Computer Science and Information Systems

Published by ComSIS Consortium

Volume 9, Number 1
January 2012

Computer Science and Information Systems

Vol 9, No 1, January 2012

ComSIS is an international journal published by the ComSIS Consortium

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
Managing editor: Gordana Rakić, 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. Andreae, 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. Böszörmenyi, 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. Đurkovic, 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. Kašćelan, 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 Zeland
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 Cantrebury, 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. Đorđević-Kajan, University of Niš, Serbia
P. Hotomski, University of Novi Sad, Serbia
Z. Jovanović, University of Belgrade, Serbia
L. Kalinichenko, Russian Academy of Scence, 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, Atilim University, Turkey
S. Mishra, Atilim 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, FYR of Macedonia

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 9, Number 1, 2012
Novi Sad

Computer Science and Information Systems

ISSN: 1820-0214

ComSIS Journal is sponsored by:

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

Computer Science and Information Systems

AIMS AND SCOPE

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

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

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

Indexing Information

ComSIS is covered or selected for coverage in the following:

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

Information for Contributors

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

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 9, Number 1, January 2012

CONTENTS

Editorial

Papers

1 **A Personalized Multimedia Contents Recommendation Using a Psychological Model**
Won-Ik Park, Sanggil Kang, Young-Kuk Kim

23 **Representation of Texts in Structured Form**
Mladen Stanojević, Sanja Vraneš

49 **A Genetic Algorithm for the Routing and Carrier Selection Problem**
Jozef Kratica, Tijana Kostić, Dušan Tošić, Djordje Dugošija, Vladimir Filipović

63 **An XML-algebra for Efficient Set-at-a-time Execution**
Maxim Lukichev, Boris Novikov, Pankaj Mehra

81 **A Decision Support Method for Evaluating Database Designs**
Erki Eessaar, Marek Soobik

107 **An Enterprise Information System Agility Assessment Model**
Rabah Imache, Said Izza, Mohamed Ahmed-Nacer

135 **Selecting a Methodology for Business Information Systems Development: Decision Model and Tool Support**
Damjan Vavpotič, Olegas Vasilecas

165 **Improving the Evaluation of Software Development Methodology Adoption and its Impact on Enterprise Performance**
Damjan Vavpotič, Tomaž Hovelja

189 **A Framework for Developing and Implementing the Enterprise Technical Architecture**
Tiko Iyamu

207 **Avoiding Unstructured Workflows in Prerequisites Modeling**
Boris Milašinović, Krešimir Fertalj

235 **Improving Program Comprehension by Automatic Metamodel Abstraction**
Michal Vagač, Ján Kollár

249 **An Approach to Automated Conceptual Database Design Based on the UML Activity Diagram**
Drazen Brđanin, Slavko Marić
285 **Usage of Technology Enhanced Learning Tools and Organizational Change Perception**
Mladen Čudanov, Gheorghe Savoiu, Ondrej Jaško
303 **Effective Hierarchical Vector-based News Representation for Personalized Recommendation**
Mária Bieliková, Michal Kompan, Dušan Zeleník
323 **Concept Extraction and Clustering for Search Result Organization and Virtual Community Construction**
Shihn-Yuarn Chen, Chia-Ning Chang, Yi-Hsiang Nien, Hao-Ren Ke
357 **A Method for Decision Making with the OWA Operator**
José M. Merigó, Anna M. Gil-Lafuente
381 **A Generic Parser for Strings and Trees**
Riad S Jabri
411 **Using Lightweight Formal Methods to Model Class and Object Diagrams**
Fernando Valles-Barajas
431 **S-TRAP: Optimization and Evaluation of Timely Recovery to Any Point-in-time (TRAP)**
Chao Wang, Zhanhuai Li, Na Hu, Yanming Nie
455 **A Comparative Evaluation of Effort Estimation Methods in the Software Life Cycle**
Jovan Popović, Dragan Bojić

DOI 10.2298/CSIS100618036P

A Personalized Multimedia Contents Recommendation Using a Psychological Model

Won-Ik Park[1], Sanggil Kang[2], and Young-Kuk Kim[1*]

[1]Dept. of Computer Science & Engineering, Chungnam National University, Gung-dong, Youseong-gu, Daejeon, South Korea
E-mail: {wonik78, ykim}@cnu.ac.kr
[2]Dept. of Computer Science & Information Engineering, Inha University, Yonghyun-dong, Nam-gu, Incheon, South Korea
E-mail: sgkang@inha.ac.kr

Abstract. With the development and diffusion of compact and portable mobile devices, users can use multimedia content such as music and movie on personal mobile devices, anytime and anywhere. However, even with the rapid development of mobile device technology, it is still not easy to search multimedia content or manage large volume of content in a mobile device with limited resources. To resolve these problems, an approach for recommending content on the server-side is one of the popular solutions. However, the recommendation in a server also leads to some problems like the scalability for a lot of users and the management of personal information. Therefore, this paper defines a personal content manager which acts between content providers (server) and mobile devices and proposes a method for recommending multimedia content in the personal content manager. For the recommendation based on user's personal characteristic and preference, this paper adopts and applies the DISC model which is verified in psychology field for classifying user's behavior pattern. The proposed recommendation method also includes an algorithm for reflecting dynamic environmental context. Through the implements and evaluation of a prototype system, this paper shows that the proposed method has acceptable performance for multimedia content recommendation.

Keywords: Multimedia Content Recommendation, Personalization, DISC Model, Dynamic Environmental Context.

1. Introduction

Recently, a great deal of information has become available and many people are sharing that information on the Internet thanks to the development of Internet technology. People can download and enjoy desired content conveniently, as multimedia content (music, video, pictures, etc.) have been

* To whom correspondence should be addressed.

digitized. In addition, personal computers can now store large volume of multimedia content with growth of computer performance. Many kinds of mobile devices are developed. In this environment, users want to use a multimedia content through a personal mobile device, anytime and anywhere. In IPTV application which is one of the applications for a multimedia content service, a user purchases a variety of multimedia content from a content service provider and stores that into a personal Set-Top Box. And then the user uses the stored content through personal mobile devices without a dependence on time and position.

In this environment, a user hopes to search and acquire multimedia content efficiently. Furthermore, the user expects that a mobile device automatically searches and proposes a customized content for the user. For these requirements, of course, it is the best solution to store all multimedia content in a personal mobile device, but that is unrealistic because of the storage-size problem of the mobile device. Another solution for acquiring a customized content is to access a content service provider through a personal mobile device directly. However this approach is not easy because the mobile device has limited resources like inconvenient I/O and tiny display. Moreover the content service provider should consider the searching and recommending service for many users. Therefore there are some problems like the scalability for a lot of users and the management of personal information.

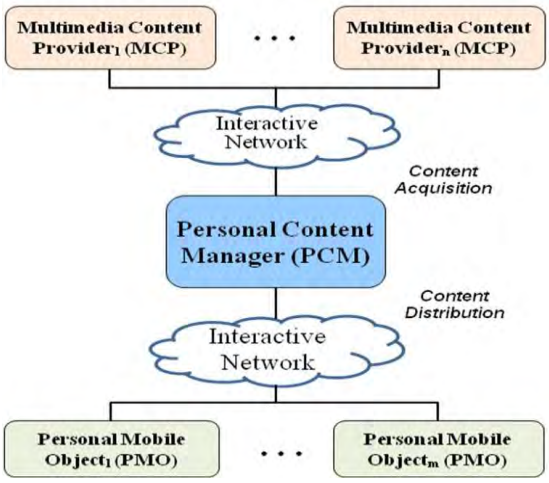


Fig. 1. Multimedia content service environment

This paper proposes the method to infer a multimedia content in a personal content manager (PCM) and recommend to a user's mobile device like Fig. 1

If the PCM recommends multimedia content, previous mentioned problems are solved naturally. Since the PCM intercommunicates between content providers and mobile objects, we can obtain available information for recommendation from two tiers.

