c b_k^r\});$$

$$S^{--}(b_k^r) = \text{card}(\{b_j^r \in B^r \mid \text{at least one decision rule exists to supports } b_k^r S^c b_j^r\}).$$

To rank the bidding alternatives in set B^r for each round according to function $S(x)$, and to select the optimal alternative on this round $b^{*(r)} = (a_1^{*(r)}, a_2^{*(r)}, \dots, a_n^{*(r)})$, making $S(b^{*(r)}) = \max_{b^r \in B^r} \{S(b^r)\}$.

On the r -th round, the effective bid of bidder k is $b_k^r = (a_{k1}^r, a_{k2}^r, \dots, a_{kn}^r)$ and satisfies the following conditions:

- (1) $\exists j \in \{1, 2, \dots, n\}$ making $a_{kj}^r > a_{0j}^r$;
- (2) $\text{card}(M) \geq \text{card}(N)$, where $M = \{a_{kj}^r \mid a_{kj}^r > a_{kj}^{r-1}, j = 1, 2, \dots, n\}$ and $N = \{a_{kj}^r \mid a_{kj}^r < a_{kj}^{r-1}, j = 1, 2, \dots, n\}$.

If the bidder gives up bidding on any round, then there is the assumption that he/she will not participate in subsequent biddings.

4.2. The decision-making steps

The process of bidding based on the dominance relation rough multi-attribute decision-making model is as follows:

Step 1: For the supplier, perform a primary election with the multi-attribute decision theory regarding the bidding alternatives of the commodity to select m qualified suppliers for bidding. Use the ranking result of m suppliers as the decision attribute to construct PCT.

Step 2: Based on PCT reflecting the bidding information of suppliers, derive four kinds of dominance decision rules using DRSA.

Step 3: Before the r -th round begins, the buyer declares his/her preference alternative b_0^r and the preference grade thresholds Δ_{ak}^r of each condition attribute.

Step 4: Suppliers carry through the r -th round bidding. First, the decision maker judges the valid bid and changes the information of the bidding alternative into PCT, obtains the decision value of each alternative on the present round with dominance decision rules to rank all alternatives, and derives the optimal alternative $b^{*(r)}$ on the round.

Step 5: Continue the process until time T or the R round bid ends. The optimal alternative on the last round is the optimal procurement alternative, and its bidder is the winning bidder.

Step 6. End.

5. Example

Assume that a firm needs to procure one commodity and has to select one supplier. The firm not only focuses on price but also on other faculties, such as capability for research and development, credit standing, servicing level, and financing status. The firm adopts a single-good and multi-attribute auction mode to choose the proper supplier. Information is publicized to invite public bidding, and then experts are invited to filter suppliers with the multi-attribute decision-making method to choose the qualified suppliers that should participate in bidding. The attribute set of commodity is

$$C = \{\text{Price, Delivery Time, Ability of Study and Development, Accounting Manner}\}.$$

Three suppliers are primarily filtered out to participate in the bidding and construct the decision table as shown in Table 1.

Table 1: Decision Table of the Qualified Suppliers

	Price (\$)	Delivery Time (Month)	Ability of Study and Development	Accounting Manner	Ranking
x1	150	6	Good	Cash on Delivery	3
x2	175	5	Medium	Advance Payment	1
x3	135	4	Bad	Advance Payment	2

In Table 1, Supplier 2 has the highest price and is ranked as the first option since the buyer focuses on delivery time and scientific research level for the required commodity.

Table 2: The Standardized Decision Table and the Threshold of Attribute

	Price	Delivery Time	Ability of Study and Development	Accounting Manner	Ranking
x1	150	6	3	2	3
x2	176	5	2	1	1
x3	135	4	1	1	2
Δ_{a1}^0	20	0.9	0	0	
Δ_{a2}^0	40	1.9	1		

In Table 2, the linguistic attribute is changed into a quantitative attribute and price, the delivery time is the converse-attribute such as cost and capability for study and development, and the accounting manner is the positive attribute. The preference grade thresholds for each attribute are determined by experts for the pair wise comparisons of alternatives. The PCT, as shown in Table 3, is constructed.

Table 3: Pairwise Comparison Table

	Price	Delivery Time	Ability of Study And Development	Accounting Manner	Decision Result
(x1, x1)	0	0	0	0	S
(x1, x2)	1	-1	1	1	S ^C
(x1, x3)	0	-2	2	1	S ^C
(x2, x1)	-1	1	-1	-1	S
(x2, x2)	0	0	0	0	S
(x2, x3)	-2	-1	1	0	S
(x3, x1)	0	2	-2	-1	S
(x3, x2)	2	1	-1	0	S ^C
(x3, x3)	0	0	0	0	S

Finally, the decision rules are extracted as follows when $\forall(x, y) \in B$:

D_{\geq} Decision rules:

Rule 1 if $h_{a1} \geq 0$ and $h_{a2} \geq 0$, and $h_{a3} \geq 0$, and $h_{a4} \geq 0$, then xSy ;

Rule 2 if $h_{a1} \geq 0$ and $h_{a2} \geq 2$, then xSy ;

D_{\leq} Decision rules:

Rule 3 if $h_{a1} \leq 0$ and $h_{a2} \leq -2$, then $xS^C y$;

$D_{\geq \leq}$ Decision rules:

Rule 4 if $h_{a1} \geq 1$, and $h_{a2} \leq -1$, and $h_{a4} \geq 1$, then xSy or $xS^C y$;

Rule 5 if $h_{a1} \leq -1$, and $h_{a2} \geq 1$, and $h_{a4} \leq -1$, then xSy or $xS^C y$;

Rule 6 if $h_{a1} \geq 1$, and $h_{a2} \leq 1$, and $h_{a3} = -1$, then xSy or $xS^C y$;

Rule 7 if $h_{a1} \leq -1$, and $h_{a2} \geq -1$, and $h_{a3} \geq 1$, then xSy or $xS^C y$.

Before the first round begins, the buyer publicizes his/her satisfied bidding

$b^{*(0)} = (176, 5, \text{Medium}, \text{Advance Payment})$,

the worst bidding

$b^{-(0)} = (150, 6, \text{Good}, \text{Cash on Delivery})$,

and the preference grade thresholds of the alternative pairwise comparisons for each attribute in Table 2.

The third suppliers' bidding alternatives on the first round are indicated in Table 4. It also shows the results of the net flow values computed with the decision rules.

Table 4: The bidding alternatives on the 1 round, ranking result and thresholds of attribute

	Price	Delivery Time	Ability of Study And Development	Accounting Net Manner	Net Flow Value	Ranking
x1	147	5	3	2	2	1
x2	161	5	2	1	-4	3
x3	153	4	2	1	2	1
Δ_{a1}^0	5	0.5	0	0		
Δ_{a2}^0	10	1.5	1			

Before the second round of bidding, the buyer declares his/her most satisfied bidding alternatives

$$b^{*(1)} = (147, 5, \text{Good}, \text{Cash on Delivery})$$

$$b^{*(1)} = (153, 4, \text{Medium}, \text{Advance Payment})$$

and the worst bidding

$$b^{-(1)} = (161, 5, \text{Medium}, \text{Advance Payment}) .$$

The preference grade thresholds of the alternative pairwise comparisons for each attribute are presented in Table 4.

Table 5 presents the third suppliers' bidding alternatives on the second round. The net flow values computed with the decision rules are also shown.

Table 5: The bidding alternatives on the 5 round and ranking result

	Price	Delivery Time	Ability of Study And Development	Accounting Net Manner	Net Flow Value	Ranking
x1	144	5	3	2	3	1
x2	153	5	2	2	-2	3
x3	138	4	2	1	0	2

If the bidding ends at this time, Supplier 1 is the winner, and the winning bidding is

$$b^{*(2)} = b_1^2 = (144, 5, \text{Good}, \text{Cash on Delivery}) .$$

6. Conclusion

This paper proposes a multi-attribute decision method based on multi-grade DRSA to select the winner of a multi-attribute auction. Previously, bidding alternatives were evaluated using the fuzzy aggregative valuation method. An in-depth analysis of this model revealed that some of its characteristics, such as the reasoning process, are similar to those of a human mind. Further, the auction rules favor the declaration of the true preference information of the

seller and the buyer. This algorithm is simple and can easily be programmed for applications. The range of possible applications of the multi-attribute decision-making method and DRSA can be explored further. Future studies can analyze and design timely dynamic multi-attribute auction decision-making models based on the model developed in this paper.

Acknowledgement. The authors thank the editor and anonymous referees for their helpful comments and suggestions, which have improved the exposition of this paper. Additionally, this paper was supported in part by Natural Science Foundation of China under Grant No.70871035, and the Program for Science & Technology Innovation Talents in Universities of Henan Province under Grant No.2010HASTIT030.

References

1. Bichler. An Experimental Analysis of Multi-attribute Auctions. *Decision Support Systems*, 29(3): 249-268. (2000),
2. Branco. The Design of Multidimensional Auctions. *RAND Journal of Economics*, 28(1): 63-81. (1997),
3. K. Che. Design Competition through Multidimensional Auctions. *RAND Journal of Economics*, 24(4): 668-680. (1993).
4. Engel, M. P. Wellman. Generalized Value Decomposition and Structured Multi attribute Auctions. *Proceedings of the 8th ACM Conference on Electronic Commerce*, 227-236.(2007)
5. X. Jing, C. Y. Shi. An Ascending Bid Multi-attribute Auction Method. *Chinese Journal of Computers*, 29(1), 145-152. (2006)
6. X. Jing, C. Y. Shi. An Ascending Bid Multi-attribute Auction Method. *Journal of Computer Research and Development*, 43(7), 1135-1141. (2006)
7. R. Karimi, C. Lucas. B. Moshiri. New Multi Attributes Procurement Auction for Agent-based Supply Chain Formation. *International Journal of Computer Science and Network Security*, 7(4): 255-261. (2007).
8. Paul Klemperer. *Auctions: Theory and Practice*. Beijing: China Renmin University Press, 2006, 3.
9. D. Mishra, D. Veeramani. A Multi-attribute Reverse Auction for Outsourcing. *Proceedings of the 13th International Workshop on Database and Expert Systems Applications*, 675-679. (2002).
10. J. Teich, H. Wallenius, J. Wallenius, A. Zaitsev. An Internet-Based Procedure for Reverse Auctions Combining Aspects of Negotiations and Auctions. *Proceedings of DEXA*. 1010-1014.(2000).
11. J. Teich, H. Wallenius, J. Wallenius, A. Zaitsev. A Multi-attribute E-auction Mechanism for Procurement: Theoretical Foundations. *European Journal of Operational Research*, 175(1): 90-100. (2006)
12. J. Wang, S. Y. Liu, J. Zhang. Rough Set Approach to Group Decision Making Based on Linguistic Information Processing. *Journal of System Engineering*, 21(1), 18-23.(2006).
13. X. J. Wang, D. Han, H. Yin, D. B. Fang. Mechanism Design for Franchise Bidding in Regional District Distribution Service. *Proceeding of the CSEE*, 26(20), 39-44.(2006)

Rong Zhang, Bin Liu, and Sifeng Liu

14. G..Wu. Hierarchical Interactive Winner Bidder Selection Approach in Multi-Attribute E-auction. *Journal of Management Sciences*, 20(3), 55-60.(2007)
15. S. Xie, Y. J. Li. Multi-attribute Decision Making Based on Fuzzy Rough Set in E-Auction. *Systems Engineering Theory Methodology Applications*, 14(2): 182-184.(2005).
16. S. Greco, B. Matarazzo, R, Slowinski. Rough approximation of preference relation by dominance relations. *European Journal of Operational Research*, 117(1): 63-83.(1999).

Rong Zhang, received her PhD in system engineering from Nanjing University of Aeronautics and Astronautics(NUAA), Nanjing, China in 2010. Now she is an associate professor in department of management science, Henan Agricultural University(HAU). Her main research interests include information system, decision analysis, and grey system theories.

Bin Liu, received his PhD in management science and engineering from NUAA, Nanjing, China in 2005. Now he is an associate professor of HAU. His main research interests include information system, supply-chain management, and grey system theories.

Sifeng Liu, received his PhD in system engineering, from Huazhong University of Science and Technology, in 1986 and 1998, respectively. Now, he is a distinguished professor at NUAA. His main scientific research activities are in grey system and econometrics.

Received: August 04, 2009; Accepted: July 16, 2010.

Design of Median-type Filters with an Impulse Noise Detector Using Decision Tree and Particle Swarm Optimization for Image Restoration

Bae-Muu Chang^{1,2}, Hung-Hsu Tsai^{*3}, Xuan-Ping Lin³, and Pao-Ta Yu¹

¹ Department of Computer Science and Information Engineering,
National Chung Cheng University, Min-Hsiung, Chia-Yi 621, Taiwan, R.O.C.
bmchang@cs.ccu.edu.tw (B.-M. Chang)
csipy@cs.ccu.edu.tw (P.-T. Yu)

² Department of Information Management,
Chienkuo Technology University, Chang-Hua, Chang-Hua 500, Taiwan, R.O.C.
bmchang@cc.ctu.edu.tw

³ Department of Information Management,
National Formosa University, Hu-Wei, Yun-Lin 632, Taiwan, R.O.C.
thh@nfu.edu.tw (H.-H. Tsai)
no300601@gmail.com (X.-P. Lin)

Abstract. This paper proposes the median-type filters with an impulse noise detector using the decision tree and the particle swarm optimization, for the recovery of the corrupted gray-level images by impulse noises. It first utilizes an impulse noise detector to determine whether a pixel is corrupted or not. If yes, the filtering component in this method is triggered to filter it. Otherwise, the pixel is kept unchanged. In this work, the impulse noise detector is an adaptive hybrid detector which is constructed by integrating 10 impulse noise detectors based on the decision tree and the particle swarm optimization. Subsequently, the restoring process in this method respectively utilizes the median filter, the rank ordered mean filter, and the progressive noise-free ordered median filter to restore the corrupted pixel. Experimental results demonstrate that this method achieves high performance for detecting and restoring impulse noises, and outperforms the existing well-known methods.

Keywords: Impulse noise detector, Decision tree, Particle swarm optimization, Median-type image filter, Noise removal.

1. Introduction

Digital images are sometimes contaminated by impulse noises during transmission processes. These noises usually degrade the image processing results severely such as edge detection, image segmentation, and object recognition. Thus, it is urgently essential to remove noises before the image

processing procedures are performed. For this purpose, several well-known filters have been presented in the past years. Nonlinear filters are usually better than linear filters because of their good impulse noise removal and edge preservation [1]. The median filter which is a nonlinear filter is the popular well-known and the most often employed for removing impulse noises [1]. Although it is effective in filtering impulse noises, the median filter still blurs some fine details and often damages edges when it is applied to the digital images uniformly.

Hence, various modified median-based filters have been proposed to enhance the typical median filter in the recent years, such as the weighted median (WM) filter [2], the tri-state median (TSM) filter [3], the progressive switching median (PSM) filter [4], the center weighted median (CWM) filter [13], the adaptive center weighted median (ACWM) filter [5], the multistate median (MSM) filter [6], the fuzzy-rule-based median (FM) filter, and the iterative median (IM) filter [7]. Although these filters achieve better detection and restoration results, they still tend to blur some fine details. In order not to damage good pixels, the decision-based median filters with switching schemes have been proposed in some papers [3, 13].

In addition, the directional weighted median (DWM) filter [8] performs well at noise ratio higher than 30% when it is compared with the TSM, the ACWM, and the MSM filters. Nevertheless, it needs to calculate some fixed parameters to perform image enhancement. Moreover, the high performance detection (HPD) filter employs the sufficient similar neighbor criteria for image restoration [9]. However, it also needs some fixed parameters, such as the intensity value and the threshold, to work correctly. Thus, the HPD filter cannot completely separate the noise-free pixels from the noise-corrupted pixels, either.

In the recent years, most proposed algorithms for selective impulse noise detection need the thresholds to classify the input pixel as either noise-corrupted or noise-free. These thresholds will severely affect the performance for noise detection. They are finally determined by a series of the iterative experiments.

The decision tree (DT) is popularly used in the field of data mining. Especially, it is a very efficient method in classification problems. In image enhancement, the purpose for noise detection is detecting the pixels whether they are corrupted or not. Thus, noise detection will be strongly regarded as a classification problem. In the past years, some well-known methods need the thresholds for classification which are obtained by manual or forced-searching approaches. They are time-consuming, inefficient, and high time complexity. Therefore, it is essential to employ a systematic approach for solving this problem. In this paper, the particle swarm optimization (PSO) is utilized to optimize the thresholds to find out the approximate optimized DT and the suboptimal solutions for a set of these parameters. Hence, employing the DT is more accurate for noise detection.

The PSO derives next generation using the error values in each generation. It is easier to find the nearer optimal solution using the PSO. The main reason is that the PSO calculates the vector of movement and derives

the position for next generation by considering the optimal solution from the first generation to the current generation. Therefore, the PSO has high ability with memory [17].

In this paper, the novel selective image filters called the median-type filters with an impulse noise detector (IND) using the DT and the PSO for image restoration (MDP), is proposed. It integrates the features of 10 impulse noise detection algorithms based on the DT to construct an adaptive hybrid IND. Also, it employs the PSO to well determine the optimized thresholds for the hybrid noise detector.

The remainder of this paper is arranged as follows. Basic concept is stated in Section 2. Principles for the DT and the PSO are then in detail described in Section 3. In Section 4, the design for the MDP filter will be next depicted. The training structure diagram for the proposed impulse noise detector in the MDP filter is shown in this section. Subsequently, the experimental results demonstrate the comparison for the PSNR values and the restored images at various noise ratios in some well-known corrupted images for different methods in Section 5. Finally, conclusion is given in Section 6.

2. Basic Concept

A gray-level image is represented as a two-dimensional $L \times K$ matrix $X = \{x_{ij} | 1 \leq i \leq L, 1 \leq j \leq K\}$, where L and K are its height and width, respectively, and $x_{ij} \in \{0, 1, 2, \dots, 255\}$ is the pixel gray-level value at position (i, j) in X .

A filter window with size $S = (2\tau + 1)^2 = 2n + 1$ will slide over the image X at position (i, j) to formulate a sample matrix X_{ij} , where $1 \leq i \leq L$, $1 \leq j \leq K$, and S is generally an odd number. Let the value for the central pixel in the filter window X_{ij} be x_{ij} . The filter window X_{ij} usually slides over the image X from left to right and top to bottom. For better clarity, the sample matrix X_{ij} can be rewritten as $X_{ij} = (x_{i-\tau, j-\tau}, x_{i-\tau, j-\tau+1}, \dots, x_{ij}, \dots, x_{i+\tau, j+\tau-1}, x_{i+\tau, j+\tau})$.

In order to change the $(2\tau + 1) \times (2\tau + 1)$ filter window into a one-dimensional vector, it is reorganized by $\mathbf{x}(k) = (x_{-n}(k), \dots, x_{-1}(k), x_0(k), x_1(k), \dots, x_n(k))$, where $x_0(k)$ (or $x(k)$) is the original central pixel value at location k . For instance, a 3×3 filter window centered at $x_0(k)$ is considered, such that $\mathbf{x}(k) = (x_{-4}(k), \dots, x_{-1}(k), x_0(k), x_1(k), \dots, x_4(k))$, where $k = (i - 1) \times K + j$ indicates the pixel located at position (i, j) in the image X , and $x_0(k)$ (or $x(k)$) stands for the central pixel in the filter window.

The general output value in the MF filter with the filter window sized $2n + 1$ is

$$y(k) = \text{median}(\mathbf{x}(k))$$

described as $= \text{median}(x_{-n}(k), \dots, x_{-1}(k), x_0(k), x_1(k), \dots, x_n(k))$,

where *median* represents the median operation.

3. Principles for the DT and the PSO

3.1. Decision Tree

The DT is a kind of tree structure which is applied in classification problems [16]. It can automatically classify the data according to the splitting condition. The DT can deal with variables with continuous type or category type. Its model can sufficiently show each variable's relative importance and effectively deal with a huge dataset with many variables [16].

The purpose for noise detection is detecting the pixels whether they are corrupted or not when the images are processed. Noise detection is certainly considered as a classification problem. Thus, it is necessary to employ a systematic approach for solving this problem. In this paper, the algorithm for classification and regression trees (CART) is employed to construct a DT for the IND in noise detection. Experimentally, employing the DT is more accurate in noise detection.

3.2. Particle Swarm Optimization

Kennedy proposed the PSO algorithm in 1995 [17]. It is designed according to birds' path of movement for food searching. The algorithm integrates the relationship between birds' individual and group features. Birds will decide the next direction and the distance of movement by referencing the past directions of movement and the current position when they search for foods. A particle in the PSO algorithm is simulated as an individual for a bird's food searching. The feature that a particle is simulated as a biological individual in the hyper-dimensional search space is employed to search for the optimal solution. Each particle will simulate the psychological tendencies from every other individual in the group of society. Apart from the individual direction searching, each individual will also learn from the best individual in the group of society. By the way of mutual learning, the individual will be able to find out the place for more food. In other words, the particles can therefore search for better solutions [17].

Here, the detailed procedure for the PSO is described as follows:

1. Set the initial parameters

The parameters swarm size, weight, range of movement for particles, and the number for training generations, must be initially set. The swarm size indicates the number of the particles in each generation. The weight represents one of the parameters employed in calculating the movement vector. The number for training generations means the number for training.

2. Get the fitness

The method for defining the fitness function can be designed by a function of the error value or the accuracy rate. The returned fitness values, such as the

error value and the accuracy rate, are usually the ultimate goal in a problem. Therefore, the fitness can be utilized to decide the merit or demerit of the current location for the particles, and can be employed to determine whether to continue training.

3. The conditions to stop training

Here, two conditions for stop of training will be listed. First, the maximum number of generations for training is reached. Second, the fitness value has satisfied the problem's requirements.

4. Get Gbest and Pbest

The current fitness value for each particle is compared with that of each individual best position. If the particle's fitness value is less than that of its Pbest, the individual best position will be replaced with the particle current position. Similarly, its fitness value is compared with that of its Gbest. If the particle's fitness value is less than that of its Gbest, the position of Gbest will be substituted with that of the particle. Here, getting Gbest and Pbest is described in Eq. (1).

$$\begin{aligned} \text{if fitness}(\mathbf{X}_i) < \text{fitness}(\text{Pbest}_i) \text{ then Pbest} &= \mathbf{X}_i(t), \\ \text{if fitness}(\mathbf{X}_i) < \text{fitness}(\text{Gbest}) \text{ then Gbest} &= \mathbf{X}_i(t), \end{aligned} \quad (1)$$

where t represents the t th generation, $i = 1, \dots, n$, n indicates the number of the particles, $\mathbf{X}_i = (X_{i1}, X_{i2}, \dots, X_{im}) \in \mathfrak{R}^m$ stands for the particle current position, m denotes the particle dimensionality, $\text{Pbest}_i \in \mathfrak{R}^m$ represents the best position for the i th individual particle's path of movement, $\text{Gbest} \in \mathfrak{R}^m$ indicates the position closest to the optimal solution in the group, and fitness means the fitness function.

5. Calculate the movement vector

The movement vector is defined as follows:

$$\mathbf{V}_i(t+1) = w\mathbf{V}_i(t) + c_1 \times r_1 \times (\text{Pbest}_i - \mathbf{X}_i(t)) + c_2 \times r_2 \times (\text{Gbest} - \mathbf{X}_i(t)), \quad (2)$$

where $\mathbf{V}_i = (V_{i1}, V_{i2}, \dots, V_{im}) \in \mathfrak{R}^m$ is the movement vector for the i th particle, w indicates the inertia weight, c_1 and c_2 are the acceleration coefficients which are random numbers in the interval $[0, 1]$, r_1 and r_2 are also random numbers in the interval $[0, 1]$, the first part $w\mathbf{V}_i(t)$ represents the particle's inertia, the second part $c_1 \times r_1 \times (\text{Pbest}_i - \mathbf{X}_i(t))$ stands for the particle's cognition-only model, and the third part $c_2 \times r_2 \times (\text{Gbest} - \mathbf{X}_i(t))$ denotes the particle's social-only model.

6. Modify the particle's position

The method for modifying the particle's position is defined as follows:

$$\mathbf{X}_i(t+1) = \mathbf{X}_i(t) + \mathbf{V}_i(t+1). \quad (3)$$

The particle i 's $(t+1)$ th generation position is the vector addition of the t th generation position and the $(t+1)$ th generation's movement vector. That is, adding the particle's current position and its movement vector becomes the particle's new position.

4. Design for the MDP Filter

4.1. The Structure for the MDP Filter

The IND employs the DT and the PSO algorithm to integrate 10 noise detection algorithms to find out the positions for the corrupted pixels in the images. Fig. 1 demonstrates the structure for the MDP image filter. In this paper, the MDP filter replaces the pixel with the modified median filters' results employing the MF, the ROM, and the PNOM filters, when the pixel is considered as noise-corrupted. Otherwise, the pixel will be kept unchanged.

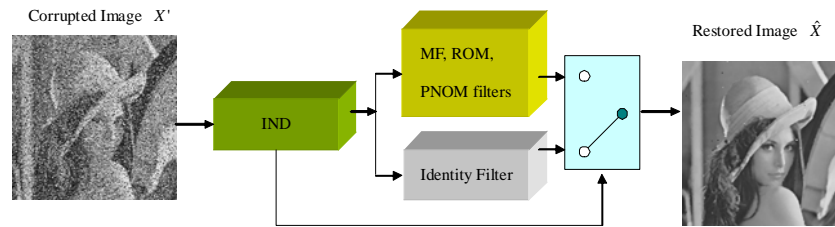


Fig. 1. The structure for the MDP image filter

4.2. The Employed Noise Detection Algorithms

Generally, the values for most pixels which are corrupted by impulse noise are more eminent than those for fine pixels. Therefore, the difference between the input pixel $x(k)$ and the median value $median(x(k))$ in the filter window shows a sufficient reason to determine a corrupted pixel [10]. Here, 10 feature variables are employed as 10 impulse noise detectors to respectively estimate the feature value for each pixel. For clarity, $x(k)$ and $x_0(k)$ will be used interchangeably throughout this paper. Also, the filter window X_{ij} and one-dimensional vector $x(k)$ will be employed alternatively.