This paper puts forward methods for solving the following issues for the customized multimedia content recommendation in the PCM.

How can personal characteristic and preference be applied to the recommendation?

There are a lot of previous researches in extracting user information from user history and profile and applying the information to a recommendation. However it is not sure that the reliability and satisfaction of their approaches have a high score because the studies about user history and information are in psychology and they are experts in computer engineering and science. Therefore we need to select and apply a verified method for manipulating user information. To extract user characteristics from user's behavior pattern, some methods are proposed and verified in psychology. Therefore we should adopt and apply one of the verified methods for the multimedia content recommendation. Although the recommendation service is based on user characteristics, users hope to obtain a different result by requesting the same recommendation service in the different time and position. In other words, users want to get a recommendation service based on user's dynamic context. Therefore we should consider how context information can be extracted and applied for the multimedia content recommendation.

This paper proposes a method to apply user characteristics to the content recommendation based on the consumption pattern derived from the user's behavior pattern, which has been thoroughly studied by the social and human sciences. The paper also proposes a method to reflect current user context to the recommendation by log history and environmental context information when the PCM accesses the content providers and the mobile objects connects the PCM, respectively. To evaluate our methods, this paper designs and implements a Jukebox based on proposed methods.

The remainder of this paper is organized as follows. Section 2 describes the related work and the DISC behavior pattern model. Section 3 explains the classification of DISC behavior patterns for our recommendation. Section 4 describes our methods for the multimedia content recommendation in detail. In Section 5, we design and implement a prototype system based on the proposed method and show the performance evaluation in Section 6. Finally, Section 7 provides concluding remarks.

2. Background and Related Work

2.1. Related Work

In general, there are three kinds of approaches in developing a recommender system: content-based, collaborative and hybrid ones[1]. Content-based recommendation is the technique whereby content is blocked or allowed based on analysis of its content, rather than its source or other criteria. [2, 3, 4, 5, 6, 7] propose a method for recommending TV programs for a user in

Personal Digital Recorder (PDR). For recommendation, [2, 4] uses implicit view history, explicit TV viewing preferences, and feedback information and [5, 7] propose an algorithm for estimating user preference score based on user history. The recommender system described in [3, 6] infer preference based on user profile and usage history and recommend a customized content.

Collaborative recommendation is recommended items that people with similar tastes and preferences liked in the past. It is categorized as a user-based collaborative recommendation[8] and an item-based collaborative recommendation[9]. Similar users and(or) items are sorted based on the memorized ratings. Relying on the ratings of these similar users and(or) items, a prediction of an item rating for a test user can be generated[10,11,12]. However they don't discuss applying personal information like user profile. Hybrid recommendations combine collaborative and content-based methods [13, 14, 15, 16, 17, 18]. They help to avoid certain limitations of content-based and collaborative systems. Even if their methods provide a personalized service based on the user preference, however, the result of their recommendation is the same when a user requests the same service in the different situations. In other words, a user wants to get a customized service based on current user situation like time, position, and weather. Therefore we need the recommendation method reflected user context. [19, 20, 21, 22, 23] propose a recommendation method based on user situation, although an application domain is not the same as ours. [19] proposes a method to apply user context by calculating context similarity among users and [22] deal with environmental context by using an ontology and rule. Especially [23] study that it is possible to infer the context from the existing data with reasonable accuracy in certain cases. Actually our approach using clustered-based method is similar with many model-based recommender systems, a classic reference of which can be found at [1]. However, previous researches are proposed each using user preference, history, and context information to the recommendation. Also there are not approaches that are optimized to apply user psychological characteristics.

This paper adopts the DISC Type to apply user psychological characteristics to the content recommendation. DISC Model is one of the popular methods to classify psychological characteristics based on user behavior pattern in psychological field [24, 25, 26]. By adopt DISC model for the multimedia content recommendation, we can expect a synergy effect in computer engineering and psychology fields.

2.2. DISC Model

Generally, people take action selectively based on their personality type and their own unique motivations - the influences that have affected them since birth. People act quite naturally and comfortably in their working or living environment, as those patterns form a kind of tendency, which is called a behavior pattern.

William Moulton Marston, a psychology professor at Columbia University, proposed the DISC behavior patterns in 1928, classifying behavior patterns into the D (Dominance) type, I (Influence) type, S (Steadiness) type, and C (Conscientiousness) type. Table 1 shows the psychological characteristics of each behavior pattern [24, 25, 26].

This paper classifies user's behavior patterns according to the DISC type, using the correlation between those four behavior patterns and the user's purchasing patterns, and proposes a new recommender system which provides a customized service for recommending a multimedia content based on user preference.

Table 1. Psychological characteristics by DISC type

DISC type	Description
D type (Dominance)	Independent, persistent, direct. Energetic, busy, fearless. Focus on own goals rather than people. Ask 'What?'
I type (Influence)	Social, persuasive, friendly. Energetic, busy, optimistic, distractible. Ask 'Who?'
S type (Steadiness)	Consistent, like stability. Accommodating, peace-seeking. Good listeners and counselors. Ask 'How?' and ask 'What?'
C type (Conscientiousness)	Slow and critical thinker, perfectionist. Logical, fact-based, organized, follows rules. Ask 'Why?' and 'How?'

[http://changingminds.org/explanations/preferences/disc.htm]

3. Classification of behavior patterns

To apply a personal characteristic and preference to the recommendation, the first issue described in the introduction, this paper proposes a method to extract and classify the characteristic and preference from user's behavior patterns based on the DISC type.

Originally, there are two methods of identifying a user's behavior pattern - an explicit method based on a questionnaire, and an implicit method that analyzes the user's purchase-making behavior pattern. The explicit method is a simple and traditional method that is used to identify the user's behavior pattern using the DISC questionnaire, based on data evaluated by the user. It is useful before user's initial behavior. However, it takes up the user's time and may cause inconvenience. On the other hand, the implicit method is convenient for users because the user's purchase-making pattern can be analyzed and identified without the user's direct involvement. However, one problem is that the user's psychological pattern cannot be identified if the user's initial purchase-making behavior cannot be understood.

The method proposed in this paper involves classification of behavior pattern according to the user's purchasing actions, by mixing those two methods. That is, the explicit method (using the summarized DISC questionnaire) is applied to users who have not purchased anything, whereas the implicit method is applied to users who have purchased something once more, in order to identify the user's behavior pattern.

Explicit method

In the explicit method, the summarized DISC questionnaire items are presented to analyze the user's behavior pattern. Normally, a description is given according to the category that best fits the behavior description, and the corresponding score is given. The user's behavior pattern is classified through the description category that the total score falls in.

Implicit method

The implicit method analyzes the user's purchase-making pattern in order to determine the behavior pattern. This section explains the implicit method, which determines the DISC behavior pattern using the search pattern in a music site with a search structure shown in Figure 2.

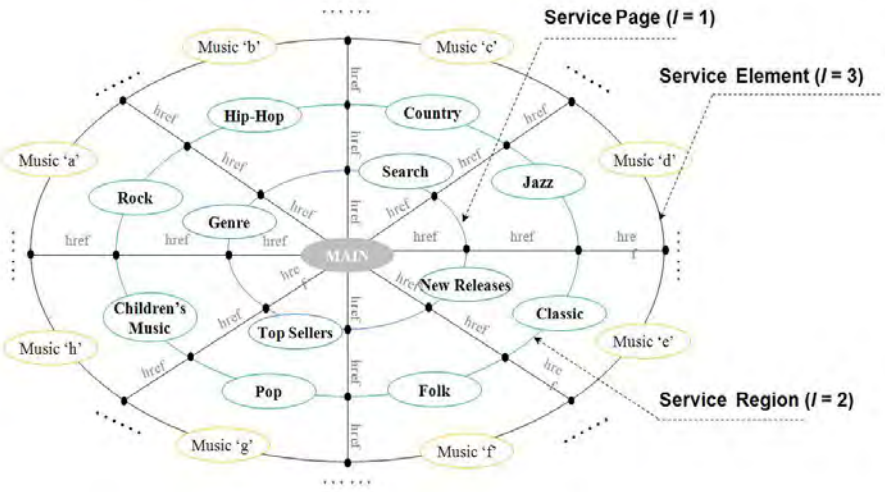


Fig. 2. The structure for searching content in a music website

Generally, websites that provide multimedia content are composed of three structural levels: a service page, a service region, and a service element. The main service page consists of one or more service pages and each service page includes one or more service regions. In addition, each service region also has at least one service element. For convenience, the service page, service region, and service element are here described as levels l=1, l=2, and l=3.

To classify the user's behavior pattern, this paper considers 'search depth', 'search variety', and 'staying time'. The 'search depth' is the number of accessing to levels for purchasing a multimedia content at level 3. For

example, if a user's search path is main → new release (level: 1) → music 'A' (level:3) → genre (level:1) → music 'B' (level:3-purchase), then the 'search depth' is 3. In addition, 'search variety', which involves the number of movements to the same level or an upper level (*l*=3→*l*=1, *l*=3→*l*=2, *l*=3→*l*=3, *l*=2→*l*=1, *l*=2→*l*=2), becomes 1, because this involves movement to music 'A' (level: 3) → genre (level: 1) in the example. Lastly, 'staying time' is the total time taken to purchase certain content after logging in.

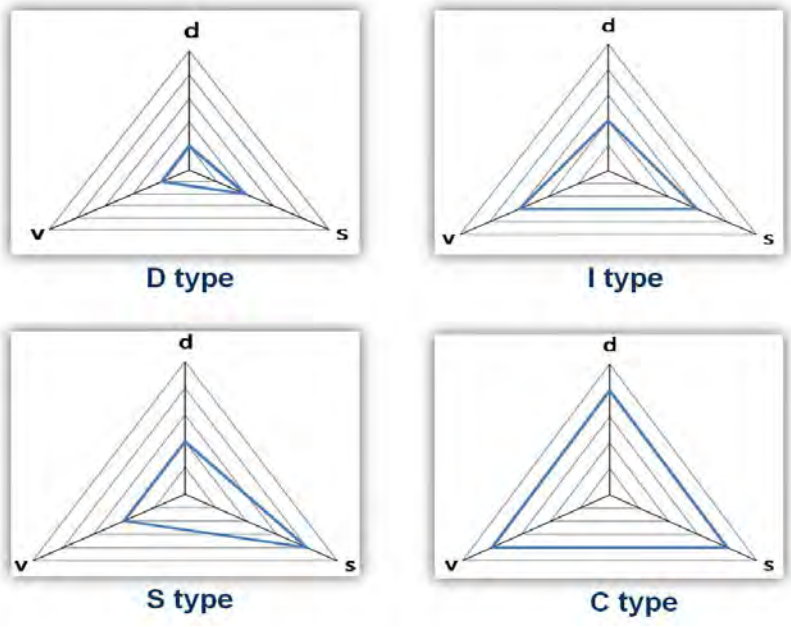


Fig. 3. Association of the *wLog* with the DISC type

Generally, in the normal web browser functions, it is difficult to extract the 'search depth', 'search variety', and 'staying time' from the client's side instead of the server's. Therefore, this paper proposes a specialized web browser to acquire this information by using the meta-tag [27]. The meta-tag is one of the methods that allow adding of information to a web page, and is not displayed on the web browser for the user. Using this tag, we can recognize the user's location and extract the final search depth. The 'search variety' also can be extracted by the meta-tag information. The 'staying time' can be checked, based on the time log that begins with accessing the page for searching and ends at the purchase time. However, it is assumed that users receive the confirmation message from the server at the time of purchase. The 'search depth', 'search variety', and 'staying time' obtained through our method are saved as the coordinate vector value on the personal content manager, instead of the server. This paper defines *wLog* to express this information as follows.

$$wLog = (\ d, v, s\ ) \quad , \begin{Bmatrix} d = search\ depth \\ v = search\ \text{variety} \\ s = staying\ time \end{Bmatrix} \tag{1}$$

To find a relationship between the classification using the wLog and the DISC model, this paper defines the following hypothesis and verifies the hypothesis through an experiment.

Hypothesis:

D type - Short staying time and insufficient search variety because the intended information is definitive and a purchase decision is made quickly. Search depth is not deep because purchases are frequently made through a search.

I type - Search depth is relatively shallow because the user likes the multimedia content preferred by others, such as popular content, top-selling content, and recommended content. Search variety and staying time are normal.

S type - Staying time at frequently visited pages is consistently longer. Search depth is shallow and search variety is insufficient because the user prefers a specific genre or artist.

C type - The user reviews or listens to samples on various pages. The user takes a long time to make a purchase decision, checking price and performance. Search depth, search variety, and staying time are relatively deep, intensive, and long.

4. The recommendation of multimedia content

To determine the user's behavior pattern, this paper classifies users based on the *wLog* value. Target users are classified using the distance from the representative value of the defined behavior pattern group. The representative value is the centroid value of 3 clusters of vectors. The centroid is calculated by dividing the sum of the apex coordinates by the number of vectors. The following equation (2) is used to decide a user's type. The equation calculates the distance of the target user *u*'s purchase behavior pattern vector p_u from the vector P_j which is the vector composed by representative values of cluster *j*'s purchase behavior pattern. *k* is the attribute suffix (1:*d*, 2:*v*, 3:*s*) of the purchase behavior pattern vector.

$$d_{u,j}(p_u, P_j) = \sqrt{\sum_{k=1}^{3} (p_{uk} - P_{j_k})^2} \tag{2}$$

For example, if the *wLog* value of the user *x* is (*d:*5, *v*:2, *s*:18) and the representative value of $P_{j=D}$, $P_{j=I}$, $P_{j=S}$, and $P_{j=C}$ are (*d:4*, *v*:4, *s*:8), (*d:8*, *v*:16, *s*:16), (*d:8*, *v*:8, *s*:16), and (*d:16*, *v*:16, *s*:16), then the value of $d_{x,j}(p_x,P_{j=D})$,$d_{x,j}(p_x,P_{j=I})$,$d_{x,j}(p_x,P_{j=S})$,and $d_{x,j}(p_x,P_{j=C})$ are $\sqrt{105}$, $\sqrt{209}$, $\sqrt{49}$, and $\sqrt{321}$, respectively. Therefore, we can know that the user *x* is in the S type because the distance value of S type is the smallest.

To solve the issue for applying dynamic user's environmental context to the recommendation which is described in the introduction, this paper infers the user's preferred genre according to the context, using *uLog* which is the user's usage history of multimedia content on the mobile device. Figure 4 is the schema of the 'UsageData' which is defined in this paper to describe *wLog*, *uLog* and the metadata of a multimedia content.

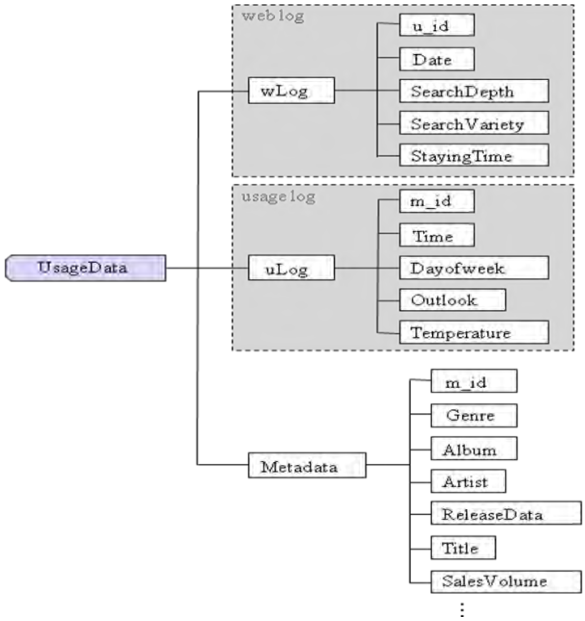


Fig. 4. The schema of the UsageData

Table 2. User Context information

	Outlook	Time	DayOfWeek	Temperature(℃)
1	Sunny	Dawn	Wed	15
2	Overcast	Afternoon	Sat	24
:	:	:	:	:
9	Rain	Morning	Mon	19
10	Sunny	Evening	Tue	23
:	:	:	:	:
14	Overcast	Night	Thu	22
15	Rain	Afternoon	Sun	21

As the algorithm for inferring the preferred genre, we use the decision tree algorithm. Table 2 and table 3 are an example of the user's context information included in *uLog* and of the decision tree structure for multimedia content, respectively.