Detector 1: the difference between the central pixel value and the median pixel value

The variable $u(k)$ specifies the absolute difference between the input pixel $x(k)$ and the median value $median(x(k))$ as follows:

$$u(k) = |x(k) - median(x(k))|, \quad (4)$$

where median indicates the median operation. The variable $u(k)$ proposes a simple method for detecting impulse noise. A large $u(k)$ value which is greater than the threshold T_u represents that the input $x(k)$ is dissimilar to the median value $median(x(k))$ in the filter window $\mathbf{x}(k)$; that is, it strongly suggests the central pixel $x(k)$ is corrupted by impulse noise.

Detector 2: the difference between the individual pixel value and the average of the sample pixels' values

The variable $p_j(k)$ of the local contrast at location k in the filter window $\mathbf{x}(k)$ is defined as follows:

$$p(k) = \frac{|x(k) - \bar{x}(k)|}{\sum_{i=-n}^n |x_i(k) - \bar{x}(k)|}, \quad (5)$$

where $\bar{x}(k)$ represents the average gray-level value in the filter window $\mathbf{x}(k)$ with size $2n+1$. A large $p(k)$ value which is greater than the threshold T_p indicates that it heavily suggests the central pixel $x(k)$ is corrupted by impulse noise.

Nevertheless, if only variables $u(k)$ and $p(k)$ are employed to determine whether the input $x(k)$ is corrupted or not, then it is difficult to completely detect impulse noise [17]. For instance, a line component usually exists in the image and its width is only one pixel; hence, if the input pixel $x(k)$ is located on the line, it may be identified as an impulse noise and be removed [10]. In order to prevent bad judgments, it is essential to utilize other observations to enhance the detecting correctness, thus $v(k)$ and $q(k)$ are proposed as follows:

Detector 3: the average of the differences between the central pixel value and two closest pixels' values

If only variable $u(k)$ is employed to determine whether the input pixel $x(k)$ is corrupted or not, then it will mistakenly detect $x(k)$ as noise-corrupted in Fig. 2(a). In order to remove the weakness, the variable $v(k)$ is further employed for better noise detection in Eq. (6).

$$v(k) = \frac{|x(k) - x_{\beta_1}(k)| + |x(k) - x_{\beta_2}(k)|}{2}, \quad (6)$$

where $|x(k) - x_{\beta_1}(k)| \leq |x(k) - x_{\beta_2}(k)| \leq |x(k) - x_i(k)|$, $-n \leq i \leq n, i \neq \beta_1, \beta_2$. The pixel values for $x_{\beta_1}(k)$ and $x_{\beta_2}(k)$ are closest to that of $x(k)$ in the filter window $\mathbf{x}(k)$. A large $v(k)$ value which is greater than the threshold T_v indicates that it strongly suggests the central pixel $x(k)$ is corrupted by impulse noise. If we employ $v(k)$ as a detector, then the pixels on the line component in the filter

window $\mathbf{x}(k)$ will not be detected as impulse noises because of the small $v(k)$ value [10, 15, 17].

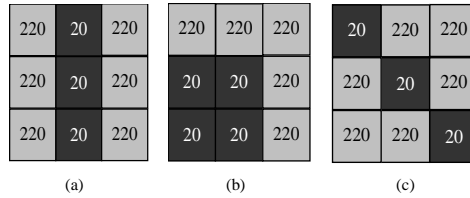


Fig. 2. The filter window $\mathbf{x}(k)$ with size 3×3 (a) the line component (b) the edge component and (c) the thin line component in the filter window, respectively

Detector 4: the difference between the central pixel value and the median value of the filter window $\mathbf{x}(k)$ with the repeating central pixels

In Fig. 2(b), the input pixel $x(k)$ is mistakenly considered as noise-corrupted by $u(k)$ when it is on the edge component in the filter window $\mathbf{x}(k)$. To get good judgment, the variable $q(k)$ is utilized for enhancement. It is described as follows:

$$q(k) = |x_0(k) - c^s(k)|, \tag{7}$$

$$\text{where } c^s(k) = \text{median} \left(x_{-n}(k), \dots, x_{-1}(k), \underbrace{x_0(k), x_0(k), x_0(k), \dots, x_0(k)}_{s \text{ times}}, x_1(k), \dots, x_n(k) \right), \tag{8}$$

and s indicates a positive integer for the repeating times [5, 11].

A large $q(k)$ value which is greater than the threshold T_q indicates that it heavily suggests the central pixel $x(k)$ is corrupted by impulse noise. In [10], Lin presented that any pixel on the edge and the line components in the filter window $\mathbf{x}(k)$ will not be detected as noise-corrupted if the variables $q(k)$ and $v(k)$ are employed. Thus, the system can correctly determine that no impulse noise is located at the pixel $x(k)$ because of the small $q(k)$ and $v(k)$ values. A small $q(k)$ value represents that it heavily suggests the central pixel $x(k)$ is not corrupted by impulse noise, in Fig. 2(b).

Detector 5: the difference between the central pixel value and the average of the sample pixels' values

The variable $g(k)$ is defined as follows:

$$g(k) = |x(k) - \text{avg}(\mathbf{x}(k))|, \tag{9}$$

where $\text{avg}(\mathbf{x}(k)) = \frac{1}{2n} \sum_{i=-n, i \neq 0}^n x_i(k)$, it denotes the average of all pixels' values in the filter window $\mathbf{x}(k)$ without $x_0(k)$. A large $g(k)$ value which is greater than the threshold T_g indicates that a big difference between the input pixel

$x(k)$ value and the average gray-level value in the filter window $\mathbf{x}(k)$, and the central pixel $x(k)$ is possibly corrupted by impulse noise [10].

Detector 6: the average of differences between the central pixel value and four closest pixels' values

The variable $d(k)$ is defined as follows:

$$d(k) = \frac{\sum_{i=1}^4 |x_0(k) - x_{\beta_i}(k)|}{4}, \quad (10)$$

where $|x_0(k) - x_{\beta_j}(k)| \leq |x_0(k) - x_{\beta_{j+1}}(k)|$, $j = 1, 2, 3$, $x_{\beta_1}, x_{\beta_2}, x_{\beta_3}$, and x_{β_4} are the first four pixels in the filter window $\mathbf{x}(k)$ which are nearest to $x(k)$. A large $d(k)$ value which is greater than the threshold T_d indicates that it heavily suggests the central pixel $x(k)$ is corrupted by impulse noise because the difference between $x(k)$ value and its four nearest pixels' values is too big, respectively [10]. Here, although $d(k)$ can effectively detect impulse noise in the image, the thin line components in the filter window $\mathbf{x}(k)$ will be mistakenly detected as noise-corrupted in Fig. 2(c).

Detector 7: the noise detection employing the edge detection median (EDM) filter

The variable $u(k)$ will mistakenly detect the input pixel $x(k)$ as noise-corrupted which is on the thin line component in the filter window $\mathbf{x}(k)$. Thus, Zhang proposed the EDM filter with the variable $r(k)$ to detect the corrupted pixels which are on the thin line component [12]. The variable $r(k)$ is defined as follows:

$$r(k) = \min \left\{ |X_{ij} \otimes K_t| \mid t = 1, 2, 3, 4 \right\} \quad (11)$$

where K_t is the t th convolution kernel and \otimes represents the convolution operator. Four different absolute values are obtained that the filter window X_{ij} is operated with 4 different convolution kernels, respectively. The minimum of four values is selected for impulse noise detection [12]. Four different convolution kernels are shown in Fig. 3. In convolution operation, each pixel value in the filter window X_{ij} is multiplied by the corresponding weight. The result for convolution operation is the summary of all operation results. That is, it is the summary for each pixel in the filter window X_{ij} multiplied by the weight of the corresponding position in the t th convolution kernel.

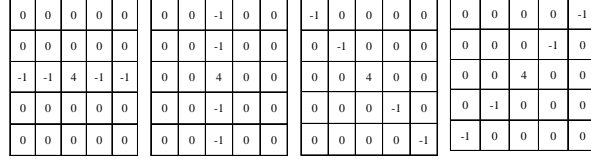


Fig. 3. four 5 × 5 convolution kernels

The variable $r(k)$ utilized to detect impulse noise has 3 following significations:

1. Large $r(k)$ value indicates all of 4 convolution products are very large. Thus, $x(k)$ is considered as an impulse noise.
2. Small $r(k)$ value denotes all of 4 convolution products are very small or close to zero. Hence, $x(k)$ is regarded as a noise-free pixel.
3. Because one of 4 convolution products is very small (or close to zero) and the others are possibly large, the variable $r(k)$ will be small when the central pixel $x(k)$ is on the edge or the thin line component in the filter window X_{ij} .

According to aforementioned $r(k)$, if the input pixel $x(k)$ is considered as noise-corrupted, the minimum of four convolution products is relatively large. The $r(k)$ for the noise-free input pixel $x(k)$ which is on the edge or the thin line component will be relatively small. Here, the threshold T_r will be employed to determine whether the pixel $x(k)$ is corrupted or not.

Detector 8: the noise detection employing the SD-ROM filter

The variable $\mathbf{h}(k)$ is defined as follows [13]:

$\mathbf{h}(k)$ is a one-dimensional vector that is obtained from the filter window $\mathbf{x}(k)$ without $x_0(k)$, shown by

$$\mathbf{h}(k) = (x_{-n}(k), \dots, x_{-1}(k), x_1(k), \dots, x_n(k)). \tag{12}$$

The elements in $\mathbf{h}(k)$ are sorted by ascending order as follows:

$$\mathbf{h}'(k) = (h'_{(1)}(k), h'_{(2)}(k), \dots, h'_{(2n)}(k)), \tag{13}$$

where $h'_{(1)}(k) \leq h'_{(2)}(k) \leq \dots \leq h'_{(2n)}(k)$. Subsequently, the rank ordered mean (ROM) is defined as follows:

$$m(k) = (h'_{(2n/2)}(k) + h'_{((2n/2)+1)}(k)) / 2, \tag{14}$$

where $m(k)$ is the value for the ROM. Also, the rank ordered difference is formulated by

$$\mathbf{z}(k) = (z_1(k), z_2(k), z_3(k), z_4(k)), \tag{15}$$

where

$$z_i(k) = \begin{cases} h'_{(i)}(k) - x(k), & \text{if } x(k) \leq m(k) \\ x(k) - h'_{(2n-i)}(k), & \text{if } x(k) > m(k) \end{cases} \text{ for } i = 1, \dots, 4. \tag{16}$$

$z(k)$ is employed to determine the input pixel $x(k)$ whether it is corrupted or not. For instance, if $z_1(k)$ is positive, it means that the input pixel $x(k)$ is larger in the filter window $\mathbf{x}(k)$. If $z_1(k)$ is greater than the threshold T_{z_1} , the input pixel $x(k)$ is considered as noise-corrupted. Similarly, $z_2(k)$, $z_3(k)$ and $z_4(k)$ can also provide the determination for noise detection. If each inequality in the following formula satisfies, the input pixel $x(k)$ is regarded as noise-corrupted.

$$z_i(k) > T_{z_i}, \quad i = 1, \dots, 4, \quad (17)$$

where $T_{Z_1}, T_{Z_2}, T_{Z_3}, T_{Z_4}$ are the thresholds and $T_{Z_1} < T_{Z_2} < T_{Z_3} < T_{Z_4}$.

Detector 9: the difference between the central pixel value and the neighbored pixels' values after sorted

In [14], Aizenberg proposed a noise detection algorithm about the variable $e(k)$ to detect the corrupted pixels. The variable $e(k)$ is defined as follows: (differential rank impulse detector, DRID)

$$e(k) = \begin{cases} |x(k) - \mathbf{x}'_{(a-1)}(k)|, & \text{if } a > (n+1) \\ |x(k) - \mathbf{x}'_{(a+1)}(k)|, & \text{if } a < (n+1) \\ 0, & \text{otherwise.} \end{cases} \quad (18)$$

The filter window $\mathbf{x}(k)$ is sized with $2n+1$. After the elements in the filter window $\mathbf{x}(k)$ being sorted, a new filter window $\mathbf{x}'(k)$ is obtained. The rank for the input pixel $x(k)$ in the new filter window $\mathbf{x}'(k)$ is a . $\mathbf{x}'_{(a-1)}(k)$ and $\mathbf{x}'_{(a+1)}(k)$ represent the $(a-1)$ th and the $(a+1)$ th pixel's values in the new filter window $\mathbf{x}'(k)$, respectively. Similarly, large $e(k)$ value which is larger than the threshold T_e represents that the input pixel $x(k)$ is more relatively corrupted by impulse noise [12].

Detector 10: the noise detection employing the adaptive center-weighted median (ACWM) filter

The ACWM is a modified center-weighted median (CWM) filter [5]. It employs the difference between the input pixel $x(k)$ and the output of the CWM filter to detect impulse noise.

$$o_i = |x(k) - c^s(k)| = |x(k) - c^{2i+1}(k)|, \quad (19)$$

where $i = 0, 1, \dots, n-1$, and $c^i(k)$ means as Eq. (8).

If any inequality satisfies in the following equation, the input pixel $x(k)$ is determined as an impulse noise [17].

$$o_i(k) > T_{o_i}, \quad i = 0, \dots, 3, \quad (20)$$

where T_{o_i} is the threshold. The corresponding threshold is obtained from the following equation:

$$T_{o_i} = \sigma \cdot \text{MAD} + \delta_i, \quad (21)$$

where σ and δ_i are predefined parameters, and MAD represents the median of the absolute deviations from the median, defined as follows:

$$\text{MAD} = \text{median}(|x_{-n}(k) - c^1(k)|, \dots, |x_0(k) - c^1(k)|, \dots, |x_n(k) - c^1(k)|), \quad (22)$$

where the filter window X_{ij} is sized with $2n+1$ and $c^1(k)$ means as Eq. (8).

Each one of 10 noise detectors has its own decision threshold. Several tests by implementation must have been done before the optimal thresholds are obtained. The thresholds for 10 noise detectors will absolutely affect the accuracy rate in noise detection. Therefore, the optimal thresholds for training the DT with the PSO algorithm will be obtained. Also, the accuracy rate which the DT estimates is considered as the fitness for the PSO.

4.3. The Training Method for the IND in the MDP Filter

Each one of 10 aforementioned detectors has its own threshold in the hybrid noise detector. The threshold in each detector usually affects the accuracy in noise detection for each noise detector. The optimal thresholds are determined only after many repeated experiments. Thus, a systematic method needs to be employed to obtain a set of optimal thresholds. Each noise detection algorithm has its strengths, weaknesses, and reciprocal characteristics. Therefore, the MDP filter adopts the DT to integrate each detector's characteristics and reciprocal relationships, and utilizes the PSO algorithm to determine the required optimal thresholds for the hybrid detection algorithm. Doing so can construct an adaptive hybrid noise detector to improve the detection rate and achieve better performance for image restoration.

The MDP filter adopts the PSO to find out the optimal thresholds for training the DT. Subsequently, it employs the error rate of the DT classification as the fitness of the PSO. Finally, it outputs a set of thresholds T determined by the PSO algorithm and a set of DT impulse noise detectors.

4.4. The Structure for the IND

Fig. 4 shows the structure for the IND. $\mathbf{T} = (T_u, T_p, T_v, T_q, T_g, T_d, T_r, T_{z_1}, T_{z_2}, T_{z_3}, T_{z_4}, T_e, \sigma, \delta_1, \delta_2, \delta_3, \delta_4)$, where $T_u, T_p, \dots, \sigma, \delta_1, \delta_2, \delta_3, \delta_4$ are the thresholds for Detector 1, Detector 2, ..., Detector 10, respectively. They are the optimal thresholds for the noise detection component which are found by the PSO algorithm.

The operation procedure is that the input image X employs the PSO to determine the optimal thresholds for noise judgment and then calculates the

Design of Median-type Filters with an Impulse Noise Detector Using Decision Tree and Particle Swarm Optimization for Image Restoration

judgment results for 10 detectors. Subsequently, the detection results of 10 noise detectors are fed into the trained DT (TDT) to judge whether the input pixel $x(k)$ is corrupted and the MDP filter will output the binary noise flag map $B(i, j)$. $B(i, j)$ clearly indicates that each pixel in the image is corrupted or not.

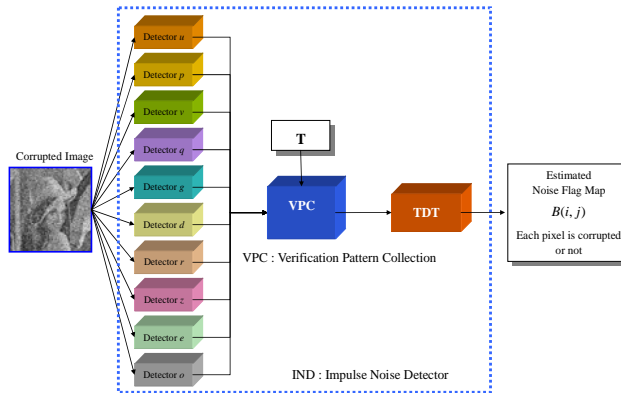


Fig. 4. The structure for the MDP noise detector

4.5. The MF, ROM, and PNOM Filters

In the MDP filter, after the noise detection procedure is finished, the binary flag map $B(i, j)$ indicates the locations where the pixels are corrupted by impulse noise. When the input pixel $x(k)$ is corrupted it will be replaced with the outputs of the MF, the ROM, and the PNOM filters, respectively. The MF filter outputs the median value of all sorted pixels in the filter window $\mathbf{x}(k)$. The ROM filter calculates the average of the fourth and the fifth pixels for all sorted pixels without the central pixel in the original filter window $\mathbf{x}(k)$. The PNOM filter sorts the noise-free pixels in the filter window $\mathbf{x}(k)$ by ascending order. They will be indicated as one dimensional vector as follows:

$$\mathbf{f}(k) = (f_{(1)}(k), f_{(2)}(k), \dots, f_{(C)}(k)) \quad (23)$$

where $f_{(1)}(k) \leq f_{(2)}(k) \leq \dots \leq f_{(C)}(k)$ and C represents the number of the noise-free pixels in the filter window $\mathbf{x}(k)$.

Here, the algorithm for the PNOM filter is described as follows:

Step 1. Input the corrupted image X' and the noise detector, output the binary noise flag map $B(i, j)$. $B(i, j) = 1$ indicates the central pixel $x(k)$ is noise-corrupted.

Step 2. Slide the filter window $\mathbf{x}(k)$ over the corrupted image X' from left to right and top to bottom.

Step 2.1 Count the number C of the noise-free pixels in the filter window $\mathbf{x}(k)$ for each input corrupted pixel $x(k)$.

Bae-Muu Chang, Hung-Hsu Tsai, Xuan-Ping Lin, and Pao-Ta Yu

Step 2.2 If $C > 0$, $x(k) = \text{median}(\text{sort}(x(k)$ excluding the corrupted pixels)),
 $B(i, j) = 0$, $C = 0$,

Else $x(k)$ is kept unchanged.

Step 3 Extend the filter window $x(k)$ with size 5×5 , perform Step 2 through Step 2.2

Step 4 Output the restored image \hat{x} .

5. Experimental Results

Fig. 5 depicts that 10 different images which are gray-level images with size 256×256 are adopted for the experiments in this work. They respectively represent 'Airplane', 'Baboon', 'Barbara', 'Bridge', 'Butterfly', 'Couple', 'Fishingboat', 'Goldhill', 'Lena' and 'Peppers'.



Fig. 5. The images adopted for evaluation in the experiments

5.1. The Training and the Testing Methods Adopted in the Experiments

In order to evaluate the performance of impulse noise detector and image restoration in the MDP image filter, the training patterns are collected from the images at 20% noise ratio. In the training and the testing methods, $P \in I_p$, where P indicates the noise ratio, $I_p \in \{2\%, 4\%, \dots, 18\%, 22\%, \dots, 30\%\}$. The testing patterns are collected from the same images at 2~30% different noise ratios excluding 20%.

5.2. The Evaluation for the MDP Image Filter

To evaluate the performance for the MDP filter, the noise detection performance needs to be examined and its detector will be compared with various noise detectors applied on different images at different noise ratios. Also, the error rate in noise detection for the noise detectors is considered as

the evaluating criterion. In the experiments, the existing well-known noise detection algorithms include: the CSAM [18], the GP [15], the SD-ROM [13], the PSM [4], the EDM [12], the ACWM [5] algorithms, where the PSM, the SD-ROM, and the ACWM algorithms employ the filter window with size 3×3 , the EDM algorithm utilizes 5×5 filter window, both the CSAM and the GP algorithms use the filter windows with size 3×3 and 5×5 . In the quality evaluation of image restoration, two other filters, the MF [1] and the TSM [3] filters, are also employed.

5.3. The Performance Evaluation for the MDP Filter Applied on the Images Corrupted by Salt and Pepper Impulse Noise

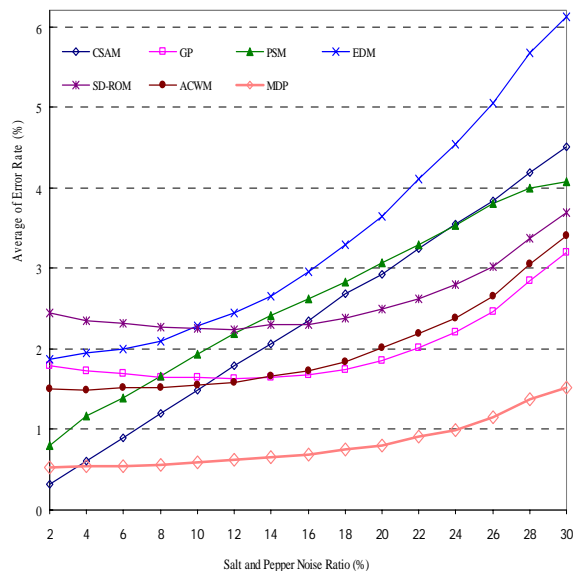


Fig. 6. The comparison for the average error rate in noise detection for different methods in 10 different images corrupted by the salt and pepper impulse noise at various noise ratios

Fig. 6 shows the quantitative comparison for the average error rate in noise detection for different detectors while 10 different images corrupted by salt and pepper impulse noise at various noise ratios. Fig. 7 depicts the quantitative comparison for the average PSNR in image restoration for different methods while 10 different images corrupted by salt and pepper impulse noise at various noise ratios. Fig. 8 illustrates the visual comparison

for different methods in image restoration when the image 'Butterfly' is corrupted by salt and pepper impulse noise at 24% noise ratio. Fig. 9 demonstrates the visual partial enlargement for the image 'Butterfly'.

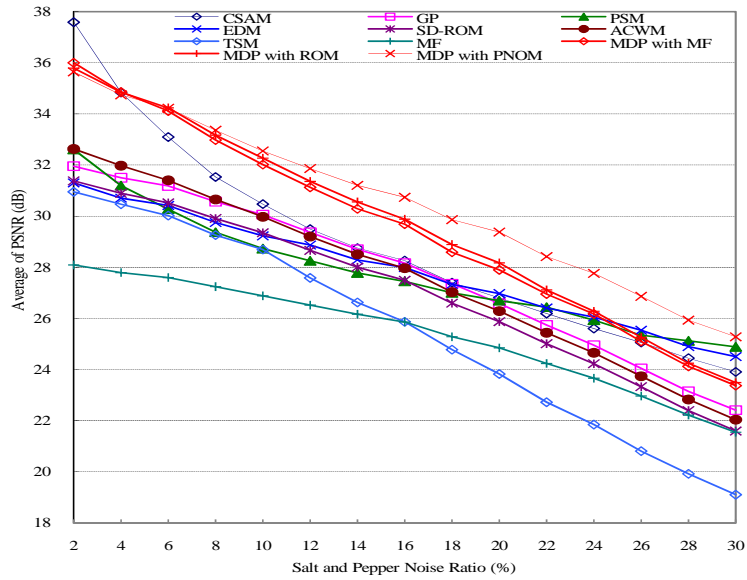


Fig. 7. The comparison for the average PSNR (dB) in image restoration for different methods in 10 different images corrupted by salt and pepper impulse noise at various noise ratios

5.4. The Performance Evaluation for the MDP Filter Applied on the Images Corrupted by Fixed-length Impulse Noise

Fig. 10 shows the quantitative comparison for the average error rate in noise detection for different detectors while 10 different images corrupted by fixed-length impulse noise at various noise ratios. Fig. 11 depicts the quantitative comparison for the average PSNR in image restoration for different methods while 10 different images corrupted by fixed-length impulse noise at various noise ratios. Fig. 12 illustrates the visual comparison for different methods in image restoration when the image 'Lena' is corrupted by fixed-length impulse noise at 18% noise ratio. Fig. 13 demonstrates the visual partial enlargement for the image 'Lena'.