Table 3. The decision tree for inferring a multimedia content

```
1:Dawn?
T-> 0:Overcast?
   T-> {'Folk': 1}
   F-> {'Country': 2}
F-> 1:Afternoon?
   T-> 3:24?
      T-> 0:Overcast?
         T-> {'Pop': 1}
         F-> {'Folk': 1}
      F-> {'Rock': 2}
   F-> 1:Morning?
      T-> 2:Tue?
         T-> {'Classic': 1}
         F-> {'Pop': 2}
      F-> 2:Sat?
         T-> 0:Overcast?
            T-> {'Hip-Hop': 1}
            F-> {'Classic': 1}
         F-> {'Jazz': 3}
```

The DISC type classified by *wLog* and preferred genre inferred by *uLog* are used to recommend a multimedia content together. In other words, a different weighted value is given to each DISC type with regard to the ‘UsageData’, ‘Popularity’, ‘Artist’, and ‘Title’ of the multimedia content in the preferred genre which is inferred, in order to finally recommend specific multimedia content. For this recommendation, this paper defines the representative attributes that affect each behavior pattern, considering the fact that the user's purchase-making behavior pattern is closely related to the DISC type. For the D type, the ‘UsageData’ is selected as the representative attribute, because users with this behavior pattern are willing to accept a challenge, make a decision quickly, and show great interest in new content. That is, the ‘UsageData’ attribute of the multimedia attributes is the main attribute for D (Dominance) type users and a weighted value is given to recently released-multimedia content. I (Influence) type users like contact with people, and respect other people's opinions and comply with them. Due to these behavior pattern characteristics, weight is given to the ‘Popularity’ attribute from among the four attributes. In this paper, the ‘Popularity’ attribute is estimated by the ‘SalesVolume’ attribute in the metadata. S (Steadiness) type users perform consistently and have introspective and relaxed behavior pattern characteristics. For S type users, the ‘Artist’ attribute is representative, because they tend to show high loyalty towards their preferred artists. Lastly,

C (Conscientiousness) type users show the behavior pattern of thinking analytically. C type users have the same level of preference for all multimedia content in the preferred genre inferred. Therefore, we define that the 'Title' is the representative attribute and weight is the frequency of the content used by the user.

The attribute vector (AV) for the multimedia content is equal to (*r, p, m, f*). Here, *r* is the release date of the multimedia content and has a value range of 1 to 12; *p* stands for the popularity of the content and has a value range of 1 to 5; and *m* stands for the number of items of the same artist and has a value range of 1 to 10. Lastly, *f* - the number of times the user has used the content, has a value range of 1 to 50. The attribute vector value, which has a different value range, can be normalized as *r', p', m', f'*, which are the values between [0, 1], using the equation (3). The evaluation value of the content according to the user's behavior pattern is calculated for the normalized attribute vector AV', using the equation (3). For example, if the AV of multimedia content *m* is (*r:*9; *p:*2; *m:*3; *f:*12) and the set of min_*n* and max_*n* pair of each attribute is (*r:*1,9; *p:*1,5; *m:*1,6; *f:*10,30), then AV' is (*r':*1; *p':*0.25; *m':*0.4; *f':*0.1).

$$n' = (n - \min\_n)/(\max\_n - \min\_n) \qquad n = \{r, p, m, f\} \tag{3}$$

Where, n stands for r, p, m, and f - the element of the attribute vector (AV), whereas min_n and max_n are the minimum and maximum values of each element.

$$w_{u,j} = \frac{1}{\left| 1 - d_{u,j}(p_u, P_j) \right|} \qquad , j = \{1:D, 2:I, 3:S, 4:C\} \tag{4}$$

That is, $w_{u,j}$ shows the extent of the user's association with each behavior pattern.

$$r_{u,c} = \sum_{i=1}^{4} w_{u,j} \times AV'_{c,i}, i = \{1:r', 2:p', 3:m', 4:f'\} \tag{5}$$

The weight w_j is applied to the user's behavior pattern *j*, as shown in the equation (4). The user *u*'s preference score $r_{u,c}$ for the content *c* applied with the weight can be estimated by the equation (5). The equation (5) can be explained in detail as follows:

$r_{u,c} = (w_{u,D} \times AV'_{c,r'}) + (w_{u,I} \times AV'_{c,p'}) + (w_{u,S} \times AV'_{c,m'}) + (w_{u,C} \times AV'_{c,f'})$

For example, The weight $w_{x,j=S}$ of the user *x* which are described in the previous example is $1/(1-\sqrt{49})$. Also the preference score $r_{x,m}$ between multimedia content *m* and the user *x* is 7/24.

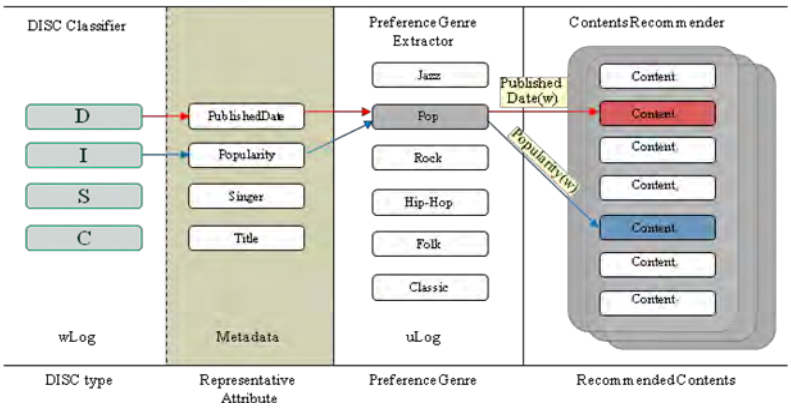


Fig. 5. An example of the schematic of the recommendation of the user's preferred multimedia content

Fig. 5 shows that the equation (5) gives different weight to each attribute value using the DISC type.

- The weighted value for the user is assigned as D type to the attribute *r'* to the extent that there is a similarity with the D type.
- The weighted value for the user is assigned as I type to the attribute *p'* to the extent that there is a similarity with the I type.
- The weighted value for the user is assigned as S type to the attribute *m'* to the extent that there is a similarity with the S type.
- The weighted value for the user is assigned as C type to the attribute *f'* to the extent that there is a similarity with the C type.

5. Personal Multimedia Jukebox

This paper proposes and implements the personal multimedia jukebox (PMJ) as an application system for the personal content manager described in the introduction. Fig. 6 shows the overall architecture of the system for providing personalized multimedia content recommendations in the mobile environment. The PMJ manages a lot of multimedia items purchased from the Content Provider (MCstore), and distributes the personalized service to mobile clients (mobile objects). The proposed Jukebox is composed of three modules. The Collecting Module (CM) collects and manages *wLog* (user's purchase behavior log), *uLog* (usage log of the mobile client), and multimedia content with metadata provided by the content provider. The Recommending Module (RM) classifies the behavior pattern using the information collected by the CA, and recommends personalized content based on the classified behavior pattern and context information. Lastly, the Propagating Module (PM) provides recommended content to the mobile client.

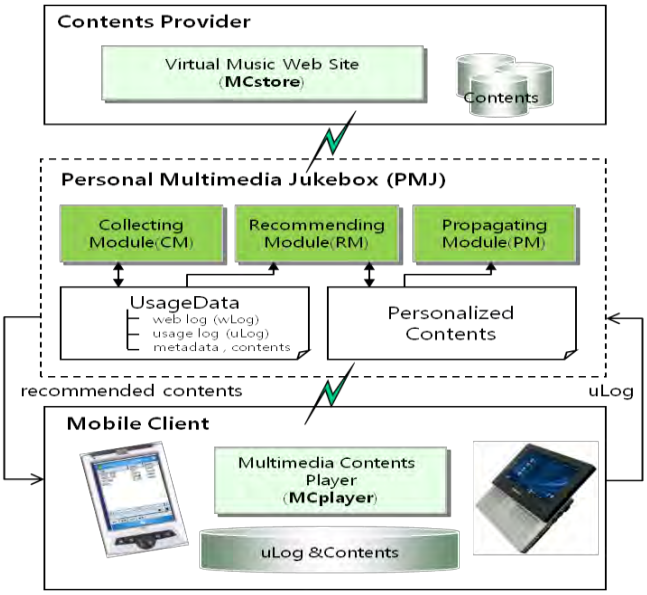


Fig. 6. The system architecture for providing personalized multimedia content recommendations

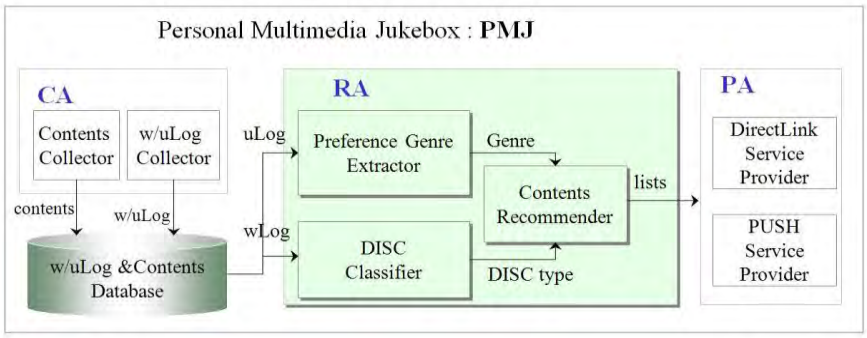


Fig. 7. The architecture of the Personal Multimedia Jukebox

Fig. 7 shows the architecture of our PMJ in detail. The CM collects *wLog*, *uLog*, and metadata regarding the content from the content provider and mobile clients. *wLog* includes the *d*, *v*, and *s* records and *uLog* is the user's usage history. Lastly, the metadata contained in the content is collected and saved. The RM recommends personalized content using the usage data collected by the CM. For recommendation, this paper uses the DISC type obtained by using the user's web log, namely *wLog* and the preferred genre obtained by using the user's usage log, namely *uLog* by the DISC Classifier and Preference Genre Extractor, respectively. The Contents Recommender generates a list of multimedia content for a user based on extracted the DISC

type and the preferred genre. The PM provides a recommendation service to mobile client through wire or wireless network.

6. Experimental Results

6.1. Prototype implementation

For evaluation of our system, this paper built the music website (MCstore) in the server system like Fig. 7 and implemented a multimedia content player (MCplayer) on a mobile client like Fig. 8.

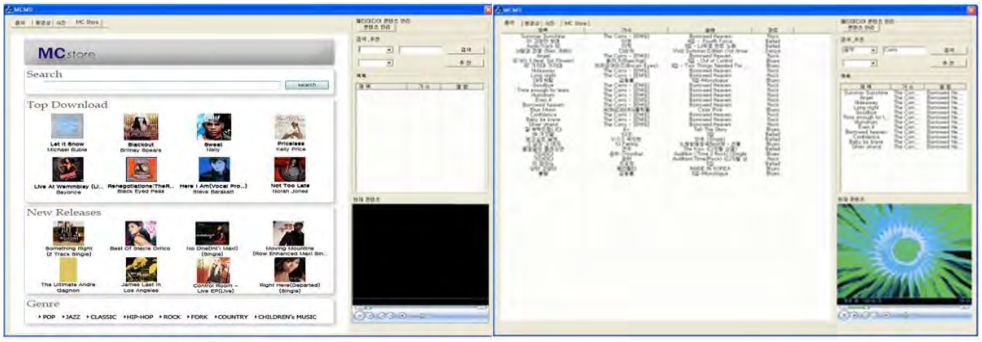


Fig. 8. The interface of the MCstore

Fig. 9. The emulator for evaluating our recommendation

MCstore was implemented using MySQL 5.0 as the DBMS to store wLog and the content on Windows XP, and Tomcat 5.5 was used as an application server for Java and JSP. To install PMJ on a personal computer, MySQL 5.0 was used to store uLog and wLog, and the content of Windows XP was also

used. MCplayer is a multimedia player that can be run on a PDA that supports Windows Mobile 5.0.

6.2. Performance Evaluation

In order to evaluate our user pattern classification and recommendation performance by comparing with conventional methods, we distributed the programs (PMJ.exe, MCplayer.exe) to 80 students enrolled at Chungnam National University and have collected their usage history during six months.
In order to verify our user behavior pattern classification developed with adopting the DISC, we test the hypothesis regarding the association between the purchase-making pattern established by this study and the DISC obtained from the DISC psychological pattern questionnaire as seen in Fig. 10. In the figure, the D type and S type show a relatively higher degree of accuracy of classification than the I type and C type. This fact proves that the proposed method of classification of the D type and S type behavior patterns is more efficient. In addition, the accuracy of all the classification types increases after the initial evaluation.

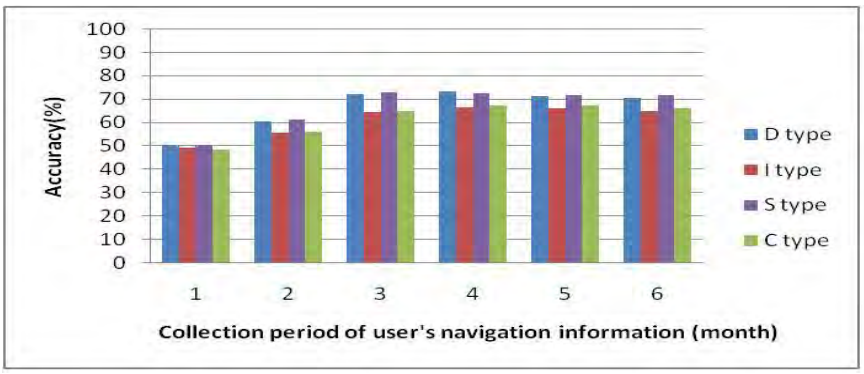


Fig. 10. Performance of user behavior pattern classification based on DISC type

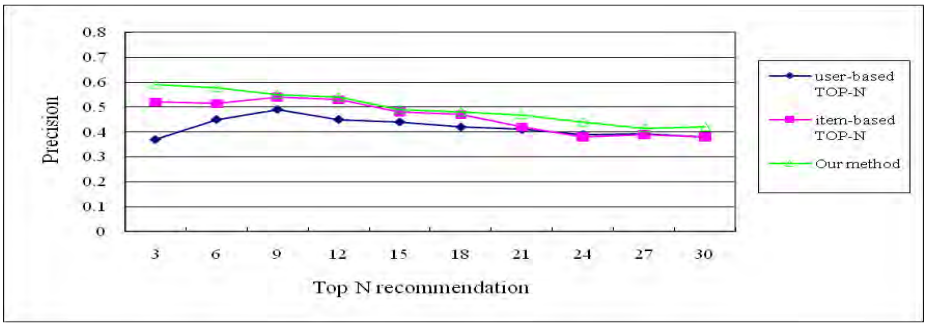
To evaluate our recommendation performance, we compare the performance of our method with conventional methods such as user-based Top-N and item-based Top-N. Also, we chose the precision and recall [28, 29, 30] as a metric of the classification performance, widely used for measuring the recommendation performance in information retrieval applications. The equations (6) and (7) are the formula of estimating the precision and recall performance respectively. Let the set of music items relevant to a query be denoted as {Relevant}, and the set of music items retrieved be denoted as {Retrieved}. The set of music items that are both relevant and retrieved is denoted as {Relevant} $\cap${Retrieved}.