Design of Median-type Filters with an Impulse Noise Detector Using Decision Tree and Particle Swarm Optimization for Image Restoration

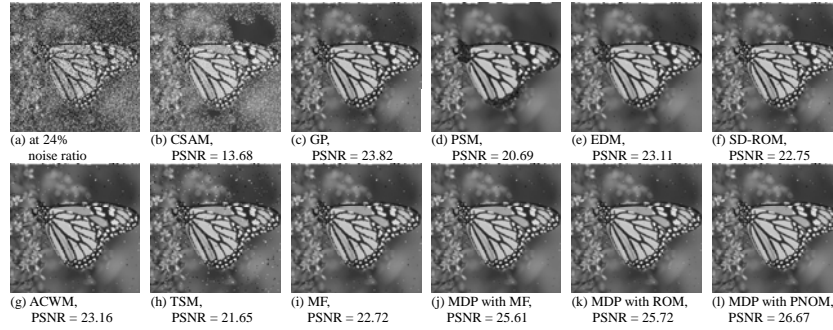


Fig. 8. The comparison for different methods in image restoration for the image 'Butterfly' is corrupted by salt and pepper impulse noise at 24% noise ratio

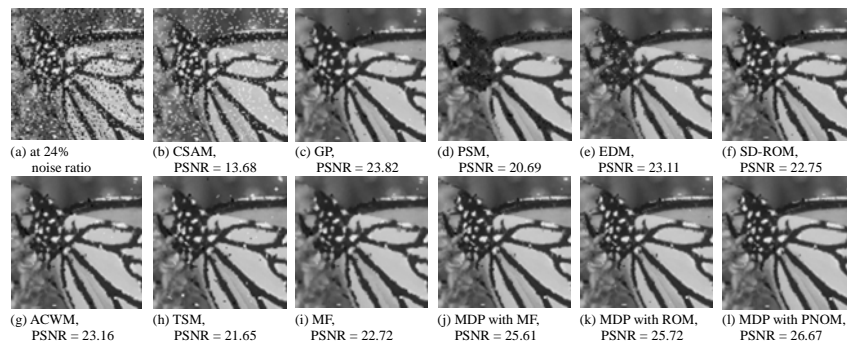


Fig. 9. The comparison for the partial enlargement using different methods in image restoration for the image 'Butterfly' corrupted by salt and pepper impulse noise at 24% noise ratio

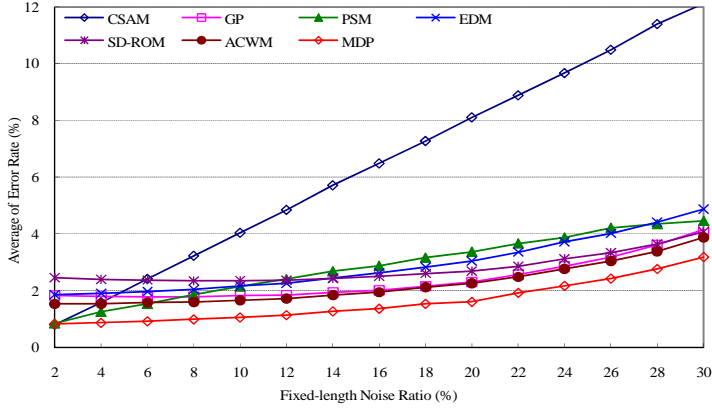


Fig. 10. The comparison for the average error rate in noise detection for different methods in 10 different images corrupted by fixed-length impulse noise at various noise ratios

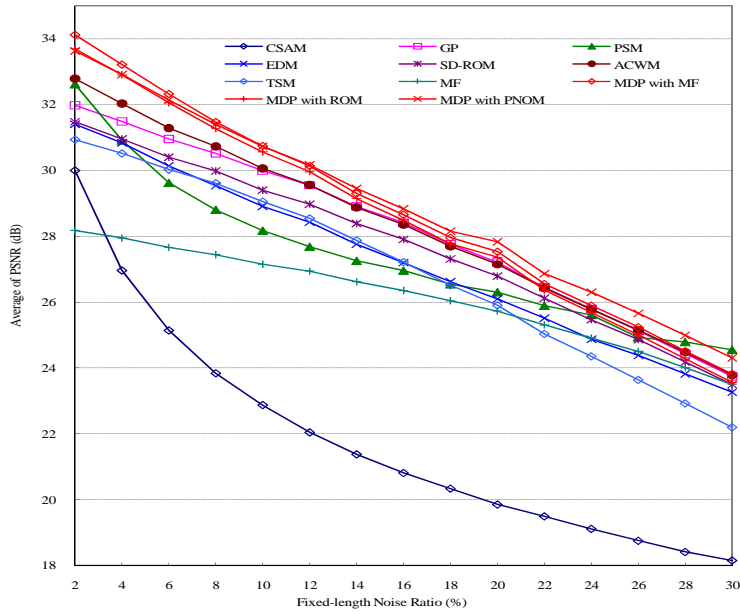


Fig. 11. The comparison for the average PSNR (dB) in image restoration for different methods in 10 different images corrupted by fixed-length impulse noise at various noise ratios

Design of Median-type Filters with an Impulse Noise Detector Using Decision Tree and Particle Swarm Optimization for Image Restoration

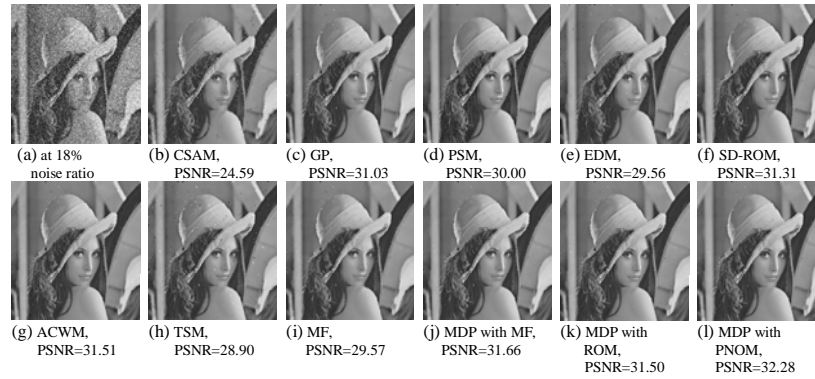


Fig. 12. The comparison for different methods in image restoration in the image 'Lena' is corrupted by fixed-length impulse noise at 18% noise ratio

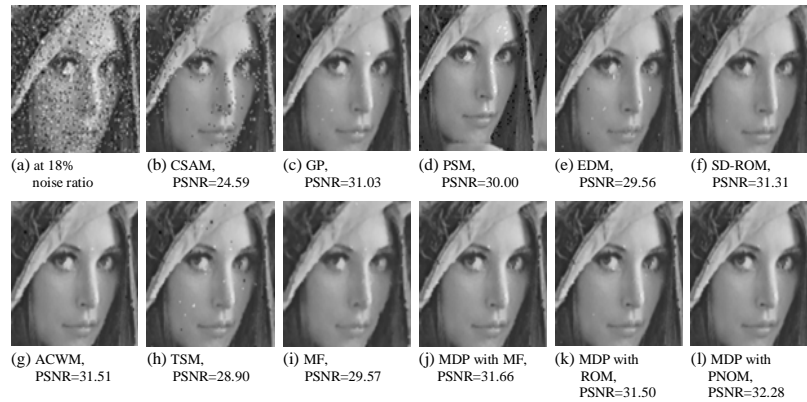


Fig. 13. The comparison for the partial enlargement for different methods in image restoration for the image 'Lena' corrupted by fixed-length impulse noise at 18% noise ratio

The important observation is proposed in Figs. 6 and 7. Although the CSAM filter can effectively detect the locations where the input pixels are corrupted by salt and pepper impulse noise and restore the corrupted images, it can not achieve good results for image restoration when the images are corrupted by fixed-length impulse noise. The reason is that the pixels will be nearly considered as noise-corrupted by the CSAM filter when their gray-level values are 0 or 255. Therefore, although the CSAM filter can achieve good performance in image restoration for filtering salt and pepper impulse noise, it can not obtain good results for fixed-length impulse noise.

5.5. The Performance Evaluation for the MDP Filter When the Images Are Corrupted by Random-valued Impulse Noise

Fig. 14 shows the quantitative comparison for the average error rate in noise detection for different detectors while 10 different images corrupted by random-valued impulse noise at various noise ratios. Fig. 15 depicts the quantitative comparison for the average PSNR in image restoration for different methods while 10 different images corrupted by random-valued impulse noise at various noise ratios. Fig. 16 illustrates the visual comparison for different methods in image restoration when the image 'Fishingboat' is corrupted by random-valued impulse noise at 10% noise ratio. Fig. 17 demonstrates the visual partial enlargement for the image 'Fishingboat'.

An important observation is presented again in Figs. 6 and 7. Although the CSAM filter can effectively detect the corrupted pixels caused by salt and pepper impulse noise and restore the corrupted images, it cannot also achieve good results in image restoration for fixed-length or random-valued impulse noise.

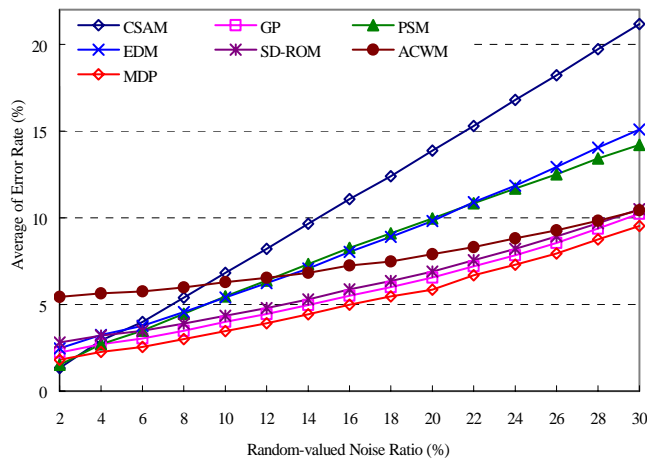


Fig. 14. The comparison for the average error rate in noise detection for different methods in 10 different images corrupted by random-valued impulse noise at various noise ratios

Design of Median-type Filters with an Impulse Noise Detector Using Decision Tree and Particle Swarm Optimization for Image Restoration

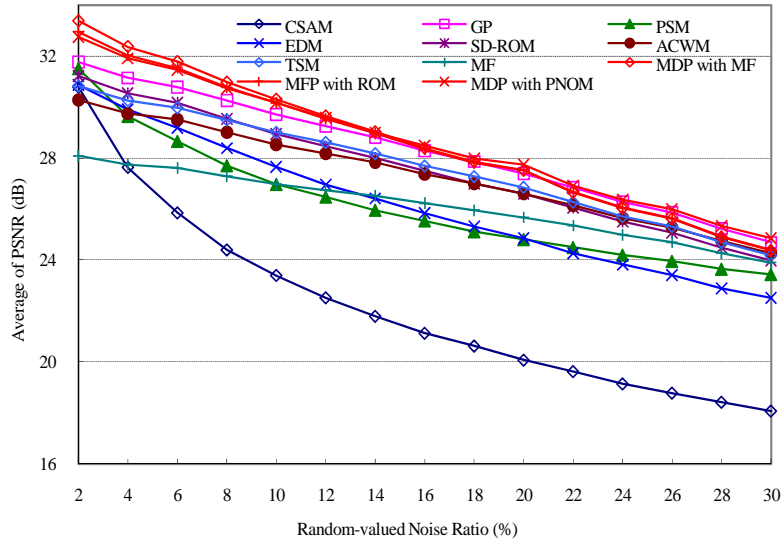


Fig. 15. The comparison for the average PSNR (dB) in image restoration for different methods in 10 different images corrupted by random-valued impulse noise at various noise ratios

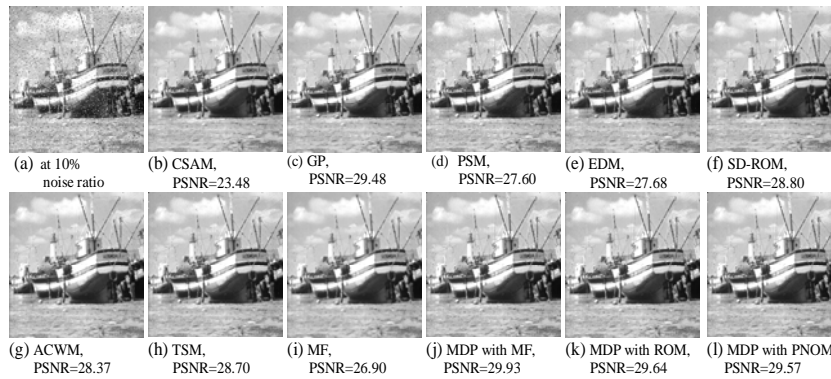


Fig. 16. The comparison for different methods in image restoration in the image 'Fishingboat' corrupted by random-valued impulse noise at 10% noise ratio

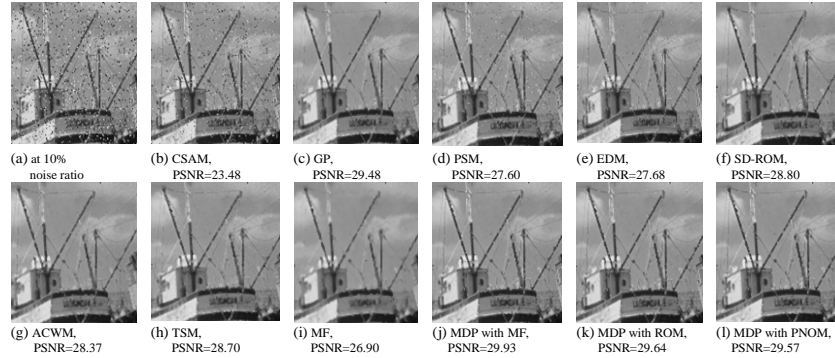


Fig. 17. The comparison for the partial enlargement for different methods in image restoration for the image 'Fishingboat' corrupted by random-valued impulse noise at 10% noise ratio

6. Conclusion

This paper has proposed a novel image filter, called the MDP filter, which is used for the recovery of the corrupted pixels by impulsive noises in gray-level images. The IND is an adaptive hybrid noise detector which is constructed by integrating 10 impulse noise detectors based on the DT and the PSO. In the MDP filter, the high performance IND is employed to powerfully detect impulse noises and the modified MFs effectively restore for the corrupted images. Subsequently, the restoring process in the MDP filter respectively utilizes the MF, the ROM, and the PNOM filters to restore the corrupted pixels. The computational complexity of the proposed filter is high. It shall be improved with a modified algorithm in the future work. Experimental results demonstrate that the MDP filter can effectively restore the gray-level images corrupted by impulse noises and outperform the existing well-known methods.

Acknowledgment. This work is partially supported by the National Science Council of Taiwan, R.O.C., for its financially supporting the research under Contract No. NSC 97-2221-E-150-070. The authors also gratefully acknowledge the helpful comments of the reviewers for promoting the quality of the paper.

References

1. Astola, J., Kuosmanen, P.: *Fundamentals of Nonlinear Digital Filtering*: CRC Press. Boca Raton, FL. (1997)
2. Yin, L., Yang, R., Gabbouj, M., Neuvo, Y.: Weighted Median Filters: a Tutorial. *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, Vol. 43, No. 3, 157-192, Mar. (1996)
3. Chen, T., Ma, K. K., Chen, L. H.: Tri-state Median Filter for Image Denoising. *IEEE Trans. Image Process.*, Vol. 8, 1834-1838, Dec. (1999)
4. Zhou, W., David, Z.: Progressive Switching Median Filter for the Removal of Impulse Noise from Highly Corrupted Images. *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, Vol. 46, No. 1, 78-80, Jan. (1999)
5. Chen, T., Wu, H. R.: Adaptive Impulse Detection Using Center-weighted Median Filters. *IEEE Signal Processing Letters*, Vol. 8, 1-3. (2001)
6. Chen, T., Wu, H. R.: Impulse Noise Removal by Multi-state Median Filtering. In *Proc. ICASSP'2000, Istanbul, Turkey*, 2183-2186. (2000)
7. Forouzan, A. R., Araabi, B. N.: Iterative Median Filtering for Restoration of Images with Impulsive Noise. *IEEE Proceedings of International Conference, Electronics, Circuits and Systems*, Vol. 1, 232-235, Dec. (2003)
8. Dong, Y., Xu, S.: A New Directional Weighted Median Filter for Removal of Random-valued Impulse Noise. *IEEE Signal Processing Letters*, Vol. 14, 193-196, Mar. (2007)
9. Awad, A. S., Man, H.: High Performance Detection Filter for Impulse Noise Removal in Images. *Electronics Letters*, Vol. 44, 192-194, Jan. (2008)
10. Lin, T. C., Yu, P. T.: Thresholding Noise-free Ordered Mean Filter Based on Dempster-Shafer Theory for Image Restoration. *IEEE Trans. on Circuits and Systems*, Vol. 53, No. 5, 1057-1064, May. (2006)
11. Ko, S. J., Lee, Y. H.: Center Weighted Median Filters and Their Applications to Image Enhancement. *IEEE Trans. Circuits System*, Vol. 38, No. 9, 984-993, Sep. (1991)
12. Zhang, S., Karim, M. A.: A New Impulse Detector for Switching Median Filters. *IEEE Signal Processing Lett.*, Vol. 9, 360-363, Nov. (2002)
13. Abreu, E., Lightstone, M., Mitra, S. K., Arakawa, K.: A New Efficient Approach for the Removal of Impulse Noise from Highly Corrupted Images. *IEEE Trans. Image Processing*, Vol. 5, 1012-1025, June. (1996)
14. Aizenberg, I., Butakoff, C.: Effective Impulse Detector Based on Rank Order Criteria. *IEEE Signal Processing Lett.*, Vol. 11, 363-366, Mar. (2004)
15. Petrovic, N. I., Crnojevic, V.: Universal Impulse Noise Filter Based on Genetic Programming. *IEEE Trans. on Image Processing*, Vol. 17, Issue 7, 1109-1120, Jul. (2008)
16. Tan, P. N., Steinbach, M., Kumar, V.: *Introduction to Data Mining*: Addison-Wesley. (2006)
17. You, P.S., Hsieh, Y. C., Huang, C. M.: A Particle Swarm Optimization Based Algorithm to the Internet Subscription Problem. *Expert Systems with Applications*, Vol. 36, Issue 3, 7093-7098, Apr. (2009)
18. Pok, G., Liu, J. C., Nair, A. S.: Selective Removal of Impulse Noise Based on Homogeneity Level Information. *IEEE Trans. on Image Processing*, Vol. 12, No. 1, 85-92, Jan. (1999)

Bae-Muu Chang, Hung-Hsu Tsai, Xuan-Ping Lin, and Pao-Ta Yu

Bae-Muu Chang received the BS degree in computer science from Tamkang University, Taipei, Taiwan, in 1977, the MS degree in computer science from New York Institute of Technology, New York, USA, in 1989, and the PhD degree in computer science and information engineering from National Chung Cheng University, Chiayi, Taiwan, in 2009. Since 1990, he has been with the Department of Information Management at Chienkuo Technology University, Changhua, Taiwan, where he is currently an associate professor. His research interests include neural networks, fuzzy systems, character recognition, intelligent filter design, intelligent networks, and e-Learning.

Hung-Hsu Tsai received the BS and the MS degrees in applied mathematics from National Chung Hsing University, Taichung, Taiwan, in 1986 and 1988, respectively, and the PhD degree in computer science and information engineering from National Chung Cheng University, Chiayi, Taiwan, in 1999. Currently, he is a professor at Department of Information Management, National Formosa University, Huwei, Yulin, Taiwan. He has worked in industry for the SYSTEX Corporation, and in academia for Nanhua University, Chiayi, Taiwan. He is an honorary member of the Phi Tau Phi Scholastic Honor Society. He has been selected and included in the 9th edition of Who's Who in Science and Engineering which has been published in 2006. He serves as a technical reviewer for various scientific journals and numerous international conferences. His research interests include computational intelligence, machine learning, support vector machines, multimedia security, digital watermarking, intelligent filter design, content-based multimedia retrieval, data mining, e-Learning and system integration with web services.

Pao-Ta Yu received the BS degree in mathematics from National Taiwan Normal University, Taipei, Taiwan, in 1979, the MS degree in computer science from National Taiwan University, Taipei, Taiwan, in 1985, and the PhD degree in electrical engineering from Purdue University, West Lafayette, Indiana, in 1989. Since 1990, he has been with the Department of Computer Science and Information Engineering at National Chung Cheng University, Chiayi, Taiwan, where he is currently a professor. His research interests include neural networks and fuzzy systems, nonlinear filter design, intelligent networks, XML technology and e-Learning.

Xuan-Ping Lin respectively received the BS degree and the MS degree in computer science & information engineering and information management from National Formosa University, Huwei, Yulin, Taiwan, in 2006 and 2009. He is currently an engineer at AU Optronics Corporation. His research interests include particle swarm optimization, support vector machine, data mining, and intelligent filter design.

Received: April 05, 2009; Accepted: May 21, 2010.

Personalising the Dynamic Information Resources in a Self Organized Web System Using CAS and MAS

Rattrout Amjad^{1,2}, Assaf Rasha², and Al-Dahoud Ali³

¹University Claude Bernard, LIRIS, Lyon, France

²Al-Quds University, Jerusalem, Palestine

amjad.rattrout@liris.cnrs.fr

rassaf@science.alquds.edu

³Al Zaytoonah University, Amman, Jordan

aldahoud@alzaytoonah.edu.jo

Abstract. The Web is a constantly growing dynamic environment where the components are changed in non-linear ways. These components represent the targets to researches in order to better understand the behavior of the Web, where the owners and the users in this environment exist as out factors. Web page Usage information is the term which describes ways and methods of using the Web. Various factors affect the use of the diversity of the resources in the Web, The non-linear way of its growth, and the evolution in the methods for how we build the Web pages which eventually leads to reflecting the users' interests. Personalizing the results of search engines are created to meet the users need for information on the Web. Generally, the researchers seek user's satisfaction through utilizing these search engines to serve the user. One of the most efficient methods in this domain is the use of semantic measure algorithms to personalize and recognize the outputs of the information resources according to the users' needs. The Web is represented as three aspects: Content, Structure, and Usage. Three components can lead to a personalized Web in order to reinforce the semantic value. This paper will present a model that uses new Web Usage information to see the effects on the semantic values, and how it will help us achieve a robust well personalized and organized Web. It will consider the Usage space as the field of our research as we will simulate this environment in the MAS "Multi Agents System" and CAS "Complex Adaptive System" paradigm.

Keywords: Complex Adaptive Systems, Complex Web, Web Content, Web Usage, Web Structure, and Multi-Agents System.

1. Introduction

Growth of information is the key aspect of World Wide Web. World Wide Web has become the main source of information. Users think that they can find the

optimal solution because retrieval time is less and search gives more related topics which they are looking for. Satisfaction of the users requires a filtered, organized, and maintained data, but because the web environment is a Complex System, it is presented as a direct graph where its nodes are websites and its vertices are links connected nodes with each other. That is why it is hard to predict the growth of information behavior on it. It is hard for a search engine to completely understand the user's desire from the given keywords. The Semantic Web is the solution to this problem which is based on a vision of Tim Berners Lee, the inventor of the WWW; he defined the semantic web as an extension of the current web in which information is given as a well-defined meaning, better enabling computers and people to work in cooperation". Agent software can utilize computation of semantic value to provide personalized user services. The main goal is to offer an efficient utilization of information. Achieving such a goal requires adopting the perspective of the CAS "Complex Adaptive System" model. This leads to finding the WebPages of relevant topics during the search process. Information on the Web is characterized in terms of three main aspects characterized as Content, Structure, and Usage. Inferring from Holland's CAS [1] properties and mechanisms, and using similarity rules adopted by Menczer model [17], to illustrate a combination of these components.

In section 2, reviewing of CAS characteristic behaviors, in sections 4, we illustrate the WACO model for organizing the content on the web dynamically, and in sections 5, 6 and 7 we propose a model that combines Content, Usage and Structure. We use the combination between the three respects (Content, Structure, and Usage), specially the Usage information to reinforce the semantic value to achieve a self organization Web. Finally, we illustrate the results obtained.

2. The Web from a CAS Perspective

It's believed by most researchers in this field [1, 2] that CAS consists of many interacting parts which give rise to emergent patterns of behavior. The behavior is believed to emerge due to the fact that at the macroscopic level, the system demonstrates new complex properties which are not found at the local level of the different components. CAS is in no need for central control or rules governing its behaviors since it is adapted and adjusted to changes in the environment. CAS systems are non –linear systems, i.e. "the whole is more than the sum of its components" [1] Global pattern arises from the local interaction of individual agents. Such pattern cannot be predicted at the local level. Such systems allow the emergence of order through a process of self-organization [2]. The study of CAS has been applied to various fields and areas of knowledge such as economy [3], organization [4], ecology (5), biology, the immune system as well as the brain.

2.1. CAS Interactions

CAS has no centralized control which governs the system overall behavior. At the local level agents govern their own rules of environment and the emergence of order takes place at the macroscopic level. There is No global control or authority control web page creation since web authors all over the world are allowed to do all the changes they need about web pages and websites as well as being able to create hyperlinks to any page or node in the web graph. However, the web organizes itself into web communities according to hyperlinks Structure analysis. Flake defines a web community in [6] as “a collection of web pages that each member page has more hyperlinks in either direction” within the community than the outside of the interests arise with no central control what so ever.

2.2. Using Holland’s Properties and Mechanisms

A bottom up approach is required for modeling such a system as CAS. They are composed of agents interacting with each other, adapting and co-evolving in their environment. Such an approach should be able to identify the various agents and their rules of behavior and interactions. The inside of the system gives rise to emergent properties. In this Paper, we used properties and mechanism proposed by John Holland which identifies a CAS as [1]:-

Aggregation: It is the property through which agent group gives rise to categories or Meta-agents which recombine to a higher level (Meta –agents), thus creating the Complex System. Meta–agents emerge because of agent interactions at the lower level. We group Content and Structure by users need into a web page which is grouped into websites which are grouped into web communities (Meta –agents). These Meta-agents arise and self-organize without any centralized control. Self-organization is a consequence of a retroactive interaction between Usage, Content and Structure. Web page designers change the Content and Structure of their web pages due to the fact that the user needs are developing rapidly. Besides, web communities are emerging continuously. Further more, the emergence of hubs and authorities in the web make aggregate behavior observable [7]

Tagging: A Tag might be considered as the major topic of a web community or the word vector “bags of words” of a certain web page that is used in text analysis as well as web page similarity Analysis.