$$precision = \frac{\left|\{\text{Relevant}\} \cap \{\text{Retrieved}\}\right|}{\left|\{\text{Retrieved}\}\right|} \tag{6}$$

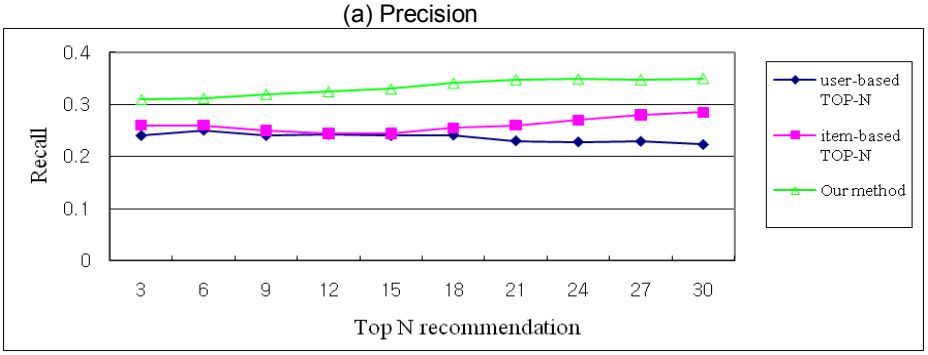
$$recall = \frac{\left|\{\text{Relevant}\} \cap \{\text{Retrieved}\}\right|}{\left|\{\text{Relevant}\}\right|} \tag{7}$$

Also the F-measure is a measure of a test's accuracy. It considers both the precision and recall of the test to compute score. The precision and recall metric is inversely related. To overcome this drawback of precision and recall metric, F-measure measure is used. Equation (8) is the definition of F-measure which considers both precision and recall.

$$F - measure = \frac{recall \times precision}{(recall + precision)/2} \tag{8}$$



(a) Precision



(b) Recall

Fig. 11. Performance comparisons of our method, user-based Top-N, and item-based Top-N

A higher value of precision metric indicates that most of the recommendation is correct and a higher value of recall indicates that most of the similar music items are recommended. Also, we repeated the performance measure as varying the number of recommended music items, *N*. As seen in Fig. 11(a) (b), our method outperforms the conventional methods. For instance, at *N*=3, user-based Top-N and item-based Top-N have the precision of 0.37 and 0.52, while our method has a precision of 0.59.

From the analysis of the performances of user-based Top-N and item-based Top-N, their poor performances are due to the lack of sufficient rating information on the music items recommended. In other words, the target users have very few similar users who have rated music items. The item-based Top-N shows a slightly improved result than the user-based Top-N because the recommendation is made based on the music items that are similar to other music items the target user has liked. Also, it is possible to retain only a small subset of music items to provide good prediction quality and similarities between the music items that are relatively static. As the number of music items recommended increases, the performance is degraded. It is because less similar music items are recommended in the recommendation process. As seen in the figures, our method outperforms user-based Top-N and item-based Top-N because our method considers the semantic relations among music items, which can compensate the drawbacks of user-based Top-N and item-based Top-N addressed above by reflecting user's current psychological mind (DISC).

The application of DISC leads our method to achieve an improved result when compared to user-based Top-N and item-based Top-N. When the number of recommended products is low our method performs the best (i.e. the highest value for precision is 0.59 when N=3). When the number of recommended music items increases, we found that the inclusion of less similar music items to the recommendation list degrades the performance slightly.

Table 4 shows the F-measure of different methods for different top N recommendations. In general, F-measure depends on the value of corresponding precision and recall values of top N recommendations. The higher value of our method for F-measure implies that the proposed method can provide better recommendation than the other methods.

Table 4. F-measure for Top N recommendation

Met \ N	3	6	9	12	15	18	21	24	27	30
user-based Top-N	0.29114754	0.32142857	0.32219178	0.31473988	0.31058823	0.30626323	0.29468075	0.28776699	0.28910789	0.28106136
item-based Top-N	0.34666666	0.34532299	0.34177215	0.33416020	0.32353591	0.33062069	0.32117647	0.31569230	0.32597014	0.32571428
Our method	0.40644444	0.40524943	0.40459770	0.40578034	0.39439024	0.39873325	0.39917647	0.38925221	0.37897382	0.38181818

Also we evaluate user's satisfaction of our recommendation algorithm. As shown in Fig. 9, there are five radio buttons at the bottom of the interface of the mobile device that allows the user to evaluate his/her degree of satisfaction with the recommended multimedia content. Users are able to give a top score of 5 (very high satisfaction) and a bottom score of 1 (very high dissatisfaction).

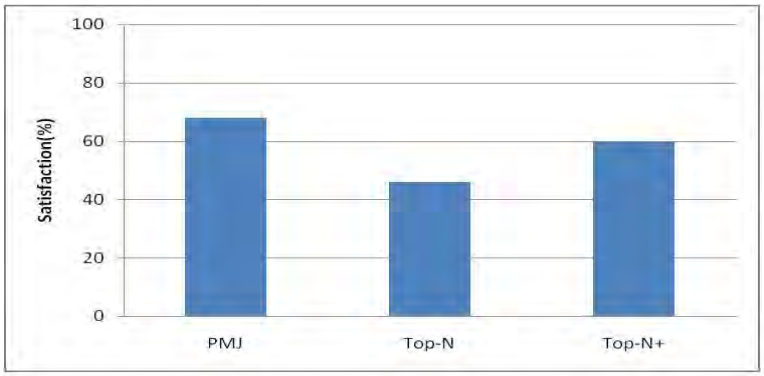


Fig. 12. Comparing Satisfaction with our method, user-based Top-N, and item-based Top-N

Fig. 12 is a graph of the average degree of satisfaction with our recommendation technique, user-based Top-N and item-based Top-N algorithm. According to the results of the experiment, the user-based Top-N method shows a level of satisfaction of 46%, whereas the item-based Top-N method shows a level of satisfaction of 60%. The PMJ shows the highest level of satisfaction at 68%.

7. Conclusion

This paper proposed the method for recommending multimedia content in a personal content manager. Our method can be used to solve a scalability problem and a limited resource problem of a server system and a mobile device, respectively. Main contributions of this paper are threefold. The first, we proposed a method to adopt and apply the DISC model verified in psychology field for the content recommendation. The second, we proposed an algorithm for reflecting dynamic environmental context to the recommendation. The third, we implemented a prototype system and showed that our method is reasonable for multimedia content recommendation through the performance evaluation. To the best of our knowledge, it is hard to find previous studies for applying user's psychology to multimedia content recommendation. However, our performance evaluation showed that the DISC model is closely related to the purchase pattern and the user's psychology is one of the useful factors for recommendation.

In our current version, the prototype system dealt with music content only. However, it will be extended and applied to various types of multimedia content.

References

1. Adomavicius, G. and A. Tuzhilin: Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering, 17(6), 734-749. (2005)
2. Ardissono L, Gena C, Torasso P, Bellifemine F, Chiarotto A, Difino A, Negro B: Personalized recommendation of TV programs. Lecture Notes in Artificial Intelligence 2829, 474-486. (2003)
3. Ardissono L, Portis F, Torasso P, Bellifemine F, Chiarotto A, Difino A: Architecture of a System for the Generation of Personalized Electronic Program Guides. In Proceedings of the Workshop on Personalization in Future. (2001)
4. Kaushal K, Srinivas G, David S, Jacquelyn M, John Z: A Multi-Agent TV Recommender. in Proc. the Workshop on Personalization in Future TV. (2001) http://www.di.unito.it/~liliana/UM01/kurapati.pdf. Accessed 9 June 2010
5. Ryu J, Kim M, Nam J, Kang K, Kim J: User Preference based Intelligent Program Guide. Journal of Korean Society of Broadcast Engineers 7(2), 153-167. (2002)
6. Sung H H: Digital Content Recommender on the Internet. IEEE Intelligent Systems 21(2), 70-77. (2006)
7. Zhiwen Y, Xingshe Z: TV3P: An Adaptive Assistant for Personalized TV. IEEE Transactions on Consumer Electronics 50(1), 393-399. (2004)
8. Upendra, S. and M. Pattie: Social Information Filtering: Algorithms for Automating "Word of Mouth". In Proceedings of the SIGCHI conference on Human factors in computing systems. (1995)
9. Mukund, D. and K. George: Item-Based Top-N Recommendation Algorithms. ACM Trans. Inf. Syst., 22(1), 143-177. (2004)
10. Greg L, Brent S, Jeremy Y: Amazon.com Recommendations Item-to-Item Collaborative Filtering. IEEE Internet Computing 7(1), 76-80. (2003)
11. Gui-Rong X, et al.: Scalable Collaborative Filtering Using Cluster-based Smoothing. In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, 114-121. (2005)
12. Xiaoyuan, S. and M.K. Taghi, A survey of collaborative filtering techniques. Advances in Artificial Intelligence, 2009, 1-19. (2009)
13. Cotter P, Smyth, B: PTV: Intelligent Personalized TV Guides. In Proceedings of the 7th National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, 957 – 964. (2000)
14. Lee W P, Yang T H: Personalizing Information Appliances: a Multi-agent Framework for TV Programme Recommendations. Expert Systems with Applications 25(3), 331-341. (2003)
15. Smyth B, Cotter P A: A Personalized TV Listings Service for the Digital TV Age. Knowledge-Based Systems, 13(2-3), 53-59. (2000)
16. Andrew, I.S., et al.: Methods and Metrics for Cold-Start Recommendations. In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, ACM: Tampere, Finland. (2002)

17. Basu, C., Hirsh, H., Cohen. W.: Recommendation as Classification: Using Social and Content-based Information in Recommendation. In Proceedings of the 15th National Conference on Artificial Intelligence (AAAI 98), Madison, WI. (1998)
18. Li, Y., Lu, L., Xuefeng, L.: A Hybrid Collaborative Filtering Method for Multiple-interests and Multiple-content Recommendation in E-Commerce. Expert Systems with Applications 28, 67–77. (2005)
19. Annie Chen: Context-Aware Collaborative Filtering System: Predicting the User's Preference in the Ubiquitous Computing Environment. Lecture Notes In Computer Science 3479, 244-253. (2005)
20. Kang D O, Lee H K, Ko E J, Kang K, Lee J: A Wearable Context Aware System for Ubiquitous Healthcare. In Proceedings of the 28th Annual International Conference of IEEE Engineering in Medicine and Biology Society, 5192-5195. (2006)
21. Mark S, Stanislav P, Johan K: Context-Aware Recommendations in the Mobile Tourist Application COMPASS. Lecture Notes In Computer Science 3137, 235-244. (2004)
22. Park W I, Park J H, Kim Y K, Kang J H: An Efficient Context-Aware Personalization Technique in Ubiquitous Environments. In Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication, ACM, (2010) pp 415-421.
23. Cosimo, P., T. Alexander, and G. Michele: Using Context to Improve Predictive Modeling of Customers in Personalization Applications. IEEE Trans. on Knowl. and Data Eng., 20(11), 1535-1549. (2008)
24. Kevin H, William H Selling: Yourself to Others. Pelican Publishing Company, USA. (2002)
25. Richard P B, Zeynep G C, Joseph R P: The Social Psychology of Consumer Behaviour (Applying Social Psychology), Open University Press, USA. (2002)
26. William M M: Emotions of Normal People (International Library of Psychology), Kegan Paul Trench Trubner & Co. Ltd., London. (2003)
27. XHTML 2.0, The eXtensible HyperText Markup Language, W3C Recommendation http://www.w3.org/TR/2006/WD-xhtml2-20060726. (2006)
28. Ziegler, C. N, Mcnee, S. M., Konstan, J. A., Lausen, G: Improving Recommendation Lists Through Topic Diversification. In Proceedings of the 14th Int'l Conference on World Wide Web(WWW), Chiba, Japan, 22-32. (2005)
29. Yang, Y., Liu, X.: A Re-Examination of Text Categorization Methods. In Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99), Berkeley, CA, 42–49. (1999)
30. Kowalski, G.: Information Retrieval Systems: Theory and implementation. Kluwer Academic publisher. (1997)

Won-Ik Park received the M.S. degree in Computer Engineering from Chungnam National University, Korea, in 2004. Currently he is a candidate for the Ph.D. in the Department of Computer Science & Engineering from Chungnam National University, Korea. His research interests include Mobile Computing, Multimedia Systems, Inference Systems, Artificial Intelligence, etc.

Sanggil Kang received the M.S. and Ph.D. degrees in Electrical Engineering from Columbia University and Syracuse University, USA in 1995 and 2002, respectively. He is currently an associate professor in the Department of Computer and Information Engineering at INHA University, Korea. His research interests include Semantic Web, Artificial Intelligence, Multimedia Systems, Computer Vision, Time Series, etc.

Young-Kuk Kim received the M.S. and Ph.D. degrees in Computer Science and Statistics from Seoul National University, Korea and Computer Science in University of Virginia, USA in 1987 and 1995, respectively. He is currently a professor in the Department of Computer Science & Engineering at Chungnam National University, Korea. His research interests include Mobile and Ubiquitous Information Systems, Real-Time, Mobile and Main-Memory Databases, etc.

Received: June 18, 2010; Accepted: August 8, 2011.

DOI 10.2298/CSIS100901038S

Representation of Texts in Structured Form

Mladen Stanojević[1] and Sanja Vraneš[1]

[1]University of Belgrade, Institute Mihajlo Pupin,
Volgina 15, 11060 Belgrade, Serbia
{mladen.stanojevic, sanja.vranes}@pupin.rs

Abstract. Although the existing knowledge representation techniques, ranging from the relational databases to the most recent Semantic web languages, are successfully applied in numerous practical applications, they are still unable to represent the information contained in text documents and web pages in structured form, suitable for productive text processing. Text files can represent text documents with no loss of information, however, this information is represented in an unstructured form. Various knowledge formalisms used in different phases of Natural Language Understanding, such as lexical, syntactic, semantic, pragmatic and discourse analysis, are still unable to represent texts in structured form with no loss of information. In this paper, we define the crucial requirements for structured text representation and then, we give a brief introduction to a representation technique that fulfills all these requirements, including the basic data types and learning techniques used to create, maintain and interpret the resulting representation formalism.

Keywords: structured representation, learning, text processing, natural language understanding, regular languages.

1. Introduction

Computers can process information efficiently, only if it is represented in a structured form. Natural language documents and web pages are the examples of the unstructured knowledge representation, so the problem is how to translate them automatically and represent the information contained in a structured form with no loss of information.

The importance of the structured data representation is obvious on an example of a relational database. Data in a relational database are represented in the structured form suitable for the automatic processing in various applications. We can dump this database into a text file that will contain the same information as the database itself and the created text file may be used to backup or transfer the database to another computer, but it is definitely not suitable for the automatic data processing. The databases are convenient means for structured data representation, and the focus of this paper is on a representation technique that will play a role of “a database for

texts", where the information found in texts will be represented in the structured form.

Different knowledge representation techniques, like conventional knowledge representation techniques (e.g. relational [1] and object-oriented [2] databases), Artificial Intelligence techniques [3], [4] (e.g. logic formalism, semantic nets, conceptual dependencies, frames, scripts, rules, etc.), or Semantic Web [5] ontology and schema languages (e.g. XOL [6], SHOE [7], OML [8], RDFS [9], DAML+OIL [10], OWL [11]) with enough expressive power to represent any kind of knowledge in structured form suitable for further computer processing, have been successfully applied to support knowledge processing in many different application domains.