Non-linearity: is the property where the emergent behaviour of the system is the result of a non-proportionate response to its stimulus. That means the behaviour resulting from the interactions between aggregate agents is more complicated than a simple summation or average of the simple agents. Thus the system can not be predicted by simply understanding how each component works and behaves. The growth of the web is a nonlinear process.

Flows: The physical resources or the information circulating through the nodes of a complex network.

Diversity: The diversity of skills, experiments, strategies, rules of different agents ensure the dynamic adaptive behaviour of a complex adaptive system. The web has a large number of interacting constituents and this diversity in the web is contributing to its robustness. We observe diversity in its Usage, Structure and Content. In [8] users were classified into random users, rational users and recurrent users. Web page authors come from different backgrounds, creating a vast variety of topics. Web pages are also diverse in their Structure, like hubs and authorities pages [7], and web pages were divided into five categories: Strongly Connected Components SCC, IN, OUT, tendrils and tubes, and disconnected.

Internal models or schemas: They are the functions or rules that the agents use to interact with each other and with their environment. These schemas direct agent's behaviours.

Building blocks: The component parts that can be combined and reused for each instance of a model. Identifying these blocks is the first step in modeling a CAS. Sub-graphs motifs form the building blocks for the WWW network [9], and web services are building blocks for distributed web based applications [10].

3. Related work

Many methods were presented in this domain of research, but the most efficient was using the semantic measures algorithms. These algorithms were used to reorganize the outputs of the information resources according to the users needs. The researchers have many problems and challenges to achieve for a durable solutions [13,12,17,20]. Some researchers in [20] used the method of CAS to understand better the behaviour of the Web as a complex system, where others use the MAS to simulate in a virtual way [13]. In the recent years, researchers start to do combinations between these two systems, and represent this environment as a heterogeneous paradigm that analyses the web from the point of view of the CAS once and MAS in the other. Using these two systems was to come over the non linear growth in the Web information resources and in its complexity. Users are a very important factor in these systems; force the researchers to study their behaviours in order to understand better the Web and the changes that emerge according to this usage. In [19] they present the Web components as an accurate access to the information in the Web information. However, some researchers [18] used some usage information to reorganize the Web resources.

In [11,19, 20] Hassas, Rattrout and Rupert, exploiting the web as a CAS, they have proposed a framework for developing CAS for complex networks such as the internet and the Web using stigmergy mechanism.

They used the situated multi-agents paradigm and behavioral intelligence for identifying the agents, and their roles (tags). Further, they used a mechanism of communication between agents, based on a spatial representation and mediated by the environment, such as the stigmergy mechanism. This favours the aggregation of control information and its spreading through the distributed environment. Finally, they Maintained equilibrium between exploration and exploitation in the behavior of different agents, to allow aggregation (reinforcement) of the agents and diversity (randomness).

In [11, 12, 18, 19, 20] Hassas, Rupert and Rattrout illustrated a model where the CAS principles are applied in the context of web Content organization called WACO. WACO (*Web Ants Content Organization*) is an approach, to organize dynamically the web Content. The internal organization system that is followed by the WACO model is very closed to various functions followed by the ants in there Complex System of work.

This system is made up of four agents with various jobs. The first agent (*Explorers Web Ants*) would be responsible for the process of discovery of the places of web document in random way to sort it. The second agent (*Collectors Web Ants*), whose function is to keep and organized semantically collected documents. The third agent (*Searchers Web Ants*) whose job is to enforce cluster of collected documents by searching the web for similar documents to add to the cluster. The fourth agent (*Requests Satisfying Web Ants*) whose job is search for the appropriate clustered based on user query .The various groups of Web Ants can achieve their tasks and work together following the feedback system without having to get direction from the internal center. Each semantic topic is identified by a kind of pheromone. But WebAnts can work in a group through a stigmergic, using a multi-Structured electronic pheromone. Synthetic pheromone is coded by a Structure with these different fields:

- Label (W_{ij}): decide the sort of information classified by the pheromone, which is in our Content the semantic value of a document (*weighted keyword*).

$$W_{ij} = L_c \cdot H_c \cdot T_f \cdot IDF$$

T_f is the number of times of the term in the present document, the H_c is a Header constant ($H_c > 1$ if the word shown in a title, =1 otherwise), which increases the weight of the term if it is shown in the title of the document, and IDF_k is the inverse of document frequency. The linkage constant L_c ($L_c > 1$ if the word is shown in a link, =1 otherwise)

- *Intensity* (t_{ij}): which is the term that shows how continuous is the flow of information about a certain topic and how high is the value of the pieces of information and also their attractive power. Every time ($t+1$) anew document is found, it adds to the site i and so to the topic j ,

as:

$$t_{ij}(t+1) = r_j \cdot t_{ij}(t) + \sum_{k=1, [D_{ij}]} D t_{il}^k(t)$$

r_j represents the persistence rate ((1- r_j) the evaporation rate), $D_{ij}^k(t)$ the intensity of pheromone comes out by a document k , on the site i for a topic j at time t , and D_{ij} is the set of documents addressing topic j on the site i .

- *Evaporation rate*: it shows the stability of the rate of information on the specific field. So, if the value of information is low, its influence will be longer, and that value is always calculated in relation to the ratio of documents related to that topic and compared to all documents in that site i .

$$\text{where } r_j = |D_{ij}| / |D_i|$$

D_{ij} is the group of documents about the topic j on the site i , and D_i is the amount of all documents on the site i . Our job is to make the clustering of documents of a certain topic more relative than separated ones. When the site has different semantic Content, it is none as insufficient pertinent and then the joined pheromone is going to evaporate faster than that emitted by the different Content.

- *Diffusion rate*: when a piece of information has higher value than other pieces of information then its scope of information is grater and it spreads in the environment faster. When we browse the Web looking for a topic we express this distance d_{ij} using the linkage topology information. And so we can explore the topic of interest by associating to each site i . The distance of the topic is characterized as the longest path from the site to the last site addressing the topic j , following a depth first search

$$d_{ij} = \text{Max}_k (d_{ij}^k)$$

k is the number of links addressing topic j , from a site i . The idea here is to make sites that are a good entrance point for a search, have a wider diffusion scope than the other ones. Doing so, we could guide the search process to handle queries like "Give me all documents k links away this one". The Web Ants in WACO are created in a dynamic way and they adapt to their environment and co-evolve. Two mechanisms direct their life cycle: *duplication* (birth) and *disappearance* (death).

The WACO Model has achieved the results it has promised. The number of sites in neighborhood can give us an idea about the function of that site locally and its relation to other sites with similar Content. By study, we noticed that disorder decreases all the time in that system while new document's appearances increase. They measure disorder by the total number of document minus the number of clustered documents and this can also show us the effectiveness of the clustering behavior. They tell that there is an evaluation and an increase is taking place every time in sizes of clusters and tell us that clustering behavior reinforce of the creation of clusters. These agents are increased when clusters are formed in large numbers and they decrease when clusters are not formed. If there is a sudden increase in there energy, new cluster appear specially when new documents are discovered or new sites are created. Agents increase in population and a lot of evaluation happen during time which leads to regulation of their activities. They know that all agents are active, and by the time the active agents increase and the inactive ones disappear. And by this way they reduced the number of initial

agents. But during the formation of new clusters and the creation of new sites, all agents become active agents again.

4. Webcomp Model

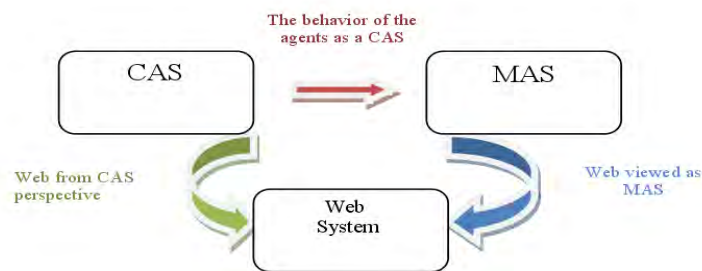


Fig1. Interrelated relation between the three major items that represent the problematic and the proposed solution

In this model, we present the problem from three concepts as in figure1.

4.1. A Model to Combine Web Components

In [18] Rattrout et al. suggest a multi-scale space model called Webcomp. Pages connect to each other whereby hyperlink. Web Pages have Content, Usage, Structure, and values of semantic. Finding the effective approach is the main idea which makes us able to do aggregation between various spaces with out losing the semantic [17] value of pages. We can enable the interaction within a single space of different spaces by using the agent's communication. Figure.2 represents the interactions between these components.

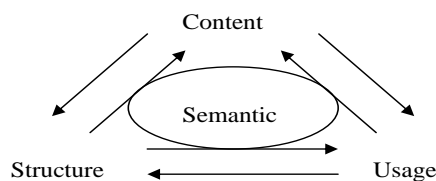


Fig 2. The interaction between the components of the Web

There are two levels of space of this model: the real main space, and the virtual sub space. The virtual spaces enrich the real ones by adding information that are related to their dynamic Structure. Different agents can

form the interaction between two pages specially the agents that have the same Head Tags and different values of semantic. We can also combine previous agents that are based on identical positive tags by using the resulted similarities. We can also combine a Multi Agent System by using further agents. We can know about interactions between different space levels by following the virtual dimension and according to the degree of similarity.

4.2. The Multi-Scale Space Definition

The Total Documentary Space (TDS) is group of pages tagging from the Web by searching engines according to one (or a set of) keyword(s) (kj). We can use algorithms of Page rank, Best First Search, InfoSpider [13-16] to make a variety of Web documents (W). These pages (pi) are downloaded locally with the goal of applying all the needed process.

The page (information) is defined as follows:

$$TDS = \{ \forall p_i \in W / k_j \in p \}, \forall p_i \in W \ \& \ p_i \in TDS$$

$p = \langle \text{information Content}(I_C), \text{information Structure}(I_S), \text{information Usage}(I_U) \rangle$.

- $I_C = \{ k / k \in p \ \& \ w_{p,K} = f(p,k) \cdot i(k) \}$, Where $f(p, k)$ is the frequency of Keywords K in the page. $i(k)$ is the opposite of the logarithmic frequency with regard to the page.
- $I_i = \{ l \in L / i \in L_{input_p} \ \text{OR} \ i \in L_{output_p} \}$, where $I_i = \langle I_S, I_U \rangle$ is the keyword frequency in the page. $i(k)$ is the inverse frequency. We should consider an extraction algorithm for the set of $L_{output_p}, L_{input_p}$.
- $I_u = \{ idp_i / idp_i \in Hp_i \}$, where idp_i is the page identifier according to its historic Hp_i .

L Stands for the set of links that exist in TDS . Two verities of agents are known for the Usage of a page p_i , a user agent (AgU_j) and an abstract agent. A user agent communicates with the agent of the page p_i (AgP_i) to register the last page visited by the user from p_i . So, a history is shared between different agents for the same page. The last information is the major type head tag connected directly to the following level of document spaces. IC denotes the head tag for the agent representing the Content space. IS denotes the head tag for the agent representing the semantic space. II denotes the head tag for the agent representing the space of links. Finally the IU denotes the head tag for the agent representing the Usage space. An abstract agent is created for each space according to the type of its main tag.

4.3. Main Level Spaces

1. Link Space (EL): is a graph space resulting from link similarity where,
 $E_L = \{\forall p_i, p_j \in TDS / p_i \sigma_L p_j\}$.
2. Content Space (EC): is a graph space resulting from similarity computation based on Content similarity where,
 $E_C = \{\forall p_i, p_j \in TDS / p_i \sigma_C p_j\}$.
3. Usage Space (EU): is a graph space resulting from Usage similarity computation based on the Usage information similarity where
 $EU = \{p_k \in (Ep_i \cup Ep_j) \& p \notin TDS\}$ (Epi) is the space of visited pages starting from pi whoever the user is (ui). The similarity is calculated between the two spaces of the two pages (pi, pj).
4. Semantic space $E_s = \{\forall p_i, p_j \in TDS / p_i \sigma_s p_j\}$ is applied on the three different spaces in order to have a semantic significance for spaces.

4.4. Content Similarity σ_c

Every relation between two pages in the Web is based on their relation for n terms. The similarity measures σ can be defined from distance measures δ using the relationship [15-17]:

$$\sigma = 1 / \delta + 1 \quad \text{Where} \quad \sigma(p, q) = \frac{(\vec{p}, \vec{q})}{(\|\vec{p}\| \cdot \|\vec{q}\|)}$$

\vec{p}, \vec{q} are representations of the pages in word vector space after removing stop words and stemming. This is actually the "cosine similarity" function, traditionally used in information retrieval IR.

4.5. Semantic Similarity

A semantic similarity between two documents is defined in [17] using the entropy of the documents' respective topics:

$$\sigma_s(d_1, d_2) = \frac{2 \log \Pr[t_0(d_1, d_2)]}{\log \Pr[t(d_1)] + \log \Pr[t(d_2)]}$$

where $t(d)$ is the topic node containing d in the ontology, t_0 is the lowest common ancestor topic for d_1 and d_2 in the tree, and $\Pr[t]$ represents the prior probability that any document is classified under topic t .

4.6. Link Similarity

Link similarity is defined with

$$\sigma_l(p, q) = \frac{|U_p \cap U_q|}{|U_p \cup U_q|}$$

where, U_p is the set containing the URLs of page p 's out-links, in links, and of p itself. Out-links are obtained from the pages themselves, while a set of in-links to each page in the sample is obtained from the list of the table of the out links that point to the pages exists. This Jacquard coefficient measures in [17] the degree of clustering between the two pages, with a high value indicating that the two pages belong to a clique.

4.7. Similarity correlation and Combinations

In [17], there was study by Menczer to know if the various similarity measures are correlated or not. We should analyses the relation between Content, Structure, and semantic similarity functions so as to make a map of the correlations and functional relationships between the three measures a cross tow pages. Menczer tried to approximate the accuracy of semantic similarity by mapping the semantic similarity landscape as a function of Content similarity and link similarity [16]. Averaging highlights of the expected semantic similarity values is akin to the precision measure used in IR. Summing captures of the relative mass of semantically similar pairs is akin to the recall measure in IR. Let us therefore define localized precision and recall for this purpose as follows:

$$R(s_c, s_l) = \frac{\sum_{p,q} \delta_c(p, q, s_c) \delta_l(p, q, s_l) \sigma_s(p, q)}{\max_{s_c, s_l} \sum_{p,q} \delta_c(p, q, s_c) \delta_l(p, q, s_l) \sigma_s(p, q)}$$

$$P(s_c, s_l) = \frac{\sum_{p,q} \delta_c(p, q, s_c) \delta_l(p, q, s_l) \sigma_s(p, q)}{\sum_{p,q} \delta_c(p, q, s_c) \delta_l(p, q, s_l)}$$

5. Agents' Modelling

We can extract the Web pages from the internet by using many types of search engines [16] and this can give us the advantage of having high quality pages. We have chosen four search engines to cover all the possibilities for sorting out the pages that are characterized by criteria such as Content, Relevance, Popularity, etc.

Our Model makes an original agent that is known as abstract agent every time user opens the session. Their will be three types of agents for the successor concerning it's Content, Structure, and Usage, and each one will

represent the space that explores according to similarity rules used by Menczer [14].

Three Agents Content, Structure, and Usage, roll the major activities in our model by the way of tagging the documents of the TDS, carrying back concerned page's tag and similarities resulted to the TDS. The abstract agent tagged by a tag and contains a similarity value. We can't find information about the existence of tags and similarities in the initial state, only the type of tag signed by the agent itself. For each sub-space, an agent created, contains certain information of its space only when it has acquired enough resources to transfer its information, it becomes active. The agent fits its ability to produce offspring. Another sub-space agent is created and starts to specify their tags according to their properties and results of similarity process. After that an aggregation process holds in, with adhesion between agents represents their spaces forming multi-agent aggregates.

5.1. Resources

We can lay the organization of the agent model if we specify a group of renewable resources of documents information that are dealt within an abstract way. We can also represent agent's resources by following Special characters concerning the type of their original correlated information. In the aggregation process, some mechanisms become active like tagging the different types of resources.

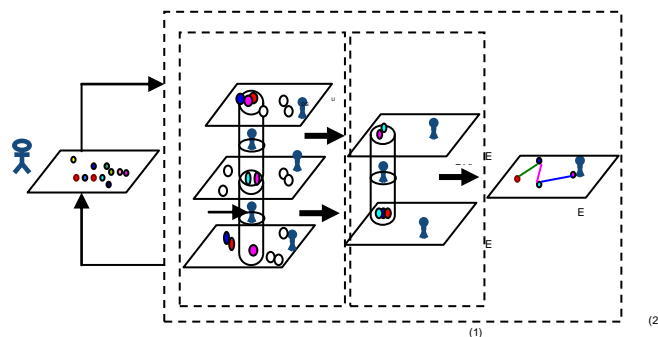


Fig3. Multi-scale space of the Web and life looping

5.2. Tagging

The “tags” analyzed as word vectors of information that could be shared by different agents. They are the marks that distinguish types of common agents from others. In our model, there are three types of tags, space tags, document tags, and value tags; space tags are classified into four types related to the four sub spaces (ex. Content space tag is TagTc), where the

document tags are the word vector (frequency of terms > threshold) exists in its Content, links, and Usage information, finally the tags value which marks the pages with tag+ or tags – according to their similarity values where it is above or under the given threshold. The use of Menczer similarity σM over TDS in the beginning of our algorithm, will guarantee that the pages will be classified into three different spaces. The Web spaces are ordered using the decreasing values of their similarity. The degree of similarity is also used for the adhesion between agents. Using the tag+ could be considered a starting point in doing aggregation between two agents from the same space or from different spaces. We can determine the tag while we are calculating the similarity values within the same space. The similarity value helps to discover the similar area between two agents.

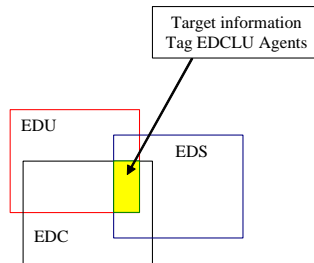


Fig 4. commons spaces zone where agents could find documents that are shared by their different characteristics

5.3. Model's Function

Our model integrates two types of users, authorized and invited. The first is related to user name and password for the documents treated or used by the user himself. Usage information will be the Usage space information that Usage agents will work in. Similarity, values and tags are found randomly in the sub space pages by the Agents in this model, which have been treated by the tags and similarity agents, follow mining process enriching pages in the TDS. These agents add the positive values and tags to a reserve concerning p_i , where these information are considered as input resources for the TDS agents in their space. From collected values, an agent goes out from one page to another, looking for values that match its information. TDS agents could have the three types of information including their total tags. In Figure 6, we show how our agents travel from sub spaces to TDS and communicate between each other in one space. In phase (1) one agent enter to a page randomly, trying to recognize the space that this page belongs to. This information is taken from the tagT, why start with this information? To avoid mixing the space's information while it tries to specialize in their activities, also, to mark the pages that have a high value in their spaces. The agent which is considered as a Content agent when it finds a page with a high value of Content similarity, starts to travel from page to page with the values

that this page has until it finds a Tag_v in this page that has value above λ while its Tag_v, c is positive, and the energy tag is also positive. At this moment the agent will change the space to see the page that has a power value from another space. Next the agent from the page value, the agent will continue in exploring the space until there are no more pages to be visited, and then it will die.

5.4. Explorer Agents

Explorer agents are created to collect information from each page visited by one of the sub space tags and similarities agents, and send it to the TDS. The pages in the TDS have four reserve boxes related to their similarities values, and marked by four different types of signals with an independent reserved tag. If page p_i frequently visited by the sub space's agents, recording its new values of similarities, and fulfilling the threshold condition; there exists an agent's trace that will stay sending signals to the TDS agents. Matching signals between agents depends on the strength of the signal itself according to the similarities values. Those who have high signal will do an adhesion, constructing a multi agent's level; otherwise, they will stay individuals.

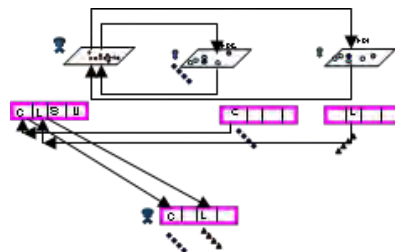


Fig 5. Explorer agents enrich the TDS page by collecting the Tags and similarities values from virtual space.

5.5. Agent's Organisation

In our model, we use the conceptual Agent/Group/Role, where many types of agents are created. Abstract, simple, global, generator, tags, space explorers, similarities, and information collectors' agents are the main variety of agents who works on documents Content, link, and Usage information. Space information considered as Food for our agents of type search and collector, or tag, where their values of similarity are used also as collected food, but with different assess, and for each agent we have a brain, that generally derives from the abstract agent. These agents roll the major activities in our model by tagging the documents of the TDS, calculate the similarities between pages in each document space, explore the spaces to

find which pages have been treated, and collect the tags and similarities values resulted from each space and register it into the TDS.

The Global agent contains two components, the tags information and similarities values.

The generator agent is the agent who generates the agent similarity. Many types of tags exist in our model, as Content tags; and these tags tagged the different sort of information. Links tags, which contains tags tagged the information in the links and tagged the Structure of the pages, and finally Usage tags, where Usage tags are divided into two types; Usage text tags and Usage links tags. To understand this behavior we added another types of tags explaining the matching tags in the different steps of aggregation property.

5.6. Agent's Construction

The agent's similarities are three types in Similarity group where each agent has many simple agents:

1. Agent Content similarity

This agent calculates the Contents similarity between pages P_i, P_j , where P is the set of the pages of the Total document space EDT as follows:

AgentSimilarity (Agent/Group/Role)

Input: Page $P = \{P\}$

$$\sigma_c(p, q) = \frac{\sum(TF.IDF_{t,p} * TF.IDF_{t,q})}{((\sum(TF.IDF_{t,p})^2 * \sum(TF.IDF_{t,q})^2)^{1/2}}$$

Output: Space (S_c)

Where

$$\text{Space } (S_c) = CDS$$

For each page P_i we calculate the summation of all the similarities between this page and the rest of the pages in the space, divided by $n-1$ as follows:

$$\sigma(c, P_i) = \sum \sigma_{cP} / n - 1$$

2. Agent link similarity

This agent works with respect to the following rule on EDT and the output will be a virtual space called Link document space:

$$\sigma_l(p, q) = \frac{|U_p \cap U_q|}{|U_p \cup U_q|}$$

We also calculate for each document the similarity value according to the other documents as:

$$\sigma(L, P_i) = \sum \sigma_{L, P} / n - 1$$

Input: Page P= {P}
Output: Space S where
SpaceL=SDL

3. Popular Link similarity:

The popular link similarities used as tool to refine or to adjust the results we have from the classical measure in the link similarity tools. We can form the popular link similarity as:

$$\sigma_l(p,q) = \frac{|U_p \cap U_q|}{|U_p \cup U_q|} * Pop$$

Where Pop is the popularity measured between two inlink factors taken from the concerned two pages p and q. Pop calculated as:

Pop = Max(Pinlink, Qinlink) / Min (Pinlink, Qinlink)
If Pop is over threshold done by the user then the pop=1
Else Pop = Pop
End

4. Agent Usage similarity

In the initial state, no information about tags and similarities exist, only the tag signed by the topic itself.

Input: Page P= {P}/ P is the historic of the pages registered when we open a session.
Output: Space S where
SpaceG=SDU
Page P= {P}/ P is the historic of the pages registered when we open a session.
For SDUC
Do
AgentContentSimilarity
Input: Page P= {P}
Output: Space S where
Space= SDUC
For SDUL
Do
AgentLinkSimilarity
Input: Page P= {P}
Output: Space S where
Space= SDUL
End

The Tags agents groups are started by the following types of agents tags:

- TagCG
CCT - Tag Content title: it tagged the information existing in the title and subtitles; C= T/min fw where T is the threshold of frequency of words

CCP - Tag Content paragraph: it tagged the information existing in the context with fw>T

ILC - Tag Content information Link: it tagged the information existing in the link information text with fw>T

- TagLG
 - TLW: it tagged the words which are hyperlinked to out-links
 - TLL: it tagged words existing in the out-links or the in-links
 - TLI: it tagged the information existing in the links information text with fw>T
 - TLP: it tagged the information existing in the hyperlinks over a paragraph.
- TagUG

```

TagAgentUsag( Agent/Group/Role)
  <OpenSession>
    Opp: get Usage space tag
  Do
    For space = DUS
      Get TUS Auth
      Get TUS inv
      Get TUSc
      Get TUSl
    End

```

5.7. WebComp Agents Composition

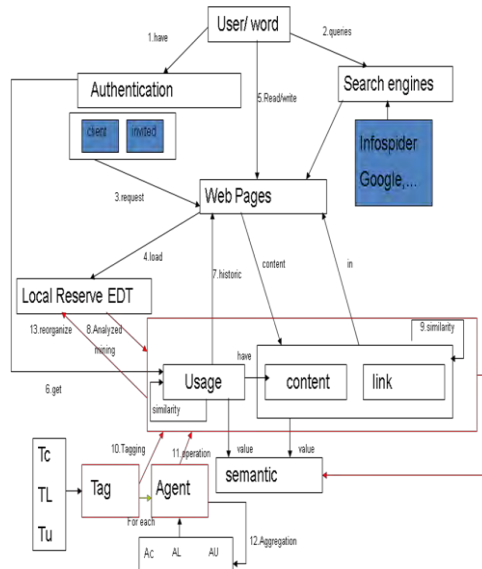


Fig 6. Webcomp architecture

In the WebComp approach the agents are created in a dynamic way. We start by creating an agent called “generator agent” which controls the entire

agent creation mechanism. The WebComp agents are sensitive to some notion, of the process order, where the similarity agent works before the collector agent, and the tagging agent works before the aggregation one. Activity of agents depends on web resources (pages in the spaces), where the higher the pages exist in the space; the more the activity happens in the space.

5.8. Enforcing the Self Organization by Using the Usage Space Information

Usage information has a great effect of recovering the relevant pages to the user's demand. Who is looking forward to finding what they are looking for Usage information includes two factors that could possibly affect the efficiency of page recover from the web. These factors are: First, using the same topic frequently through search engine process. Second, the time spent by users to browse the page. Furthermore, there are various factors that vary dynamically influencing the suitable data process, add, update, and delete. Of any page or site i.e. this change occurs in a non-linear manner.

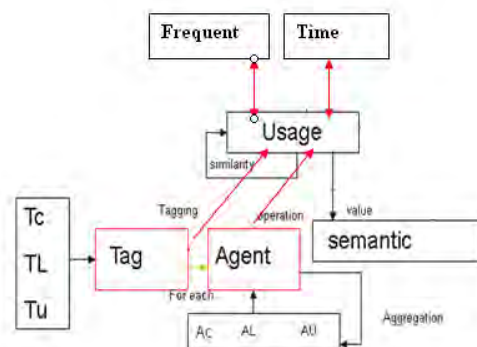


Fig7. Usage space

5.9. Definitions: Usage information:

- The Usage Space represents the information extracted from the session's history for each user invited or authorized.
- Space (Su) represent the environment where agent is living as a member in a set (Group(Gu)).the Web Pages represent the target for agent to get their information.
- Usage Space: Agent, Page, Operation, Tag, Time, Frequent. Where the agents are navigating and exploiting the entities searching for similar information to their tag. The agent defined as a set of (Group(G)) of agents that navigate in his space, search and collect specific information

according to its speciality. Page is a set of target pages, Operation is the role that the agent will do, and, aggregation, Tag is the kind of information that illustrates each agent. Time is the time that the user spends in using Web Page. And Frequent is the summation of term that user used it.

- Usage Agents: UsPace, Operation, Tag, Life time cycle, Number of agents exist in agent's group, and type where the link, time, and frequent information represented types of information.
- Role "operation": is an abstract representation of an Agent function service or identification within a group. Each agent can handle a role, and each role handled by an agent is local to a group.
- Group: a set of Agent Aggregation. Each agent is part of one or more group formed by three spaces (Content, Structure, and Link)
- The Usage space is construct from those URLs that the user used concerning a topic, knowledge, and technique in exploiting the information while he is exploring the Web. Let $U = \{u_1, u_2... u_w\}$ be the set of pages in the user's historic and $P = \{p_1, p_2... p_v\}$ the set of pages in the EDT where EDT is the set of all the pages collected by the search engines for a keyword. Using similarity rules between the user site U and the pages collected by the search engines P could be useful.
- Here, the user is interested in finding some pages in P that are similar to a page in U. For example, the user reads an article on a particular topic and may want to know what has been published on similar sites on the same topic.
- Our comparison is done as follows: Given a U page u_j , we use the cosine measure to compute the similarity between u_j and each page in P. After the comparison, the pages in P are ranked according to their similarities in a descending order. Example: In this example, we have 4 pages in U and 3 pages in P, which are shown bellow (the number in each pair is the frequency of the keyword in the page).

If we want to find the corresponding page(s) of Upage 1, we obtain the following ranking:

Rank 1: Ppage 1 Rank 2: Ppage 3

Ppage 2 is not shown in the ranking as its similarity value with Upage 1 is 0.

U pages	P pages
Upage ₁ : (office, 1), (home, 1)	Ppage ₁ : (office, 2), (home, 2), (Web, 3)
Upage ₂ : (information, 2), (mining, 1), (office, 2)	Ppage ₂ : (association, 3), (mine, 2), (rule, 1)
Upage ₃ : (Web, 2), (probability, 2)	Ppage ₃ : (clustering, 3), (segment, 2), (office, 2)
Upage ₄ : (clustering, 2), (segment, 1)	

5.10. Communication Agents and Aggregation Property

By using the terms in the space for Content terms agents can communicate with each other. And it is straightforward to them to communicate if it is clear that they point at the same set of terms. If we link these spaces, the agents can commit the terms consistently with the Usage mandated in that space. But if they are not using the same space, they may still be able to communicate. Any agent can communicate with another if there is a common document space and if all mapping where perfect.

6. Results of Webcomp Model

In this section we will illustrate how our approach can reduce the complexity and reorganizing effort. That includes three steps: step1: Collecting Process and creating TDS. This is the first step of our algorithm. The URLs are stored in the TDS by using keywords as a topic table and are ready to get processed by the mining algorithms and their basic similarities. Step2 TDS is categorized into two sub spaces (Content, Link) according to it information type. Content Similarity calculated by TFIDF among all pair of pages. And Link similarity calculated using similarity rules as shown in table 1, Person Correlation coefficient applied over the results between the two majors for TDS. As shown in figure 8 At time t=0 just Content and Link spaces are formed. Step3: in t>2 user start using the system. As a result, Usage space is created and Usage agent's algorithms start working. As a result, we can find that when a page is added from the Usage space to the TDS, another is brought from the edge of TDS to the center. So, a correlation starts emerging.

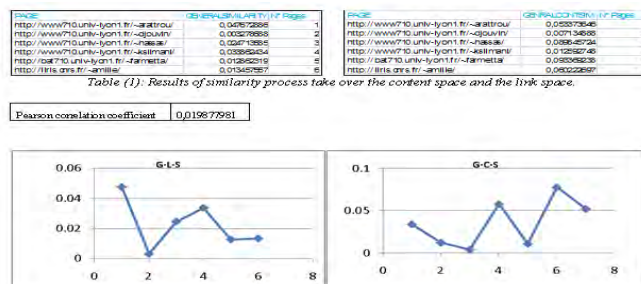


Fig 8. The general link and Content similarity distribution for the TDS at t=0.

Pages added to Usage space are compared if they found in TDS or not. If not added to TDS and recalculating and re- ranking is followed by it as shown in figure9. As a result a new TDS is created. In Usage space, Content and Link similarity is calculated in the same way as it is in TDS. Tabel2 illustrate that:

- If another page is added from Usage space to TDS, which is different from the first page, we can see that the correlation is becoming clearer figure10.

By adding another page at t=4. It is found that the system has transformed into a new form of aggregation and that correlation becomes higher and higher as shown in figure 11 and 12.

PAGE	GENERAL SIMILARITY IN PAGES	PAGE	GENERAL CONTENT SIMILARITY IN PAGES
http://www.710.univ.lycent.fr/~fahad/NAKCS07	0.01102465	1	http://www.710.univ.lycent.fr/~fahad/NAKCS07
http://www.710.univ.lycent.fr/~ahadtrou	0.04039963	2	http://www.710.univ.lycent.fr/~ahadtrou
http://www.710.univ.lycent.fr/~ahadtrou/	0.0273224	3	http://www.710.univ.lycent.fr/~ahadtrou/
http://www.710.univ.lycent.fr/~ahad	0.02732087	4	http://www.710.univ.lycent.fr/~ahad
http://www.710.univ.lycent.fr/~ahadman	0.03219985	5	http://www.710.univ.lycent.fr/~ahadman
http://www.710.univ.lycent.fr/~ahadmetla	0.01718589	6	http://www.710.univ.lycent.fr/~ahadmetla
http://www.710.univ.lycent.fr/~ahad	0.01214632	7	http://www.710.univ.lycent.fr/~ahad

Table (2). The new TDS at t=2, weak correlation exists

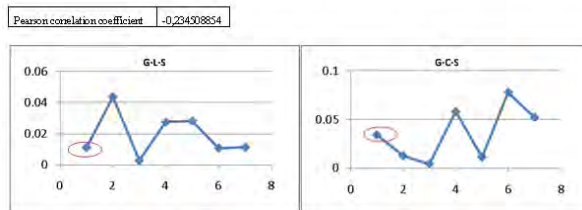


Fig 9. The general link and Content similarity distribution for the TDS at t=2, we can see that emergence of new changes in the distribution take place.

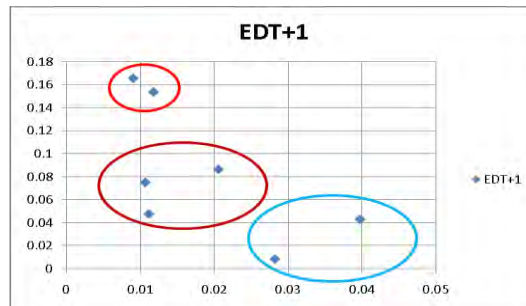


Fig10. The σ_c , σ_L , distribution for new TDS at t=2, good correlation emerges, and aggregation can happen.

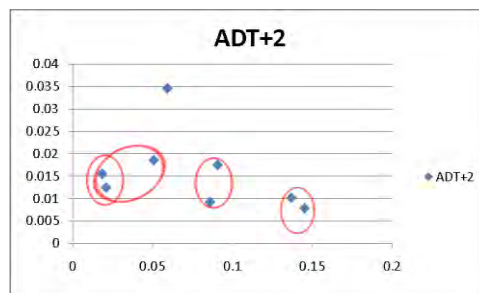


Fig11. The σ_c and σ_L , distribution for new TDS at t=4, good correlation emerges, and aggregation can happen between three or more agents.

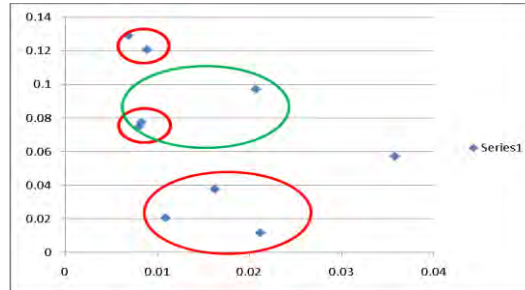


Fig 12. The σ_c and σ_L distribution for new TDS at $t=6$, good correlation emerges, and aggregation can happen between four groups of agents.

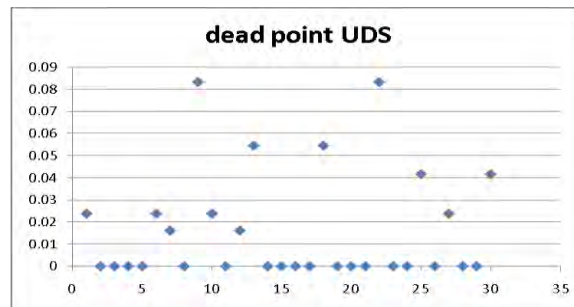


Fig13. Dead point in the UDS where the similarity value is equal to zero. These values isolate the concerned pages from the tagging collecting and aggregation later on

7. Conclusion

In this paper we have presented an analysis of the Web as a complex adaptive system (CAS), and have proposed its modeling using the seven characteristics and mechanisms, which are proposed by J. Holland, to overcome its evolving complexity. It focused on the association of a semantic, to information contained on the web, through the combination of the web content, usage, structure, the time factor and frequent information exist in Web Page and their interrelated multi-scale spaces are enriched by the Usage information. The proposed model opens a new approach in the web-searching domain using complex adaptive systems, properties and mechanisms especially Non-linear, and Multi-agent system paradigm in order to reduce the complexity of the complex web. The Usage Space is opened for researches and several scholars are investigating this matter.

References

1. J. Holland, *Hidden Order*. Addison-Wesley, 1995.
2. S. Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, 1993.
3. W. B. Arthur, S. Durlauf, and D. Lane, *The Economy as an Evolving Complex System II*. Addison Wesley Longman, 1997.
4. E. Mittleton-Kelly, "Organisations as Co-Evolving Complex Adaptive Systems," presented at British Academy of Management Conference, 1997.
5. S. Levin, "Ecosystems and the Biosphere as Complex Adaptive Systems," *Ecosystems*, vol. 1, pp. 431-436, 1998.
6. G. W. Flake, S. Lawrence, C. L. Giles, and F. M. Coetzee, "Self-Organization of the Web and Identification of Communities," *IEEE Computer*, vol. 35, pp. 66--71, 2002.
7. J. Kleinberg, "Authoritative Sources in a Hyperlinked Environment," presented at ACM-SIAM Symposium on Discrete Algorithms, 1998.
8. J. Liu, S. Zhang, and Y. Ye, "Understanding emergent Web regularities with information foraging agents," presented at Proceedings of the first international joint conference on Autonomous agents and multiagent systems, 2002.
9. R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network motifs: simple building blocks of complex networks.," *Science*, vol. 298, pp. 824-827, 2002.
10. M. Kirtland, "The Programmable Web: Web Services Provides Building Blocks for the Microsoft .NET Framework," in *MSDN Magazine*, vol. 15, 2000.
11. S. Hassas, "Systèmes complexes à base de multi-agents situés," University Claude Bernard Lyon, 2003.
12. S. Hassas, "Using swarm intelligence for dynamic Web Content organization," presented at IEEE Swarm Intelligence Symposium, Indianapolis, IN, USA, 2003.
13. F. Menczer, A.E. Monge: Scalable Web Search by Adaptive Online Agents: An InfoSpiders Case Study . In M. Klusch, ed., *Intelligent Information Agents* , Springer, 1999
14. F. Menczer, R.K. Belew: Adaptive Retrieval Agents: Internalizing Local Context and Scaling up to the Web .*Machine Learning Journal* 39 (2/3): 203-242, 2000.
15. G. Pant, F. Menczer: MySpiders: Evolve your own intelligent Web crawlers .*Autonomous Agents and Multi-Agent Systems* 5(2): 221-229, 2002 *Systems* 5(2): 221-229, 2002.
16. F. Menczer, G. Pant, P. Srinivasan: Topical Web Crawlers Evaluating Adaptive Algorithms, *ACM TOIT* 4(4): 378-419, 2004
17. F. Menczer: Lexical and Semantic Clustering by Web Links .*JASIST* 55(14): 1261-1269, 2004.
18. A.Rattrout, M.Rupert, S.Hassas," Reorganizing Dynamic Information Resources in the ComplexWeb,"*ACIT2005*, Jordanian. pp. 103-110. ISSN ISSN: 1812-0857. 2005.
19. M. Rupert, A. Rattrout, S. Hassas, "The web from a complex adaptive systems perspective," (*Journal of Computer and System Sciences*) *Journal ELSEVIER* 2008
20. M. Rupert, S. Hassas, A. Rattrout,"The Web and Complex Adaptive Systems," *AINA* (2) 2006: 200-204 IEEE.

Amjad Rattrout, PhD, Assistant Professor, Received the M.Sc. and Ph.D. degrees in computer science from the Ecole des Mines and Claude Bernard Lyon1 University in 2002 and 2007 respectively. Actually he is a researcher and lecturer in the University of Claude Bernard Lyon1, Lyon, France. LIRIS laboratory at research group: Knowledge and Complex Systems (CoSyCo), and SILEX team: Supporting Interaction and Learning by Experience. He is member in Complex Systems and Multi Agents paradigm team, and member in the « Association Francophone de Recherche d'Information et Applications » (ARIA). His research addresses the study, programming and modeling of complex, dynamic and distributed systems in the Web using Multi Agent Systems paradigm with respect to the Complex Adaptive Systems perspectives. His PhD topic: Dynamic organization of distributed sources of information following the complex systems perspective. He is member of technical committee of IAJIT and member in many international conferences technical committees like EUPS-09 USA, SKIMA09, ACIT, ICIT09, and ICIT2011. He published many journal and conferences papers. He is the coordinator of Franco-Palestinian project PLUS II - Project MAS-CS. Interested in the Complex Adaptive Systems, Complex Web and Multi Agents Systems paradigms, AI.

Assaf Rasha Received the BSc in Computer Information System from AL-Quds Open University, Jenin-Palestine 2002 , and her Master in computer science from Alquds University, Jerusalem, Palestine 2009. Lecturer since 2006 in Al-Ummah College, AL- Ram, Jerusalem. Her scientific interests directed inthe Web intelligence systems. She published one paper in the same domain.

Al-Dahoud, is an associated professor at Al-Zaytoonah University, Jordan. He worked at Al-Zaytoonah University since 1996. He established the ICIT conference since 2003 and he is the program chair of ICIT until now. He has directed and led many projects sponsored by NUFFIC/Netherlands. He participates in the following conferences as general chair, program chair, session's organizer or in the publicity committee: ICITs, ICITST, ICITNS, DepCos, ICTA, ACITs, IMCL, WSEAS, and AICCSA. Al-Dahoud worked as Editor in Chief or guest editor or in the Editorial board of the following Journals: JDIM, IAJIT, JCS, IJITST, and UBICC. He published many books and journal papers.

Received: June 08, 2009; Accepted: March 29, 2010.

A Hybrid Variable Neighborhood Search Algorithm for Solving Multi-Objective Flexible Job Shop Problems

Jun-qing Li¹, Quan-ke Pan¹, and Sheng-xian Xie¹

¹College of Computer Science, Liaocheng University
Liaocheng, 252059, People's Republic of China
{Lijunqing, qkpan, xsx}@lcu.edu.cn

Abstract. In this paper, we propose a novel hybrid variable neighborhood search algorithm combining with the genetic algorithm (VNS+GA) for solving the multi-objective flexible job shop scheduling problems (FJSPs) to minimize the makespan, the total workload of all machines, and the workload of the busiest machine. Firstly, a mix of two machine assignment rules and two operation sequencing rules are developed to create high quality initial solutions. Secondly, two adaptive mutation rules are used in the hybrid algorithm to produce effective perturbations in machine assignment component. Thirdly, a speed-up local search method based on public critical blocks theory is proposed to produce perturbation in operation sequencing component. Simulation results based on the well-known benchmarks and statistical performance comparisons are provided. It is concluded that the proposed VNS+GA algorithm is superior to the three existing algorithms, i.e., AL+CGA algorithm, PSO+SA algorithm and PSO+TS algorithm, in terms of searching quality and efficiency.

Keywords: Flexible Job Shop Scheduling Problem; Multi-objective; Genetic Algorithm; Variable Neighborhood Search.

1. Introduction

The job-shop scheduling problem (JSP) is one of the most popular scheduling models existing in practice, which has been proven to be among the hardest combinatorial optimization problems [1, 2] and has got more and more research focus in recent years. The Flexible job-shop problem (FJSP), an extension of the classical JSP, is harder than the latter because the addition of assignment of a suitable machine from a set of candidate machines for each operation.

The FJSP recently has captured the interests of many researchers. The first paper about solving the FJSP was proposed by Brucker and Schlie (Brucker & Schlie, 1990) [3], which gives a simple FJSP model with only two jobs and

each operation performed on each machine with the same processing time. The first author with the hierarchical idea to solve the FJSPs was Brandimarte (Brandimarte, 1993) [4], who solved the first stage with some existing dispatching rules and the second stage with tabu search heuristic algorithms. Kacem (Kacem, Hammadi & Borne, 2002) [5] solved the two stage problems with the genetic algorithm (GA). Gao (Gao & Gen et al., 2006) [6] used bottleneck shifting method in genetic algorithm for solving the FJSP. Saidi-mehrabad (Saidi-mehrabad, 2007) [7] gave a detailed solution with tabu search method. Li et al. (2009) [8] presented a hybrid particle swarm optimization (PSO) combining with a fast neighborhood structure algorithm for the problem.

The research on the multi-objective FJSP is much less than the mono-objective FJSP. Kacem et al. (2002a, 2002b) [5, 9] developed an effective evolutionary algorithm controlled by an assigned model based on the approach of localization (AL). Xia and Wu (2005) [10] presented a practical hierarchical solution approach by making use of PSO to assign operations on machines and simulated annealing (SA) algorithm to schedule operations on each machine. Zhang et al. (2009) [11] developed a hybrid algorithm combining PSO with tabu search (TS) algorithm. Most of the above algorithms solved the multi-objective FJSP problem by transforming it to a mono-objective one through giving each objective a different weight.

In this paper, we propose a novel hybrid variable neighborhood search algorithm combining with the genetic algorithm (VNS+GA) for solving the multi-objective FJSP problems to minimize the makespan, total workload of all machines, and workload of the busiest machine. In the hybrid algorithm, the three objectives are also combined into a single objective by assigning each objective a different weight. GA is used to produce a swarm of candidate solutions, whereas VNS is introduced to obtain more optimal solutions around the given candidate solutions. A mix of two machine assignment rules and two operation sequencing rules are developed to create high quality initial solutions. To produce effective perturbations in the machine assignment module, two adaptive mutation rules are used in the hybrid algorithm. A speed-up local search method based on public critical blocks theory is proposed to produce perturbation in operation sequencing component.

The rest of this paper is organized as follows: In section 2, we briefly describe the problem formulation. Then, the framework of our hybrid algorithm is presented in Section 3. The GA for perturbation in machine assignment component is introduced in Section 4. Section 5 illustrates the VNS approach for local searching in operation sequencing component. Section 6 shows the experimental results compared with other algorithms. Finally, Section 7 gives the conclusion of our works.

2. Problem formulation

FJSP is an extension of the classical JSP; therefore, we can formulate the FJSP based on the JSP. Consider a set of n jobs, denoted as $J = \{J_1, J_2, \dots, J_n\}$, each job J_i ($1 \leq i \leq n$) in J has a pre-defined number of operations, and should be operated on a selected machine from a machine set named $M = \{M_1, M_2, \dots, M_m\}$. The main different between FJSP and JSP lies in two aspects: first, in the classical JSP problem, with n jobs and m machines, there are $n \times m$ operations, whereas in FJSP, given n jobs and m machines, the number of operations may large or small than $n \times m$; second, in the classical JSP, an operation should be operated on a pre-defined machine, whereas in FJSP, an operation can be operated by a set of machines. Therefore, FJSP is harder than JSP. There are two kinds of FJSP problems, i.e., T-FJSP (Total Flexible Job-shop Scheduling Problem) and P-FJSP (Partial Flexible Job-shop Scheduling Problem) [8, 9]. For the T-FJSP, each job can be operated on every machine from the set M , whereas for the P-FJSP, there is an additional problem constraint, that is, one operation of a job can be processed by a sub set of machines $M' \subset M$.

In this paper, the following objectives are to be minimized:

- (1) c_M . Maximal completion time of all machines, i.e., the makespan;
- (2) w_T . Total workload of all machines;
- (3) w_M . Workload of the critical machine or the busiest machine.

The weighted sum of the above three objective values is taken as the objective function in this study:

$$F(c) = w_1 \times c_M + w_2 \times w_T + w_3 \times w_M$$

Where, w_1 , w_2 , w_3 represent the weight assigned to the objective c_M , w_T and w_M , respectively.

The following assumptions are given in this study [8-11]:

- (1) Each machine can perform at most one operation at any time and can not be interrupted during its work.
- (2) Each operation can not be interrupted during its performance.
- (3) Setting up time of machines and move time between operations are negligible.
- (4) Jobs are independent from each other.
- (5) Machines are independent from each other.

Some useful notations are given as follows:

- Let $J = \{J_i\}_{1 \leq i \leq n}$, indexed i , be as set of n jobs to be scheduled.
- Let $M = \{M_k\}_{1 \leq k \leq m}$, indexed k , be a set of m machines.
- Each job J_i can be operated on a given set of machines M_i .
- The $O_{i,j}$ represents the j th operation of J_i .

- The set of candidate machines waiting for processing $O_{i,j}$ is denoted as $M_k \subseteq M$.
- $p_{i,j,k}$ represents the processing time of $O_{i,j}$ operated on the k th machine.
- Two sub-problems of the FJSP: T-FJSP and P-FJSP.

$$FJSP = \begin{cases} T-FJSP, & \text{if } M(O_{i,h}) = M; \forall i, h \\ P-FJSP, & \text{if } M(O_{i,h}) \subset M; \forall i, h \end{cases}$$

- Decision variables

$$x_{i,h,k} = \begin{cases} 1, & \text{if machine } k \text{ is selected for the operation } O_{i,h} \\ 0, & \text{otherwise} \end{cases}$$

$c_{i,h}$: completion time of the operation $O_{i,h}$.

The formulation of the multi-objective FJSP in this study is then given in Fig.1.

$$\min c_M = \max_{1 \leq i \leq n} \{c_{i,n_i}\} \tag{1}$$

$$\min w_M = \max_{1 \leq k \leq m} \left\{ \sum_{i=1}^n \sum_{h=1}^{n_i} p_{i,h,k} \right\} \tag{2}$$

$$\min w_T = \sum_{k=1}^m \sum_{i=1}^n \sum_{h=1}^{n_i} p_{i,h,k} x_{i,h,k} \tag{3}$$

$$\text{s.t. } c_{i,h} - c_{i,h-1} \geq p_{i,h,k} x_{i,h,k}, \quad h = 2, \dots, n_i; \forall i, k \tag{4}$$

$$\sum_{k \in M(O_{i,h})} x_{i,h,k} = 1, \forall i, h \tag{5}$$

$$M(O_{i,h}) \subseteq M, \forall i, h \tag{6}$$

$$x_{i,h,k} \in \{0,1\}, \forall i, h, k \tag{7}$$

$$c_{i,h} \geq 0, \forall i, h \tag{8}$$

Fig. 1. Problem formulation of the multi-objective FJSP

Equation (4) ensures the operation precedence constraints. Equation (5) guarantees that for each operation one and only one machine must be selected from the set of available machines. Inequity (6) indicates that the set of available machines for each operation come from the given machine set M .

3. Framework of the Hybrid Algorithm

In this study, we propose a hybrid variable neighborhood search algorithm combining with the genetic algorithm (VNS+GA) for solving the multi-objective FJSPs. The detail steps of the VNS+GA algorithm are listed as follows.

Step1. Initialization

Step1.1: Set up parameters.

Step1.2: Produce machine assignment component for each chromosome in the population C_{pop} .

Step1.3: Produce operation sequencing component for each chromosome in the population C_{pop} .

Step1.4: Evaluate each chromosome, and then obtain the best solution C_{best} .

Step1.5: if stop criteria is satisfied, then go to Step 4; otherwise, go to Step 2.

Step2: Perturbation in machine assignment component

Step2.1: Produce P_{size} child chromosomes by applying crossover operation.

For $i=0$ to P_{size}

(1) Generate a random number r , if $r < p_{select}$, then randomly select one parent chromosome in the population C_{pop} denoted as P_1 , and select the current best solution C_{best} as P_2 . Otherwise, select two parent chromosomes in the population C_{pop} at random denoted as P_1 and P_2 , respectively.

(2) Produce two child chromosomes denoted as C_1 and C_2 by applying crossover function with probability p_c on the two parent chromosomes P_1 and P_2 .

(3) Evaluate the two child chromosomes; if one chromosome from the two child chromosomes denoted as C_{be} which is better than C_{best} , then replace C_{best} by C_{be} .

(4) Select the best solution from the four chromosomes (i.e., P_1 , P_2 , C_1 and C_2), and then insert it into a temp population T_{pop} .

End for

Step2.2: Produce P_{size} child chromosomes by applying mutation operation.

For $i=0$ to P_{size}

(1) Select the i th chromosome in population T_{pop} as the parent chromosome PP_1 .

(2) Produce a child chromosome denoted as CC_1 by applying mutation operation with probability p_m^c on the selected parent

Jun-qing Li, Quan-ke Pan, and Sheng-xian Xie

chromosomes PP_1 .

(3) Evaluate the child chromosome CC_1 ; if CC_1 is better than C_{best} , and then replace C_{best} by CC_1 .

(4) Insert the best solution among CC_1 and PP_1 into T_{pop} .

End for

Step2.3: Select P_{size} better chromosomes in T_{pop}

(1) Sequence all $2 \times P_{size}$ chromosomes from population T_{pop} in descending order on chromosome fitness, that is, the chromosome with optimal fitness value will appear at the relative top position.

(2) Select the top P_{size} chromosomes as the current population C_{pop} for next generation.

Step3: Perturbation in operation sequencing component

For the current best solution C_{best} , operate the following steps:

Step3.1: Get all critical operations.

Step3.2: Get all public critical operations.

Step3.3: Get all public critical blocks.

Step3.4: Use the function *effectiveNeighbor()* to search the best neighbor solution of the current best solution. If the former is more optimal than the latter, then replace the current best solution C_{best} by the new neighbor solution. Then go to step 1.5.

Step4: Output the current best solution C_{best} and stop.

4. Machine assignment algorithm: the genetic algorithm

4.1. Genetic Algorithm

Genetic Algorithm (GA), proposed by J. Holland in 1975 [12, 13], has been used to solve optimization problems in recent years [13]. The GA is based on the genetic process of biological organisms. Several key factors including populations, crossover functions, mutation functions, evolution approaches and stop criterion are important for the efficiency of the GA. The main steps for the process of GA can be described as follows [13].

Step1: Let $k=0$. Randomly produce N chromosomes as the initial population $p(k)$.

Step2: Evaluate each chromosome in the population and get the fitness value of every solution.

Step3: If stop criterion is satisfied, then output the best solution; otherwise operate steps 4-8.

Step4: Let $m=0$.

Step5: Then, select two chromosomes in the population $p(k)$ using certain selection rules, which are named p_1 and p_2 , respectively.

Step6: Randomly produce a real number $\zeta \in [0,1]$, if $\zeta < p_c$, where p_c is crossover probability, then apply given crossover function on the two selected parent chromosomes. The resulted two chromosomes are selected as two temp chromosomes, namely t_1 and t_2 , respectively. Otherwise, the two parent chromosomes will be selected as the two temp chromosomes t_1 and t_2 , respectively.

Step7: Randomly produce a real number $\delta \in [0,1]$, if $\delta < p_m$, where p_m is mutation probability, then apply given mutation function on t_1 and t_2 respectively. The two resulted chromosomes will be inserted into the new population $p(k+1)$.

Step8: Let $m=m+2$. If $m < N$, then go back to step 5. Otherwise, let $k=k+1$, and then go back to step 2.

4.2. Encoding

The FJSP problems involve two decision stages, i.e., machine assignment stage and operation sequencing stage. Therefore, a solution consists of two parts of vectors, $A_1 = \{A_1(1), A_1(2), \dots, A_1(\kappa)\}$ (machine assignment vector) and operation sequencing vector $A_2 = \{A_2(1), A_2(2), \dots, A_2(\kappa)\}$, where κ equals to the operation number. $A_1(i), 1 \leq i \leq \kappa$ represents the corresponding selected machine for each operation. Fig. 2 shows an example of a machine assignment vector. For example, it can be seen from Fig. 2 that the operation O_{11} is performed on machine M_4 , O_{12} is performed on machine M_3 , and so on. An operation sequencing vector is shown in Fig. 3, which tells us the operation sequence as follows.

$$O_{21} \succ O_{11} \succ O_{31} \succ O_{22} \succ O_{12} \succ O_{23} \succ O_{13} \succ O_{32}$$

position	1	2	3	4	5	6	7	8
operation	O_1	O_1	O_1	O_2	O_2	O_2	O_3	O_3
machine	¹ 4	² 3	³ 2	¹ 1	² 2	³ 1	¹ 2	² 3

Fig. 2. Machine assignment vector example

position	1	2	3	4	5	6	7	8
operation	2	1	3	2	1	2	1	3

Fig. 3. Operation sequencing vector example

4.3. Initialization of Machine assignment component

Following are two approaches for the initialization of machine assignment component:

- Random rule, denoted as **MS_a**. For each operation J_i , a random selected machine from a set of candidate machines, denoted as M_i , will be placed in position i in the machine assignment component.

- Local minimum processing time rule, denoted as **MS_b**. Table 1 gives an example about the steps of this rule, the example data come from [14]. For operations of the same job, finding the minimum processing time, fixing the assignment, and then adding this processing time to every other entry in the same column.

4.4. Crossover operation

Given two parent chromosomes p_1 and p_2 in Fig. 4. The steps of the crossover operator are as follows.

- Step1:** Generate two random numbers r_1 and r_2 , $2 \leq r_1, r_2 \leq (\kappa - 1)$, where κ equals the number of operations.
- Step2:** Select the subsection between r_1 and r_2 of one parent chromosome such as p_2 .
- Step3:** Produce a temporary vector named c_1 by copying the selected subsection into the corresponding position.
- Step4:** Copy the corresponding operation from p_1 into the unfixed position.

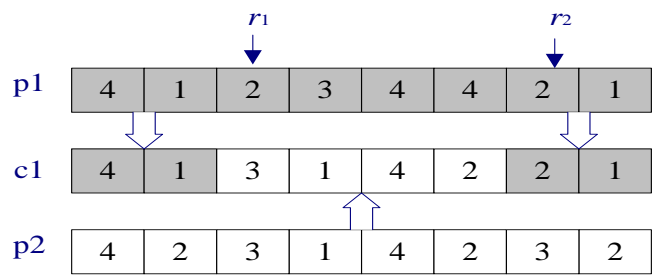


Fig. 4. Crossover operator

4.5. Mutation operation

Mutation operation is very important in GA with aim to produce population diversity. In order to obtain population diversity as well as population convergence, two mutation-operation rules are proposed as follows.

- Random rule denoted as ϖ_1 . (1) randomly select an operation with more than two candidate machines, denoted O_s ; (2) randomly select a machine from $Machines(O_s)$ different with the current machine; (3) replace the current machine by the selected machine at the position O_s .
- Last Processing rule denoted as ϖ_2 . (1) record the last release time for each machine; (2) create a vector M_{lp} including all machines with last release time equals the current makespan; (3) get all public critical operations; (4) for each machine M_{old} in M_{lp} , firstly, find a public critical operation O_s which is processed on M_{old} ; secondly, select a candidate machine for processing O_s which is not in M_{lp} , denoted M_s ; (5) replace the current machine M_{old} by the selected machine M_s at position O_s .

5. Operation sequencing algorithm: variable neighborhood search algorithm

5.1. Initialization of the operation sequencing component

Once the machine assignment component is fixed, we should consider how to sequence the operations on each machine, i.e., to determine the start time of every operation. In this section, we should consider two issues: the operation precedence constraint of the same job and the objective of the problem. In our hybrid algorithm, the operation sequence is obtained through a mix of following two different approaches:

- Random rule, denoted as OS_a . OS_a is the naive and direct approach for sequencing operations. The advantage of this approach is its simplicity. The disadvantage is also obvious too, that is, it can easily produce idle time interval and make the finding solution process more time consuming.
- Most Work Remaining (MWR) denoted as OS_b . This approach selects the operation with the most remaining work for each machine. The operation precedence constraint of the same job must be considered at the same time.

5.2. Public critical block theory

The critical problem of local searching is how to design an effective neighborhood around a given solution. The promising neighborhood is based on the concept of critical path, which was firstly proposed by Adams [15] in solving JSP problems.

The feasible schedules of FJSP problems can be represented with a disjunctive graph $G=(N, A, E)$, where N is the node set, A is the conjunctive arc set, and E is the disjunctive arc set. The number aside the node indicates the processing time of this operation on the assigned machine. Each arc in A represents the operation precedence constraint. The dashed arcs (E) correspond to pairs of operations to be performed on the same machine. For example, given a chromosome $\{1,2,2,3,2,3,3,1 \mid 1,1,4,2,3,3,2,4\}$, Fig. 5 shows the disjunctive graph for a feasible solution of the example chromosome.

If G has more than one critical path, noted $CP_i, 1 < i \leq nc$, where nc represents the number of critical path. Those critical operations belonging to all nc critical paths are called public critical operations. A public critical block is a maximal sequence of adjacent public critical operations processed on the same machine. Fig. 6 shows the Gantt chart for the feasible solution of the above chromosome example. In the example solution, there are six public operations, i.e. $\{O_{11}, O_{12}, O_{21}, O_{31}, O_{32}, O_{22}\}$, whereas there are three public critical blocks, i.e. $\{O_{11}\}, \{O_{12}, O_{21}, O_{31}\}, \{O_{32}, O_{22}\}$.

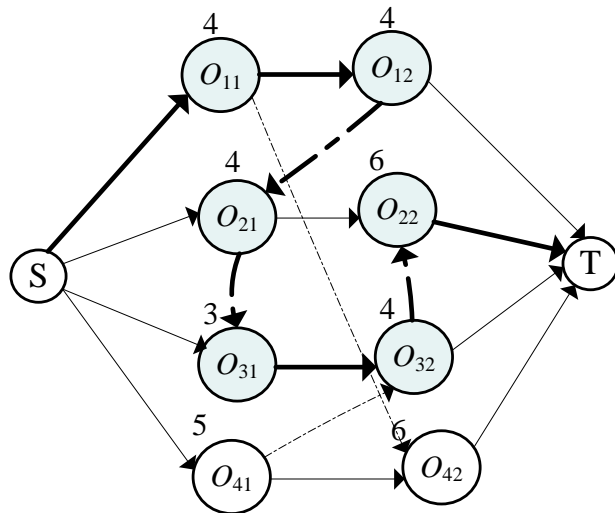


Fig. 5. The disjunctive graph for the example chromosome

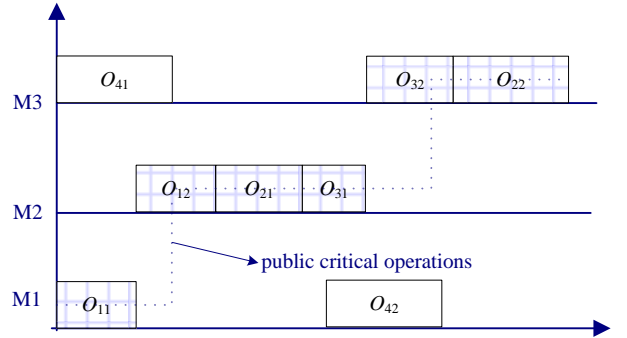


Fig. 6. The Gantt chart for the example chromosome

To develop neighborhood structure based on the public critical block theory, we give some notations as follows.

- JP_i, JS_i, MP_i, MS_i indicates the immediate job predecessor, job successor, machine predecessor and machine successor of the operation J_i , respectively.
- λu : the critical path with operation u in it.
- $O_{\lambda u}$: those operations in the same critical path λ which has operation u in it.
- $PR_{O_{\lambda u}}$: those operations belonging to the operation set $O_{\lambda u}$ and be operated before operation u .
- $L(u, v)$: the length of the longest path from the operation u to v .

Next, we give three theorems about the neighborhood structure based on public critical path theory.

Theorem 1. *If G has more than one critical path, and two operations u and v are critical operations but not public critical operations, then moving u or v cannot yield a better solution.*

Proof. First, if there exists a public critical operation in $PR_{O_{\lambda u}}$, denoted as O_k .

The path subsection from O_k to u is called $sub(u)$. Because the operation u is not a public critical operation, there exists an operation u' with the processing time interval crossover with the processing time interval of u . Therefore, u' is in another critical path but not a public critical operation either. The path subsection from O_k to u' is called $sub(u')$. The movement in $sub(u)$ does not affect the length of $sub(u')$. So, the start time of O_k will not change and the makespan of the solution will not be improved. Second, if there does not exist any operation in $PR_{O_{\lambda u}}$, which means that the public critical operations are all operated after u . The first public critical operation after u denoted as O_k . There exists another critical operation u' which is before O_k and with the processing time crossed over with u . We name the critical path subsection

from u to O_k $\text{sub}(u)$, and the critical path subsection from u' to O_k $\text{sub}(u')$. The movement in $\text{sub}(u)$ also cannot affect the length of the $\text{sub}(u')$, and useless for improvement of the makespan.

Theorem 2. *If two public critical operations u and v to be performed on the same machine, v is the block rear and $L(v, T) \geq L(JS_u, T)$, then inserting u right after v yields an acyclic complete selection.*

This theorem derives the idea that if two following conditions are satisfied: (1) there is no directed path from JS_u to v in G ; (2) the complete time of v is not after the complete time of the immediate job successor of u . Then, inserting u right after v can produce a feasible solution as shown in Fig 7. The proof is analogous to the proof of the theorem 1 in [16].

Theorem 3. *If two public critical operations u and v to be performed on the same machine, u is the block head and $L(0, u) + p_u \geq L(0, JP_v) + p_{JP_v}$, then inserting v right before u yields an acyclic complete selection.*

This theorem derives the idea that if two following conditions are satisfied: (1) there is no directed path from JP_v to u in G ; (2) the complete time of u is not before the complete time of the immediate job predecessor of v . Then, inserting v right before u can produce a feasible solution as shown in Fig 8. The proof is analogous to the proof of the theorem 2 in [16].

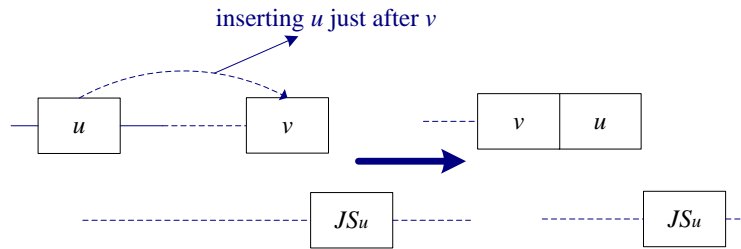


Fig. 7. A chart for inserting the inner operations just after the block rear

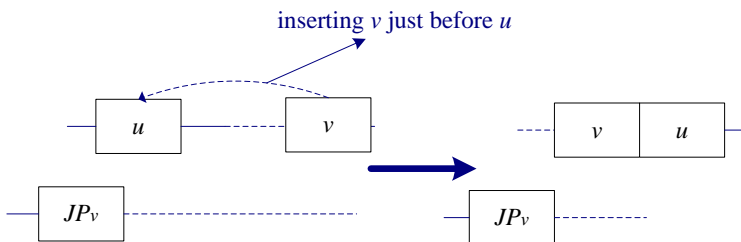


Fig. 8. A chart for inserting the inner operations just before the block head

5.3. Effective neighborhood structure

The makespan of a solution equals the length of its critical path, in other words, the makespan cannot be reduced while maintaining the current critical paths. The right direction of the local search is to identify and break the entire existent critical paths one by one in order to get a better solution.

The first successful critical path neighborhood structure for the classical JSP was introduced by Van Laarhoven et al [17], and is often denoted by N1. The N1 neighborhood is generated by swapping any adjacent pair of critical operations on the same machine. Dell Amico and Trubian [18], Nowicki & Smutnicki [19] and Balas & Vazacopoulos [20] proposed N4, N5 and N6 respectively. The N4 neighborhood is developed by moving an internal operation to the very beginning of its block or the very end of its block. The N5 neighborhood is created by swapping the first two or the last two operations in a block. The N6 neighborhood is produced by moving an operation to the beginning of the block or to the end of the block. In this study, we extend the critical path neighborhood structure for solving the FJSP, and propose some novel neighborhood structures based on the public critical blocks theory.

For discuss conveniently, we give two neighborhood categories as follows:

Definition 1: *Insert neighborhood*

First, randomly select two different positions i and i' in a feasible schedule chromosome, and then delete the i th operation and insert it before or after position i' .

Definition 2: *Swap neighborhood*

First, randomly select two different positions i and i' in a feasible schedule chromosome, and then swap the two operations at the selected position.

Based on the public critical block theory and the block structure listed above, we give an effective local search operator as shown in Fig. 9.

Procedure *effectiveNeighbor()*

Input: a set named pb including all public critical blocks

Output: an optimal neighbor solution

for $i=0$ to $pb.size()$

$pb_i = pb[i]$

 if pb_i contains more than two public critical operations, then

$k \leftarrow$ the number of public critical operations in pb_i

$u \leftarrow pb_i[0]$ //block head

$v \leftarrow pb_i[k-1]$ //block rear

Step1: *Insert structures*

 for $j=0$ to $k-1$

$q \leftarrow pb_i[j]$

Step1.1: if $(L(0, q) + p_q \geq L(0, JP_v) + p_{JP_v})$ then

 Insert v right before q

```

     $pb_t = pb[i]$ 
     $q \leftarrow pb_t[j]$ 
Step1.2: if  $(L(v, T) \geq L(JS_q, T))$  then
        Insert  $q$  right after  $v$ 
         $pb_t = pb[i]$ 
         $q \leftarrow pb_t[j]$ 
Step1.3: if  $(L(0, u) + p_u \geq L(0, JP_q) + p_{JP_q})$  then
        Insert  $q$  right before  $u$ 
         $pb_t = pb[i]$ 
         $q \leftarrow pb_t[j]$ 
Step1.4: if  $(L(q, T) \geq L(JS_u, T))$  then
        Insert  $u$  right after  $q$ 
         $pb_t = pb[i]$ 
    end for
Step2: Swap structures
Step2.1: Swap  $pb_t[0]$  with  $pb_t[1]$ 
         $pb_t = pb[i]$ 
Step2.2: Swap  $pb_t[k-2]$  with  $pb_t[k-1]$ 
    end for
    Evaluate each neighbor solution produced by the above two steps.
    If the new solution is better than the current solution, then replace
    the current solution with the new one.
    Output the current optimal solution.

```

Fig. 9. Pseudo-code of *effectiveNeighbor()*

6. Experimental results

This section describes the computational experiments to evaluate the performance of the proposed algorithm. For this purpose, we made a detail comparison with three existing algorithms, i.e., AL+CGA algorithm [9], PSO+SA algorithm [10] and PSO+TS algorithm [11]. The test samples come from [5]. The dimensions of the problems range from 4 jobs \times 5 machines to 15 jobs \times 10 machines. The current instantiation was implemented in C++ on a Pentium IV 1.8GHz with 512M memory.

6.1. Setting parameters

Each instance can be characterized by the following parameters: number of jobs (n), number of machines (m), and the number of operations (op_num).

Followings are the detail parameters value:

- Population size P_{size} : 1000;
- Maximum number of generations gen_{max} : $n \times m$;
- Maximum number of iteration with no improvement of the best solution during the local search $iter_{max}$: $op_num / 2$;
- Crossover probability for the machine assignment component: 45%;
- Minima mutation probability p_m^{\min} : 40%;
- Maxima mutation probability p_m^{\max} : 95%;
- Current mutation probability at t generation p_m^c :

$$p_m^c = p_m^{\min} + \left(\frac{t}{gen_{max}}\right) \times (p_m^{\max} - p_m^{\min})$$

- Probability for selection between the best chromosome and a random one as a parent chromosome for crossover p_{select} :

$$p_{select} = 0.8 - \left(\frac{t}{gen_{max}}\right) \times (0.8 - 0.2)$$

- Rate of initial assignments with MS_a : 20%;
- Rate of initial assignments with MS_b : 80%;
- Rate of initial assignments with OS_a : 20%;
- Rate of initial assignments with OS_b : 80%;
- Rate of initial assignments with ϖ_1 : 50%;
- Rate of initial assignments with ϖ_2 : 50%;

6.2. Results of the Kacem instances

The test instances come from the five Kacem instances [5] (i.e. problem 4×5 , 8×8 , 10×7 , 10×10 and 15×10). The five tables from Table 2 to Table 5 show the comparison results for the five problems. Some notations are given as follows: The column labeled 'AL + CGA' refers to Kacem's method [9] and the column labeled 'PSO + SA' refers to the algorithm proposed by Xia and Wu [10]. The column labeled 'PSO+TS' shows the results from the hybrid algorithm developed by Zhang et al. [11]. Figs 10 to 14 show the optimal solutions obtained by our approach in the form of Gantt chart. The pair of number (in the form of [job, operation]) inside the blocks is the operation to be processed on the corresponding machine. The two numbers just below the block represent the start time and end time of the operation, respectively.

Problem 4×5

This is an instance of total flexibility, in which 4 jobs with 12 operations are to be performed on 5 machines. The obtained solutions by our hybrid algorithm are characterized by the following values:

Jun-qing Li, Quan-ke Pan, and Sheng-xian Xie

Solution 1: $c_M=12, w_T=32, w_M=8$
 Solution 2: $c_M=11, w_T=32, w_M=10$
 Solution 3: $c_M=11, w_T=34, w_M=9$

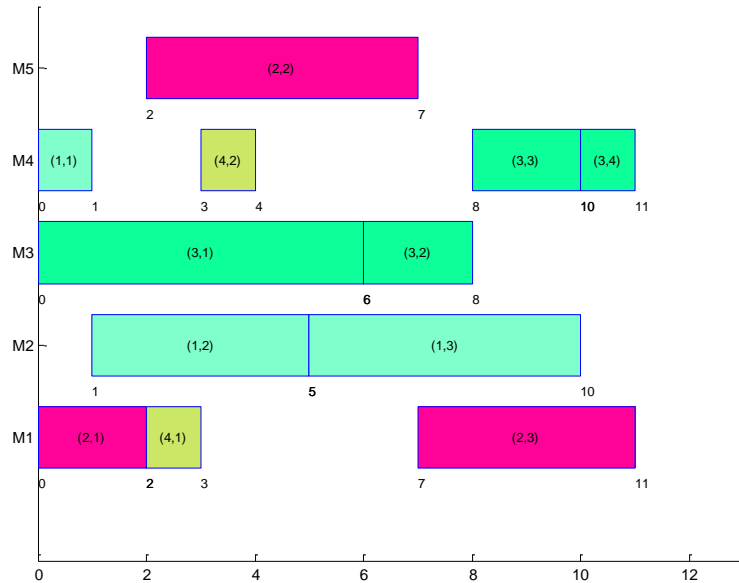


Fig. 10. The obtained optimal solution of instance 1 (4 jobs 12 operations 5 machines: $F_1(c)=11, F_2(c)=34, F_3(c)=9$)

Table 2. Comparison of results on problem 4×5 with 12 operations

	AL+CGA	PSO+TS	VNS+GA		
c_M	16	11	11	11	12
w_T	34	32	32	34	32
w_M	10	10	10	9	8

It can be seen from Table 2 that the VNS+GA algorithm dominates the AL+CGA algorithm in solving the problem 4×5 . Our approach can obtain richer optimal solutions than the PSO+TS algorithm. Fig. 10 shows the obtained Solution 3 in the form of Gantt chart.

Problem 8×8

This is an instance of partial flexibility, in which 8 jobs with 27 operations are to be performed on 8 machines. The obtained solutions by our hybrid algorithm are characterized by the following values:

Solution 1: $c_M=14$, $w_T=77$, $w_M=12$

Solution 2: $c_M=15$, $w_T=75$, $w_M=12$

Solution 3: $c_M=16$, $w_T=73$, $w_M=13$

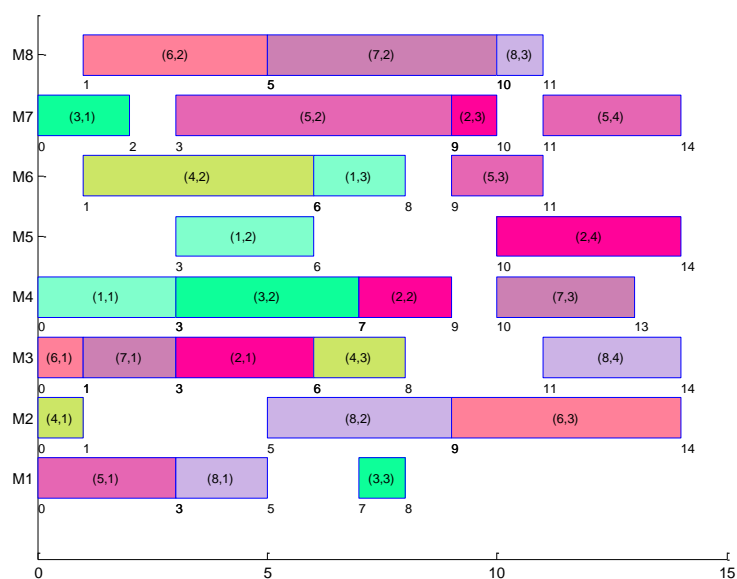


Fig. 11. The obtained optimal solution of instance 2 (8 jobs/27 operations /8 machines: $F_1(c)=14$, $F_2(c)=77$ $F_3(c)=12$)

Table 3. Comparison of results on problem 8×8 with 27 operations

	AL+CGA		PSO+SA		PSO+TS		HTSA		
c_M	15	16	15	16	14	15	1	1	1
							4	6	5
w_T	79	75	75	73	77	75	7	7	7
							7	3	5
w_M	13	13	12	13	12	12	1	1	1
							2	3	2

It can be seen from Table 3 that the VNS+GA algorithm dominates the AL+CGA algorithm. The hybrid algorithm can obtain richer optimal solutions than both the PSO+TS algorithm and the PSO+SA algorithm. Fig. 11 shows the obtained Solution 1 in the form of Gantt chart.

Problem 10×7

This is an instance of total flexibility, in which 10 jobs with 29 operations are to be performed on 7 machines. The obtained solutions by our hybrid algorithm are characterized by the following values:

Solution 1: $c_M=11, w_T=62, w_M=10$

Solution 2: $c_M=11, w_T=61, w_M=11$

Fig. 12 shows the obtained Solution 2 in the form of Gantt chart.

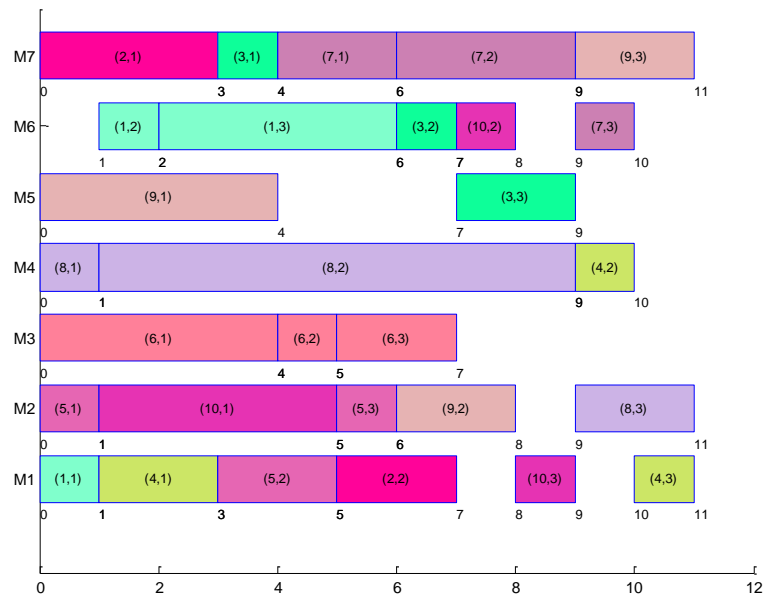


Fig. 12. The obtained optimal solution of instance 3 (10 jobs/29 operations /7 machines: $F_1(c)=11, F_2(c)=61, F_3(c)=11$)

Problem 10×10

This is an instance of total flexibility, in which 10 jobs with 30 operations are to be performed on 10 machines. The obtained solutions by our hybrid algorithm are characterized by the following values:

Solution 1: $c_M=7, w_T=43, w_M=5$

Solution 2: $c_M=7, w_T=42, w_M=6$

Solution 3: $c_M=8$, $w_T=42$, $w_M=5$

It can be seen from Table 4 that the VNS+GA algorithm dominates the above three algorithms and can also obtain richer optimal solutions. Fig. 13 shows the obtained Solution 2 in the form of Gantt chart.

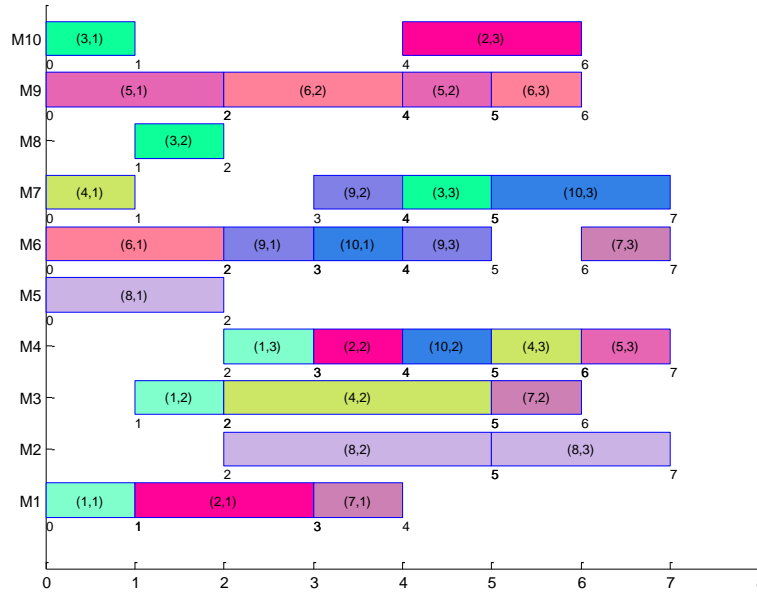


Fig. 13. The obtained optimal solution of instance 4 (10 jobs/30 operations/10 machines: $F_1(c)=7$, $F_2(c)=42$ $F_3(c)=6$)

Table 4. Comparison of results on problem 10×10 with 30 operations

	AL+CGA	PSO+SA	PSO+TS	HTSA
c_M	7	7	7	7
w_T	45	44	43	42
w_M	5	6	6	5

Problem 15×10

This is an instance of total flexibility, in which 15 jobs with 56 operations are to be performed on 10 machines. The obtained solutions by our hybrid algorithm

Jun-qing Li, Quan-ke Pan, and Sheng-xian Xie

are characterized by the following values:

Solution 1: $c_M=11$, $w_T=92$, $w_M=11$

Solution 2: $c_M=12$, $w_T=91$, $w_M=11$

It can be seen from Table 5 that the VNS+GA algorithm dominates both the AL+CGA and the PSO+TS algorithms and can also obtain richer optimal solutions than the PSO+SA algorithm. Fig. 14 shows the obtained Solution 2 in the form of Gantt chart.

Table 5. Comparison of results on problem 15×10 with 56 operations

	AL+CGA		PSO+SA	PSO+TS	HTSA	
c_M	23	24	12	11	11	12
w_T	95	91	91	93	92	91
w_M	11	11	11	11	11	11

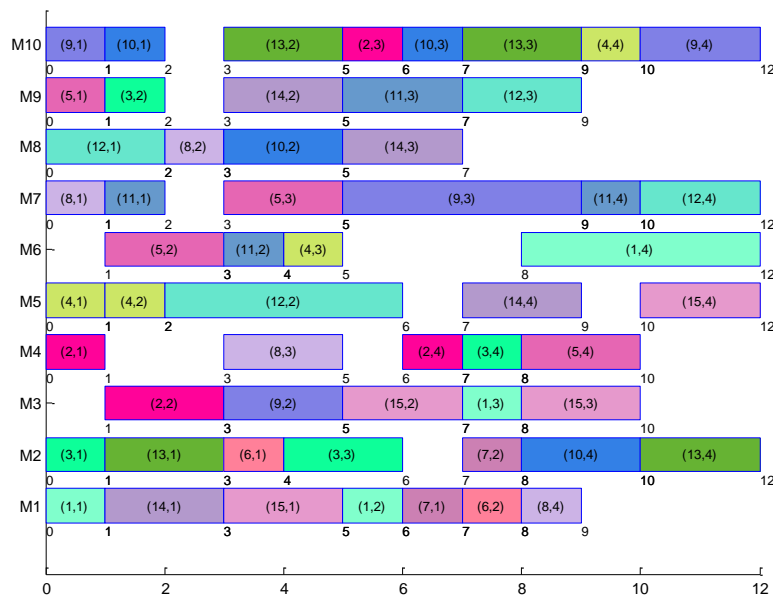


Fig. 14. The obtained optimal solution of instance 5 (15 jobs/56 operations/10 machines: $F_1(c)=12$, $F_2(c)=91$, $F_3(c)=11$)

From the above comparison with other three existing algorithm for solving the five Kacem instances, we can conclude that our algorithm either obtain superior solutions or can obtain richer non-dominate solutions than the other

approaches, especially for solving large scale instances.

7. Conclusions

Flexible job shop scheduling problem is very important in both fields of combinatorial optimization and engineering management. Most literature focus on proposing hybrid algorithms for solving mono-objective FJSPs. The research on the multi-objective FJSP is much less than the mono-objective FJSP. Kacem proposed a hybrid algorithm named AL+CGA combining GA and approach of localization. Xia and Wu presented a hybrid algorithm named PSO+SA which making use of PSO for solving the assignment sub problem and SA for solving the routing sub problem. Zhang et al. developed a hybrid algorithm named PSO+TS for the multi-objective FJSPs with the same three objectives. In this paper, we introduce a novel algorithm named VNS+GA combining VNS and GA for solving the multi-objective FJSPs to minimize the makespan, the total workload, and the workload of the busiest machine. There are mainly three contributions in the hybrid algorithm. Firstly, a mix of two machine assignment rules and two operation sequencing rules are developed in the initialization stage to produce enough high quality initial solutions. Secondly, an adaptive mutation rules are introduced for considering both population diversity and convergence speed in perturbation in the machine assignment component. Thirdly, a speed-up variable neighbor search operator based on public critical block theory was investigated. The new local searching approach makes the search space dwindled deeply and produces high quality neighbor solutions in very short time. Experimental results compared with the three existing algorithms (i.e., AL+CGA algorithm, PSO+SA algorithm and PSO+TS algorithm) show that our hybrid algorithm can either obtain superior solutions or obtain richer non-dominated solutions than the other algorithms, especially for larger scale instances.

The future work is to extend the initial solution rules and the public critical block method for solving other combinatorial problems. In addition, we will develop other heuristic algorithms with the public critical block neighborhood structure for solving the multi-objective FJSPs.

Acknowledgments. This research is partially supported by National Science Foundation of China under Grants 60874075, 70871065, Open Research Foundation from State Key Laboratory of Digital Manufacturing Equipment and Technology (Huazhong University of Science and Technology), Science Research and Development of Provincial Department of Public Education of Shandong under Grant J08LJ20, J09LG29, J08LJ59, and Soft Science Foundation of Shandong under Grant 2009RKB125.

References

1. Jain A.S., Meeran S.: Deterministic job-shop scheduling: Past, present and future, *European Journal of Operation Research*, Vol. 113, No. 2, 390-434. (1998)
2. Garey M.R., Johnson D.S., Sethi R.: The Complexity of Flowshop and Job shop Scheduling, *Mathematics of Operations Research*, Vol. 1, No. 2, 117-129. (1976)
3. Bruker P., Schlie R.: Job-shop scheduling with multi-purpose machines, *Computing*, Vol. 45, No. 4, 369-375. (1990)
4. Brandimarte P.: Routing and scheduling in a flexible job shop by tabu search, *Annals of Operations Research*, Vol. 22, 158-183. (1993)
5. Kacem I., Hammadi S., Borne P.: Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic, *Mathematics and Computers in Simulation*, Vol. 60, No.3, 245-276. (2002a)
6. Gao L., Peng C.Y., Zhou C., Li P.G.: Solving flexible job shop scheduling problem using general particle swarm optimization, In *Proceedings of the 36th CIE Conference on Computers & Industrial Engineering*, Kaohsiung, Taiwan, 3018-3027. (2006)
7. Saidi-mehrabad M., Fattahi P.: Flexible job shop scheduling with tabu search algorithms, *International Journal of Advanced Manufacturing Technology*, Vol. 32, No. 5-6, 563-570. (2007)
8. Li J.Q., Pan Q.K., Xie S.X., Li H., Jia B.X., Zhao C.S.: An effective hybrid particle swarm optimization algorithm for flexible job-shop scheduling problem. *MASJUM Journal of Computing*, Vol. 1, No. 1, 69-74. (2009)
9. Kacem I., Hammadi S., Borne P.: Approach by localization and multi-objective evolutionary optimization for flexible job-shop scheduling problems, *IEEE Transactions on Systems, Man and Cybernetics, Part C*, Vol. 32, No. 1, 408-419. (2002b)
10. Xia W.J., Wu Z.M.: An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems, *Computers and Industrial Engineering*, Vol. 48, No. 2, 409-425. (2005)
11. Zhang G.H., Shao X.Y., Li P.G., Gao L.: An effective hybrid swarm optimization algorithm for multi-objective flexible job-shop scheduling problem, *Computers and Industrial Engineering*, Vol. 56, No. 4, 1309-1318. (2009)
12. Goldberg D.E., Holland J.H.: *Genetic Algorithms and Machine Learning*, Machine Learning, Vol. 3, No. 2-3, 95-99. (1988)
13. Wang L.: *Intelligent Optimization Algorithms with Applications*. Tsinghua University Press, Beijing, 2001.
14. Pezzella F., Morganti G., Ciaschetti G.: A genetic algorithm for the Flexible Job-shop Scheduling Problem. *Computers & Operations Research*, Vol. 35, No. 10, 3202-3212. (2008)
15. Adams J., Balas E., Zawack D.: The shifting bottleneck procedure for Job Shop Scheduling. *Management Science*, Vol. 34, No. 3, 391-401. (1988)
16. Zhang C.Y., Li P.G., Guan Z.L., Rao Y.Q.: A tabu search algorithm with a new neighborhood structure for the job shop scheduling problem. *Computers & Operations Research*, Vol. 34, No. 11, 3229-3242. (2007)
17. Van Laarhoven P.J.M., Aarts E.H.L., Lenstra J.K.: Job shop scheduling by simulated annealing. *Operations Research*, Vol. 40, No. 1, 113-125. (1992)
18. Dell'Amico A.M., Trubian A.M.: Applying Tabu Search to the Job-shop Scheduling Problem. *Annals of Operation Research*, Vol. 41, No. 3, 231-252. (1993)
19. Nowicki E., Smutnicki C.: A fast taboo search algorithm for the job-shop problem. *Management Science*, Vol. 42, No. 6, 797-813. (1996)

20. Balas E., Vazacopoulos A.: Guided Local Search with Shifting Bottleneck for Job Shop Scheduling. *Management Science*, Vol. 44, No. 2, 262-275. (1998)

Jun-qing Li was born in Liaocheng, China, in 1976. He received the Master degree of computer science and technology in 2004 from Shandong Economic University, Shandong, China. He is currently an associate professor in Department of Computer Science and Technology, Liaocheng University, Liaocheng, China. He is now a member of IEEE and CCF. His research is related to the evolutionary optimization methods for discrete events systems, job shop systems, operational research, computer network, peer-to-peer systems and network security. He has (co-)authored around 50 research papers published. He has been serving on editorial boards or reviewers of two international journals. He has been the member of program committees of many international conferences.

Quan-ke Pan was born in Liaocheng, China, in 1971. He received the PhD of manufacturing and automation in 2003 from Nanjing University of Aeronautics and Astronautics, Nanjing, China. His research interests include computational intelligent, operational research and swarm optimization algorithms. He is currently a professor in Department of Computer Science and Technology, Liaocheng University, Liaocheng, China. He has (co-)authored around 120 research papers published. He has been serving on editorial boards of several international journals and has edited special issues in international journals. He has been member of program committees of many international conferences.

Sheng-xian Xie was born in Liaocheng, China, in 1957. He is currently a professor in Department of Computer Science and Technology, Liaocheng University, Liaocheng, China, and he is the chair of the department. His research interests include network security, access control and peer-to-peer systems.

Received: August 08, 2009; Accepted: February 04, 2010.

Table 1 Local minimum processing time rule (MS_b)

	M_1	M_2	M_3	M_4	M_1	M_2	M_3	M_4	M_1	M_2	M_3	M_4	M_1	M_2	M_3	M_4
O_{11}	7	6	4	5	7	6	4	5	7	6	4	5	7	6	4	5
O_{12}	4	8	5	6	4	8	9	6	4	8	5	6	4	8	5	6
O_{13}	9	5	4	7	9	5	8	7	13	5	4	7	9	5	4	7
O_{21}	2	5	1	3	2	5	5	3	6	5	1	3	2	5	1	3
O_{22}	4	6	8	4	4	6	12	4	8	6	8	4	4	6	8	4
O_{23}	9	7	2	2	9	7	6	2	13	7	2	2	9	7	2	2
O_{31}	8	6	3	5	8	6	7	5	12	6	3	5	8	6	3	5
O_{32}	3	5	8	3	3	5	12	3	7	5	8	3	3	5	8	3

Automation of the Moving Objects Movement Prediction Process Independent of the Application Area

Ivana Nižetić¹ and Krešimir Fertalj¹

¹Faculty of Electrical Engineering and Computing, University of Zagreb,
10000 Zagreb, Croatia
{ivana.nizetic, kresimir.fertalj}@fer.hr

Abstract. Whereas research on moving objects is involved in a variety of different application areas, models and methods for movement prediction are often tailored to the specific type of moving objects. However, in most cases, prediction models are taking only historical location in consideration, while characteristics specific to certain type of moving objects are ignored. In this paper, we presented a conceptual model for movement prediction independent on an application area and data model of moving objects considering various object's characteristics. Related work is critically evaluated, addressing advantages, possible problems and places for improvement. Generic model is proposed, based on an idea to encompass missing pieces in related work and to make the model as general as possible. Prediction process is illustrated on three case studies: prediction of the future location of vehicles, people and wild animals, in order to show their differences and to show how the process can be applied to all of them.

Keywords. moving object, spatiotemporal data, movement prediction, generic model for movement prediction

1. Introduction

Recent wide-ranging usage of Global Positioning System (GPS) devices and wireless communication devices, together with enhancements of supportive technology, induced the expansion of the research on moving objects. By modelling and analyzing moving objects data, we learn about the moving objects behaviour and even become able to predict their future locations [16].

All moving objects in the real world comprise time and space attributes simultaneously, with characteristic of having changeable location or shape through the time [23]. In many applications, knowing moving objects locations in advance can be substantial. Discovery of behavioural patterns and prediction of future movement can greatly influence different fields. Examples are the analysis of the wild animals' movement in order to predict their migrations and predator-prey behaviour [7] [24], monitoring and analysis

of vehicle movement in order to predict driver's intentions [15] or traffic congestions [13], mobile user movement and access point availability prediction in order to assure the requested level of quality of service in wireless networks [6]. There are also situations in which the exact position of a moving object cannot be determined, for example when the moving object enters a shadow area of GPS and the estimation of future location by tracking the previous ones that were provided in visible regions becomes necessary [24].

As technology advances, more available data about moving objects is encountered, thus increasing ability to mine spatiotemporal data [5]. In [9], researchers found out that human movement shows a high degree of temporal and spatial regularity. Froehlich and Krumm [8] show that a large portion of a typical driver's routes are repeated. It is thus reasonable to expect the extraction of the regularity, its description and usage in prediction of future movement.

Different data mining techniques can be used to extract behavioural patterns from moving objects data [5] [11], several prediction techniques can be used to model and predict moving object's future location, such as neural networks, Markov models, and specific types of dynamic Bayesian networks like hidden Markov models or Kalman filter [3] [6] [7] [14] [20]. Still, all aforementioned methods have to be adapted to deal with moving object's data. Furthermore, the most of them focus on managing historical and current movement data of moving objects and only a few of them have been proposed to deal with moving object's future movement prediction [21].

Paper is organized as follows: In the next chapter related work is critically evaluated, addressing advantages, possible problems and limitations. In chapter 3 according to problems addressed in chapter 2, objectives and places for improvement are suggested. Schema of prediction process is presented, based on an idea to encompass missing pieces in related work and to make the prediction model as general as possible. Conceptual data model independent of application area and moving object type is proposed. In chapter 4 three case studies illustrates prediction process, in order to show differences between different moving object types and to show how the process can be applied to all of them. Conclusion is made in chapter 5 with an announcement of the future work.

2. Related Work

Considerable research on moving objects has been done in various application areas so far. J.Froehlich and J.Krumm [8] predict the route of a vehicle. They claim prediction is the missing piece in several proposed ideas for intelligent vehicles. Prediction is useful for giving warnings about upcoming traffic hazards or giving information about upcoming points of interest, including advertising, to the driver. C.S. Jensen et al. [13] track a population of vehicles. They list a range of applications that may utilize this

kind of tracking, such as mobile services in relation to traffic monitoring, collective transport, and the management of fleets, (e.g. of emergency vehicles, police cars, delivery trucks, and vehicles carrying dangerous or valuable cargo). In [9] authors present methods for person motion prediction in order to enable a mobile robot to keep track of people in its environment and to improve its behaviour. They state that robots operating in populated environments can improve their service if they react appropriately to the activities of the people in their surrounding and in the same time not interfering with them.

J. Petzold [20] presents context prediction evaluated by the people walking through the office building, recording their movements on PDA.

G. Yavas et al. [27] consider mobility prediction a hot topic in management research field. J.M. François et al. [6] claim that prediction can be particularly useful to assure the given level of quality of service despite the typically large jitter and error rates in wireless networks.

D.W.Sims et al. [25] analyse large amount of data representing displacements of diverse marine predators – sharks, bony fishes, sea turtles and penguins, while A. Franke et al. [7] encapsulate movement and kill-site behaviour in three wolf packs.

Furthermore, J. Krumm and E. Horvitz [15] mention usefulness of next location prediction in ubiquitous computing research. As they claim, beyond current object location, location-based services can be developed by taking into account object future locations, providing more efficient service.

Some work has been done concerning moving objects in general [5][26]. It is also worth to mention a recent database research on moving objects [4][23], where important issues are development of spatiotemporal databases to support moving objects, efficient indexing techniques and efficient extraction of spatiotemporal data.

Shortcoming of related work is in most cases considering only location and time as attributes of moving object's movement (Recently, some ideas of enriching object's movement data with geographical and semantic information is proposed [10][1][2]). Furthermore, the prediction (in the way it is handled in previous work) assumes dispose of an amount of training data concerning observed area, i.e. area in which prediction has to be made. It means that we are not able to predict the object movement in areas where the object has never been before. In Table 1 and Table 2 comparison of some aforementioned works is given.

3. Automation of the Modelling Process and Prediction

Based on an idea to encompass missing pieces in related work and to make the movement modelling and prediction process as general as possible, we will present generic model and propose the conceptual data model independent of application area and independent of moving object type.

Table 1. Comparison of related work (1/2)

Moving objects	Application area	Method	Technology	Advantages	Limitations
Mobile users (walking or driving)	Predicting the next router that a host will be linked to in order to assure high quality of service level [0]	Hidden Markov model (HMM)	GPS or antenna	Non-similar patterns distinction; Model states are not predefined (they depend on the data)	Number of models grows as the number of neighbour access points squared; Impossible to predict future locations in new areas
Vehicles	Predicting end-to-end route of a vehicle [0]	Clustering, similarity measuring	GPS	Predicting driver's entire route, not only destination; Independence of map matching	Assuming to have been given driver's historical data; Impossibility to predict future locations in new areas
Vehicles	Tracking populations of vehicles and predicting their movement [0]	Tracking techniques (point-, vector- and segment-based)	GPS (INFATI ¹ data)	The tracking component can be used in a variety of applications; Combining different prediction techniques into a single robust tracking technique	Very simple short-term prediction algorithm
Vehicles	Predicting a driver's near-term future path for giving the warnings about upcoming road situations [0]	Markov model	GPS	Simple and accurate algorithm; Prediction is based on only a single previous observation of a few road segments	Short-term route predictions; Impossible to predict future locations in new areas

¹ INFATI is a Danish acronym for "INtelligent FArtIilpasning" (Intelligent Speed Adaptation). INFATI data is data about drivers' locations collected during the INFATI project. More details in [12].

Automation of the Moving Objects Movement Prediction Process Independent of the Application Area

Table 2. Comparison of related work (2/2)

Moving objects	Application area	Method	Technology	Advantages	Limitations
People	Predicting the indoor movement, i.e. the next room a user will enter in office building [0]	Artificial Neural Networks	Manual location recording on PDAs	Comparison of several prediction techniques; Predicting not only future locations, but the duration of stay at and the locations and the time of the location change	Manual location recording; Impossible to predict future locations in new areas; Applicable solely on indoor places or other places with the clear space division
General	Any, but tested on data about vehicles movement data [0]	Clustering, similarity measuring	GPS (INFATI data)	Using clusters instead of raw data (which is cheaper to mine); Exceptional points removal	Assuming periodic movement (the same for each object's data); Impossibility to predict future locations in new areas
People	Improving robot's behaviour in populated environments by keeping track of people and predicting their movement [0]	HMM, Clustering	Laser-range finders	Long-term prediction; Maintaining and estimating positions of multiple persons; Complete solution (from data collection to prediction)	Clustering depends on trajectory length; Impossible to predict future locations in new areas
Animals	Prediction of wolf kill-sites for gaining insight into predator-prey dynamics [0]	HMM	GPS (collars and manual recorded from aircraft)	Examining interaction between predator and prey; Exact location independence (measuring distance, angle and travel rate instead of solely coordinates)	Predicting only kill-sites; not future locations; Manual prey location recording

3.1. Objectives

We believe that considering geospatial conditions (e.g. type of a habitat, climate, and various object vicinity) that pertain to the location and temporal conditions (e.g. season, time of the day) leads to a more accurate description of moving objects behaviour and prediction of their future movement [18]. The most of additional attributes can be extracted from coordinates and time attributes. Knowing space coordinates, attributes such as vegetation, altitude or type of road can be scanned from geospatial maps [16]. Similarly, knowing time attribute, attributes such as season, temperature or rainfall can be get from historical data collected by weather stations.

We propose adaption of existing methods in order to predict movement in the new areas. Problem could be modelled considering the analogous areas as the same, thus allowing the data collected at one area to be used as training data for other areas.

Another possible improvement could be in supplementing prediction process with expert knowledge about particular moving object characteristics and behaviour.

3.2. Generic Model for Moving objects Movement Prediction

The schema of prediction process that consists of creating the model, learning and finally predicting the movement is given in Fig. 1. "The model" here refers to the prediction model constructed using different prediction techniques and adaptable to the given data about moving objects (e.g. historical data, expert knowledge data, geographical data).

In the first phase, with a help from experts in an application domain, the problem is modelled (for example space partition and/or expert movement rules are defined). The gathered historical data is prepared (using for example clustering techniques, with additional removing of outliers, errors in data or random movement). They are eventually used to define model structure as well (using for example clustering methods to extract interesting places and then to define the parts of model). The historical data is (iteratively) used to estimate the model parameters. Further on, additional information such as terrain or climate characteristics is included in modelling and/or data preparation. The result of the first phase is a model, automatically adapted to the application domain, i.e. to the given data and knowledge representing certain class of moving objects.

In the second phase, the prediction is done based on the given test data by using prepared model and appropriate algorithms. Test data is the part of the historical data, not included in building the model, left to measure and improve model accuracy. As well as historical data, it is prepared to fit the model input. Predicted locations are compared to actual locations stored in test data and model accuracy and statistics is calculated. Test data can be used to tune model parameters.

In the third phase, the prediction is done for given new data using the prepared model and appropriate algorithms. Information about accuracy (calculated in the previous phase) is provided to the user. New data can be used to tune model parameters. Presence of particular process elements differs for various applications. For example, expert knowledge or additional information could be included just to formulate problem or it could be used in an iterative process of parameters estimation. They don't have to be included at all, but that will lead to the loss of model completeness and model accuracy.

The main idea is to automate this process, which encompasses automated fetch of additional information using different services, discovery of expert knowledge from additional data sets (e.g. rule mining), embedding expert knowledge without model perturbation and the independence of application area.

3.3. Data Model of Moving Objects

In order to encompass all additional characteristics of moving objects, we present data model of moving objects in general (Fig. 2 – Rectangle represents entity; diamond shape relationship; symbol 1-N represents one-to-many and N-M many-to-many relationship). All moving objects share mentioned characteristics, although some of them are more important to some types of moving objects, especially in specific prediction process.

The central data in the data model is moving object – representing unique instance of certain class (moving object type) of moving objects. Both moving object and moving object type have properties characteristic to the object/type. Moving object can be seen during observation (that could mean that object is seen, heard, object's trace is seen or location of the object is collected from GPS device). Both time and location (mostly coordinates) are recorded. Location at certain time has certain weather conditions described by weather type entity. Further, locations are parts of areal (region) which can have various characteristics and semantics. Areal can represent the geographical region of for example forests or cities, humid zone etc. The main groups of areal characteristics are: vegetation, climate and urban environment. Some areal can have special meaning to the moving object, such as "home" or "work" regions.

Regardless to which object is of our interest, it can be modelled by data model presented in Fig. 2, with the difference just in object parameters.

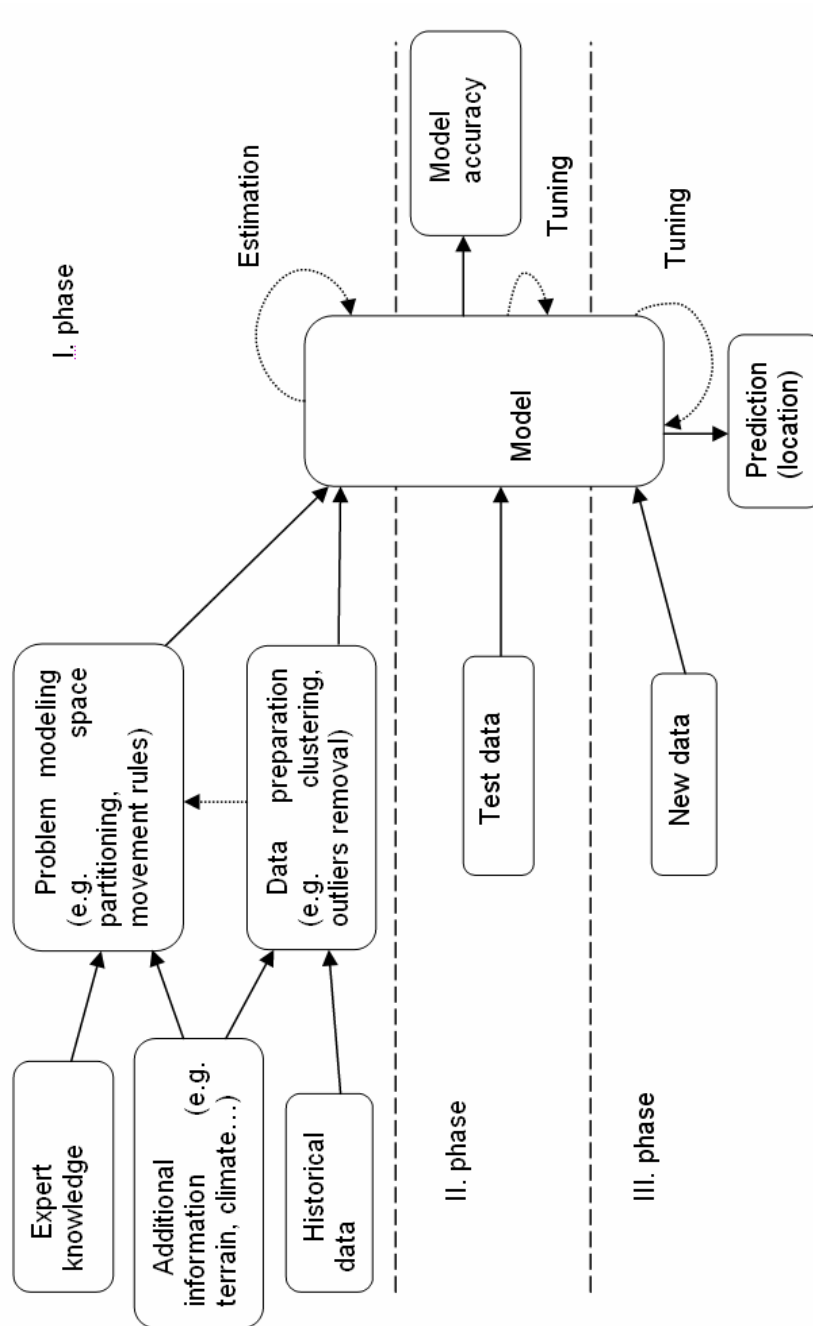


Fig. 1. A schema of moving object's movement prediction process

Automation of the Moving Objects Movement Prediction Process Independent of the Application Area

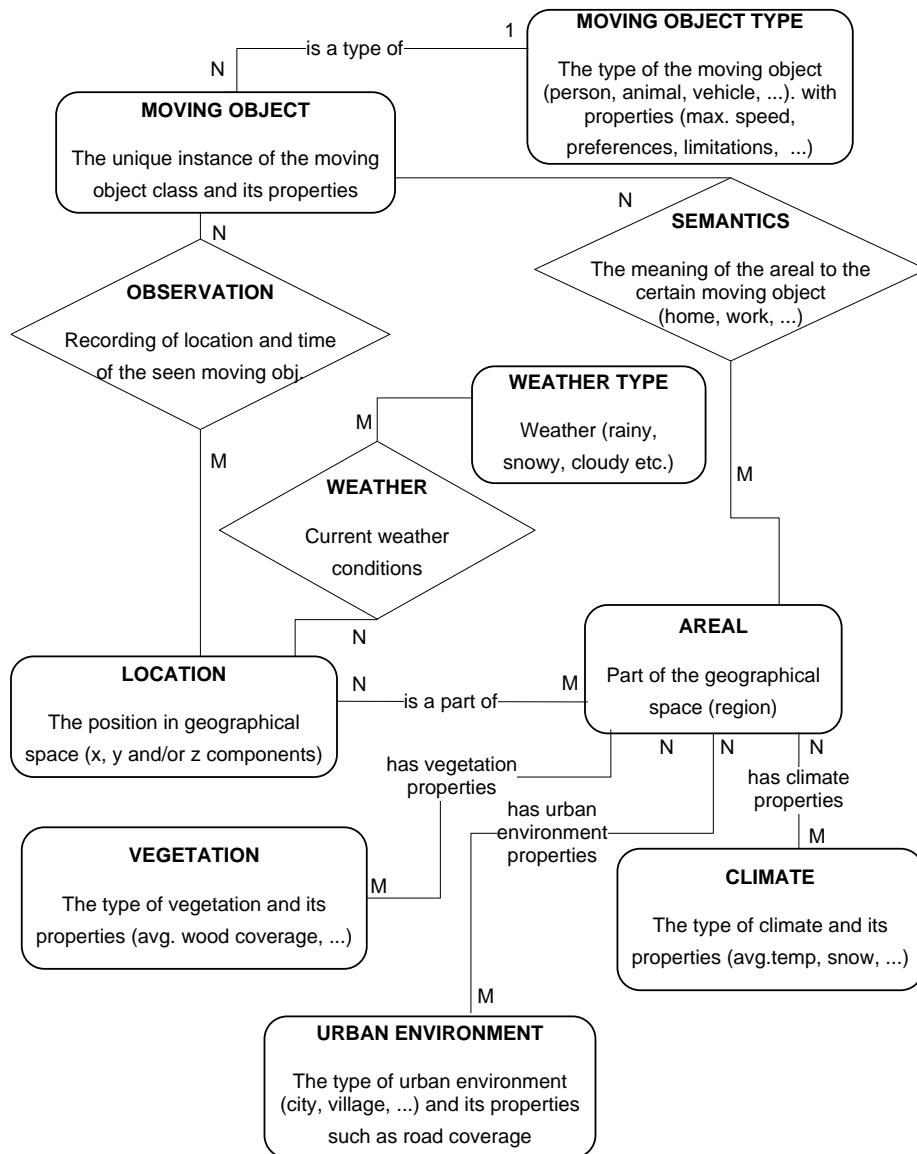


Fig. 2. Data model of moving objects

4. Case Studies

To illustrate the prediction process presented in Fig. 1, we have chosen three case studies: prediction of the future location of vehicles, people and wild animals. We have chosen aforementioned types of moving objects in order to show their differences and to show how the process can be applied to all of them.

We will present one possible method and algorithm selection scenario for each of case studies and shortly discuss about other possibilities.

4.1. Case Study 1 – Vehicles

The first example is prediction of the next location of the vehicle. The main characteristic of vehicles is that they move within the road network. The location of the vehicle equipped with GPS device can be received almost continually (for example, every second).

Since drivers (or people in general) have habits and tend to follow the similar routes [8], it is reasonable to expect some regularity in their movement. Parts of driver's trajectories are overlapping and we expect prediction according to historical locations to be better than just random guessing.

Prediction process is shown on the example of Markov model [22].

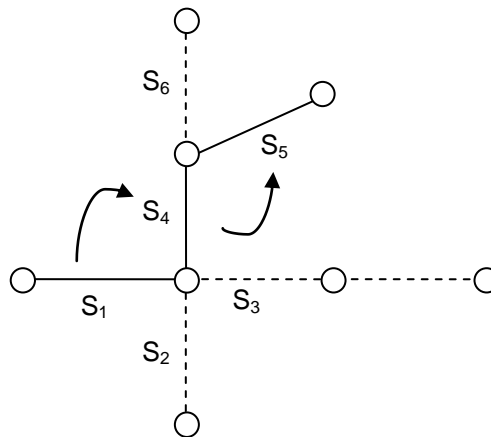


Fig. 3. Vehicle is moving within the road network – all the lines represent road segments, filled lines (states of the model S1, S4 and S5) represent the actual movement of the vehicle

In the first phase of prediction process, according to the knowledge about the road network (additional information), road segments are chosen as Markov model states (S_1, S_2, \dots – see Fig. 3). Information about driver's

previous movement (historical information) is used to define transition probabilities of the model. Similar approach is presented in [14].

Additional information such as the characteristics of road segments, knowledge about the position of driver's home, work etc. (additional information) and information about characteristics about the vehicle such as maximal and average speed (expert knowledge) could give the better insight of driver's intentions and thus could lead to more accurate prediction.

In the second phase, depending on the chosen model, accuracy of predicted location is differently defined. Accuracy of the model in this case should be compared to the accuracy of random guessing.

Another possible modelling method is presented in [5]. However, there is no additional information or expert knowledge used in both [14] and [5].

4.2. Case Study 2 – People

Unlike the vehicles, people are moving more freely in the space. Although they are moving along the street, they can cross the square, park etc. Position of the person is provided by mobile phone equipped with GPS or more often by the identification of close access points and calculation of person's approximate position. That complicates identification of the road that person is moving along, and model presented in 4.1 is not usable.

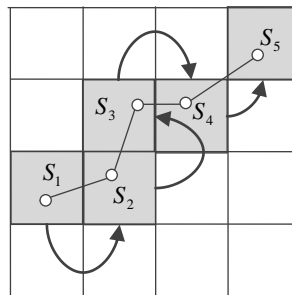


Fig. 4. Person is moving in the space – lines represent actual movement of the person (states of the model S_1 , S_2 , S_3 , S_4 and S_5)

Prediction process is shown on the example of Markov model, but unlike in 4.1, space is partitioned in equally sized cells, which become states of the model (see Fig. 4). Cell size is chosen according to the characteristics of moving object (such as maximum speed) in order to make transitions possible only between neighbouring cells. Similar approach is presented in [11] [19].

States of Markov model could be enriched with additional information such as information about the type of the terrain that pertains to the state of the model. Based on the knowledge about the person's movement preferences (for example it is more likely that person crosses a square than a loan), this could be used to identify transition probabilities.

Another possible modelling method is presented in [27]. Methods like the one presented in [5] can also be used.

4.3. Case Study 3 – Wild Animals

Wild animals are moving even more freely in the space. Another problem is that locations of moving animal are received in sparse intervals (for example every 6 hours). Due to mentioned problems, neither model proposed in 4.1 nor 4.2 is appropriate for this kind of moving objects.

Prediction process is shown again on the example of Markov model.

In the first phase of prediction process, based on historical data and/or expert knowledge, states of Markov model are defined, using for example clustering algorithms to discover frequent regions (see Fig. 5). Interesting regions could also be chosen according to spatial characteristics (additional information), not only based on historical data.

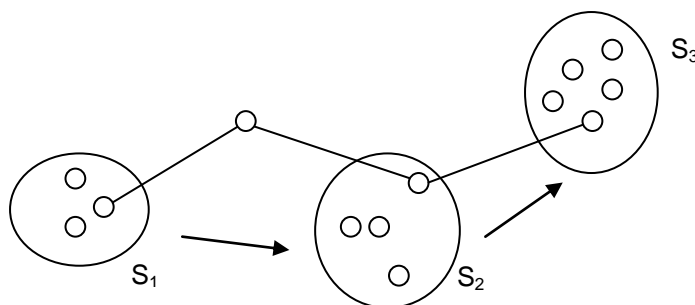


Fig. 5. Wild animal is moving in the space – lines represent actual movement of the animal (states of the model S₁, S₂ and S₃ – chosen as frequent regions)

Additional information (geographical) could be used to get better insight into the meaning of frequent regions and to calculate transition probabilities as well. Further, patterns and association rules could be extracted according to historical or additional information.

As far as we know, such approach has not been described in literature regarding to this type of moving objects. Idea for this approach is based on [1].

4.4. Discussion

Different types of moving objects have different characteristics, but they all move in space with spatial characteristics and their movement has certain characteristics presented in Fig. 2. We suggested possible modelling methods for movement prediction for three different types of moving objects,

based on their characteristics. Despite of their differences, movement prediction process is satisfying presented model (Fig. 1).

For example, given the historical locations of vehicle, underlying road network information, locations of driver's home and work, expert knowledge about behaviour of people (such as: "people tend to follow the same route every day, rather than the shortest one"), model will adapt to the model presented in 4.1. If the road network information is not available, model will adapt to the solution similar to the one presented in 4.2. Furthermore, if the locations of home and work are not provided, model could try to identify them based on historical locations.

To conclude, what we claim is that given the all available information about the observed moving object, generic model presented in Fig. 1 could adapt to the given information and according to them provide the most appropriate prediction model. Naturally, if less information is given, model will be more general (probably less accurate, too). If more information is given, model is more specific, adapted to the specific kind of data and more accurate.

5. Conclusion and Future Work

Knowing moving objects behaviour and predicting moving objects future locations can be very useful in many application areas. Beside analysis which we could perform to extract some regularity in the movements and get better insight into moving objects' behaviour, the great issue is to predict moving object's next position.

Short overview and comparison of related work is given, in order to encompass main characteristics of moving objects and to address present problems. Conceptual model for moving objects movement prediction is presented. The directions for future work are to include the knowledge about the type of a habitat that pertains to the location and the knowledge about moving objects' behaviour. The main goal of our future work is to suggest mechanisms to incorporate those elements to a proposed generic prediction model.

References

1. Alvares L. O., Borgony V., Kuijpers B., de Macedo J. A. F., Moelans B. and Vaisman A.: A Model for Enriching Trajectories with Semantic Geographical Information. In Proceedings of the 15th ACM Symposium on Advances in Geographic Information Systems, Seattle, Washington, Article No. 22, (2007)
2. Andrienko N., Andrienko G., Pelekis N. and Spaccapietra S.: Basic Concepts of Movement Data (book chapter in Mobility, Data Mining and Privacy), Springer Berlin Heidelberg, (2008)

3. Bennewitz M., Burgard W., Cielniak G., Thrun S.: Learning Motion Patterns of People for Compliant Robot Motion. *International Journal of Robotics Research*, Vol. 24, No. 1, 31-48. (2005)
4. Brakatsoulas S., Pfoser D. and Tryfona N.: Modeling, Storing and Mining Moving Object Databases. In *Proceedings of International Database Engineering and Applications Symposium (IDEAS'04)*, 68-77. (2004)
5. Elnekave S., Last M. and Maimon O.: Predicting Future Locations Using Clusters' Centroids. In *Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems*, (2007)
6. François J.-M., Leduc G. and Martin S.: Learning movement patterns in mobile networks: a generic approach. In *Proceedings of European Wireless 2004*, 128-134. (2004)
7. Franke A., Caelli T., Kuzyk G. and Hudson R. J.: Prediction of wolf (*Canis lupus*) kill-sites using hidden Markov models. *Ecological Modelling*. Vol. 197, Issues 1-2, 237-246. (2006)
8. Froehlich J. and Krumm J.: Route Prediction from Trip Observations. *Society of Automotive Engineers (SAE) 2008 World Congress*, Detroit, MI, USA, (2008)
9. Gonzales M. C., Hidalgo C. A. and Barabasi A.-L.: Understanding individual human mobility patterns. *Nature* 453, 779-782. (2008)
10. Guc B., May M., Saygin Y. and Korner C.: Semantic Annotation of GPS Trajectories. *11th AGILE International Conference on Geographic Information Science 2008*, Girona, Catalonia, Spain, (2008)
11. Ishikawa Y.: Data Mining for Moving Object Databases. In: Laurence T. Y, editor. *Mobile Intelligence: Mobile Computing and Computational Intelligence*, John Wiley & Sons, (2008)
12. Jensen C. S., Lahrmann H., Pakalnis S., and Runge J.: The INFATI Data. A TimeCenter Technical Report, (2004)
13. Jensen C. S., Lee K.-J., Pakalnis S., Saltenis S.: Advanced Tracking of Vehicles. In *Proceedings of the Fifth European Congress and Exhibition on Intelligent Transport Systems*, Hannover, Germany (2005)
14. Krumm J.: A Markov Model for Driver Turn Prediction. *SAE World Congress*, Detroit, MI, USA, (2008)
15. Krumm J. and Horvitz E.: Predestination: Inferring Destinations from Partial Trajectories. *International Conference on Ubiquitous Computing*, Orange County, CA, USA, (2006)
16. Larson R. R.: Geographic Information Retrieval and Spatial Browsing" (book chapter in *Geographic Information Systems Patrons Maps and Spatial Information*), University of Illinois, Urbana-Champaign, 81-123. (1995)
17. Miller H. J.: Geographic Data Mining and Knowledge Discovery (book chapter in *Handbook of Geographic Information Science*), Wiley-Blackwell, (2007)
18. Nižetić I. and Fertalj K.: Analyzing Moving Objects Behaviour and Predicting their Future Location. In *Proceedings of the Junior Scientist Conference*, Vienna, Austria, 87-88. (2008)
19. Nižetić I., Fertalj K., Kalpić D.: A Prototype for the Short-term Prediction of Moving Object's Movement using Markov Chains. In *Proceedings of the ITI 2009*, Cavtat, Croatia, 559-564. (2009)
20. Petzold J., Pietzowski A., Bagci F., Trumler W. and Ungerer T.: Prediction of Indoor Movements Using Bayesian Networks. *Lecture Notes in Computer Science* 3479, Springer-Verlag, 211-222., (2005)

Automation of the Moving Objects Movement Prediction Process Independent of the Application Area

21. Praing R. and Schneider M.: A Universal Abstract Model for Future Movements of Moving Objects. Lecture Notes in Geoinformation and Cartography, 111-120. (2007)
22. Rabiner L. R.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In: Proceedings of the IEEE, Vol. 77, No. 2, (1989)
23. Ryu K. and Ahn Y.: Application of moving objects and spatio-temporal reasoning. A TimeCenter Technical Report TR-58., (2001)
24. Saab, S.S., Kassas, Z.M.: Power matching approach for GPS coverage extension. In proceedings of Intelligent Transportation Systems, IEEE Transactions, Vol. 7, Issue 2, 156-166. (2006)
25. Sims W. et al.: Scaling laws of marine predator search behaviour. Nature 451, 1098-1102. (2008)
26. Tao Y., Faloutsos C., Papadias D. and Liu B.: Prediction and Indexing of Moving Objects with Unknown Motion Patterns. In Proceedings of the 2004 ACM SIGMOD international conference on Management of data, 611-622. (2004)
27. Yavas G., Katsaros D., Ulusoy O. and Manolopoulos Y.: A data mining approach for location prediction in mobile environments. Data & Knowledge Engineering. Vol. 54, Issue 2, 121-146. (2005)

Krešimir Fertalj is an associate professor at the Department of Applied Computing at the Faculty of Electrical Engineering and Computing, University of Zagreb. Currently he lectures a couple of undergraduate and postgraduate courses in Computing. His professional and scientific interest is in computer-aided software engineering, complex information systems and in project management. He has written over a hundred scientific and professional publications and participated in conferences locally and abroad. He participated in a number of information system designs, implementations and evaluations. Fertalj is member of ACM, IEEE and PMI.

Ivana Nižetić is a research assistant at the Department of Applied Computing at the Faculty of Electrical Engineering and Computing, University of Zagreb and Ph. D. student in Computer Science. She graduated in 2005. at the Department of Mathematics at Faculty of Science, University of Zagreb. Her research interests include software development and spatio-temporal representation and reasoning. She is a member of IEEE.

Received: June 08, 2009; Accepted: February 03, 2010.

CIP – Каталогизacija y publikaciji
Народна библиотека Србије, Београд

004

COMPUTER Science and Information
Systems : the International journal /
Editor-in-Chief **Mirjana Ivanović**. – Vol. 7,
No 4 (2010) - . – Novi Sad (**Trg D. Obradovića**
3): ComSIS Consortium, 2010 - (Belgrade
: Sgra star). –30 cm

Polugodišnje. – Tekst na engleskom jeziku

ISSN 1820-0214 = Computer Science and
Information Systems
COBISS.SR-ID 112261644

Cover design: **V. Štavljanin**
Library Administration: **M. Petković**
Printed by: Sgra star, Beograd

ComSIS Vol. 7, No. 4, December 2010



Contents

Editorial

Disclosure of Plagiarism

Invited Paper

- 679 Advanced Indexing Technique for Temporal Data
Bela Stantic, Rodney Topor, Justin Terry, Abdul Sattar

Papers

- 705 COLIBROS: Educational Operating System
Zarko Živanov, Predrag Rakić, Miroslav Hajduković
- 721 An Approach to Project Planning Employing Software and Systems Engineering
Meta-Model Represented by an Ontology
Miroslav Liška, Pavol Návrát
- 737 Facilitating Information System Development with Panoramic View on Data
Dejan Lavbič, Iztok Lajovic, Marjan Krisper
- 769 Metrics for Evaluation of Metaprogram Complexity
Robertas Damaševičius, Vytautas Štuikys
- 789 An Evolutionary Solution for Multimodal Shortest Path Problem in Metropolises
Rahim A. Abbaspour, Farhad Samadzadegan
- 813 Trojan horses in mobile devices
Daniel Fuentes, Juan A. Álvarez, Juan A. Ortega, Luis Gonzalez-Abril, Francisco Velasco
- 823 A Tabular Steganography Scheme for Graphical Password Authentication
Tsung-Hung Lin, Cheng-Chi Lee, Chwei-Shyong Tsai, Shin-Dong Guo
- 843 A Multi-attribute Auction Model by Dominance-based Rough Sets Approach
Rong Zhang, Bin Liu, Sifeng Liu
- 859 Design of Median-type Filters with an Impulse Noise Detector Using Decision Tree and
Particle Swarm Optimization for Image Restoration
Bae-Muu Chang, Hung-Hsu Tsai, Xuan-Ping Lin, Pao-Ta Yu

Papers selected from "The 4th International Conference on Information Technology", June 3 - 5, Amman, Jordan 2009

- 883 Personalising the Dynamic Information Resources in a Self Organized Web System
Using CAS and MAS
Rattout Amjad, Assaf Rasha, Al-Dahoud Ali
- 907 A Hybrid Variable Neighborhood Search Algorithm for Solving Multi-Objective
Flexible Job Shop Problems
Jun-qing Li, Quan-ke Pan, Sheng-xian Xie
- 931 Automation of the Moving Objects Movement Prediction Process Independent
of the Application Area
Ivana Nižetić, Krešimir Fertalj