However, it has been widely recognized in academic circles that neither of these techniques can be successfully applied in the representation of information found in natural language documents and web pages. They cannot be used to automatically translate various texts (natural language documents, web pages, etc.) into the structured form with no loss of information.

Although these knowledge representation techniques may look rather different, they actually all share the basic principle, which limits their representational ability. This basic representational principle is related to the way we perceive the world around us. We use named symbols to distinguish different phenomena and to capture their semantics.

We observe the world as it is composed of separate and distinct phenomena, objects, entities, which are mutually connected by a set of specific relations. All these phenomena, objects and entities may be more closely described using some features. Hereby, we commonly use names to describe the meaning of observed phenomena, objects, entities, features and relations.

Almost all existing knowledge representation techniques use naming to describe the meaning of represented knowledge. Instead of just representing the knowledge, all these techniques provide also the means for interpreting its meaning using names. Naming actually creates the limitations of the existing knowledge representation techniques. These knowledge representation techniques can be called "symbolic", because they represent the real world domains using simple and complex symbols and the corresponding relations between them. However, there is a knowledge that is not symbolic in its essence (e.g. information found in paragraphs, sections, documents or web pages in natural languages), which cannot be represented using symbolic knowledge representation techniques.

The proposed representation technique is completely equivalent to text files regarding the information represented by these two techniques. However, while text files represent this information in an unstructured form, the proposed technique represents the same information in a structured form suitable for automated text processing.

The first attempt to implement a novel technique able to translate texts into the structured form without loss of information resulted in the proposal of Hierarchical Semantic Form (HSF) and SOUL algorithm [12], while a more

advanced approach is applied in Natural language Implicit Meaning Formalization and Abstraction (NIMFA) [13].

The organization of the paper is the following: Section 2 presents some related work, which outlined the ideas beyond symbolic knowledge representation knowledge representation formalisms used to represent different kind of knowledge in Natural Language Understanding, Section 3 describes the main characteristics of natural languages and the requirements for structured representation of texts, Section 4 gives the basic insights into the implementation of the requirements for the representation of texts in structured form, while Section 5 provides some conclusions.

2. Related Work

There is an acute lack of references related to the structured representation of natural language documents. However, there are some papers that outline a new way of thinking, which is beyond symbolic knowledge representation.

The so-called “radical connectionism” [14] claims that the natural language is not used as representational, but rather as communicational medium. The modification of the “localist” approach of the “connectionist” model [15] could be used as a starting point for the implementations of ideas of radical connectionism. One implementation of a knowledge representation not based on symbols is represented by Hierarchical Temporal Memory (HTM) [16], used for image processing, while another solution [17], used in text processing, relies on the Hopfield-like neural networks.

However, these approaches failed in defining clear requirements for structured representation of natural language documents, which will not be based on symbols. In structured representation of natural language documents, the tasks of representing knowledge is clearly separated from the task of interpreting the meaning of the represented knowledge, which provides the basis for overcoming the limitations of symbolic knowledge representation.

The representation technique proposed in this paper should facilitate text processing and more precisely Natural Language Understanding (NLU), which is actually not in the focus of this paper, hence, we will only pass briefly through the knowledge representation formalisms used to represent different forms of knowledge relevant for NLU [18] (e.g. morphological, syntactic, semantic, pragmatic and discourse knowledge).

The morphological knowledge used in lexical analysis is usually represented by Finite State Automata (FSA) and Finite State Transducers (FSTs) to implement the electronic dictionaries for the given natural languages. The morphological knowledge used in lexical analysis corresponds to the structure of words.

In syntactical analysis various classical, statistical and connectionist approaches are used, whereby classical and statistical approaches are based on the corresponding grammars. Among the most popular grammars are the

dependency grammars, which, as a result of parsing a statement, produce the corresponding dependency tree to reflect the syntactic structure of the given statement. The grammars used in syntactic analysis are language dependent and subjective, because even for the same language different linguists may propose different grammars.

The semantic knowledge, as a result of semantic analysis of a sentence, can be represented in logical form (e.g. first order predicate calculus - FOPC). However, the statements in FOPC can represent the meaning of natural language sentences only, whereby some information considered semantically irrelevant may be lost. Moreover, FOPC statements are not structured.

The discourse knowledge is used in different discourse structuring theories like the theory of segmentation, attention shift and hierarchical inclusion of topic-related discourse segments [18] and Rhetorical Structure Theory [19]. However, the structures used to represent the discourse knowledge are only temporarily used to interpret correctly the meaning of sentences in the given context. There are also different discourse representation models [20] usually implemented using the specific ontologies, but all these models are manually built.

As we can see neither of the above described knowledge representation formalism used in NLU can be used to automatically translate texts into the structured form with no loss of information. They are either too specific, enabling the representation of words or phrases or sentences, or are temporal by their nature and not designed for persistent storage of texts in the structured form.

Another way to represent information found in a text in the structured form is to use some information extraction technique, which combines an NLU technique with a suitable knowledge representation technique [21], [22]. However, information extraction techniques also have some drawbacks: 1) they cannot be used to represent the complete information found in a text, but only a fraction of it; 2) information extraction systems are language dependent (they use language dependent dictionaries and grammars); 3) for each new subset of data that should be extracted from a text, a new information extraction system must be developed.

3. Requirements for Representation of Texts in Structured Form

Various natural language documents and web pages are represented in the plain text form and it seems that there is no apparent structure behind it. However, there is an intrinsic structure behind any natural language. Unlike the syntactic structures, which are subjective and language dependent, this structure is objective and the same for all languages.

This structure is composed of letters, syllables, words, phrases, sentences, paragraphs, etc. Hereby, syllables are composed of letters, words are

composed of syllables, phrases are composed of words and so on. Another important characteristic of natural languages is their sequentiality. Each structural element at the higher level is composed of a sequence of structural elements at the lower level.

Having in mind these simple observations, we can formulate two basic requirements for structured representation of text documents:

1. Provide support for context representation, i.e. provide means for determining the exact position of each structure element regarding the hierarchical level and place in the corresponding sequence(s).
2. Provide unique representation for each structure element in different contexts.

Strictly speaking, the context of a structure element at the given hierarchical level is defined by its predecessor and successor structure element, where each structure element must be uniquely represented.

Why these two requirements are so important? Because they enable an efficient search of natural language documents. Each word, each phrase, each sentence, etc. would be uniquely represented in a hierarchical network that could represent thousands of natural language documents or web pages. We would be able to efficiently identify all contexts in which the given word or phrase appear in, i.e. all sentences, paragraphs, sections and documents. The search through the hierarchically organized structure must be much faster compared to the sequential search of natural language documents represented in an unstructured form.

The above-described requirements are also in line with the fact well known in linguistics that any limited language is at the same time a regular language. Since all natural languages are limited, all text documents can be represented using a deterministic finite state machine. A Finite State Automaton (FSA) [18] in lexical analysis is usually used to locate morpho-syntactic patterns in corpora.

A common way to represent text documents is to use files, which can be observed as Finite-State Automata, where each text character is followed by the next character until the end-of-file mark is encountered leading to the terminating state. However, for information to be processed by computers effectively, it must be represented in a structured form. Since files represent text documents in plain form, they are not particularly useful for automatic processing.

Recursive Transition Networks (RTNs) [18] are used in syntactic analysis to construct various grammars. However, RTN can be also used to introduce structure to FSA, where a part of an existing FSA can be named to represent a simple graph. The only problem with RTN is that it is manually constructed where all constituent graphs must be named.

Eventually, both requirements for structured text representation are satisfied by RTN, but in its present form it is useless, because it must be constructed manually.

4. Technique for Representing Texts in Structured Form

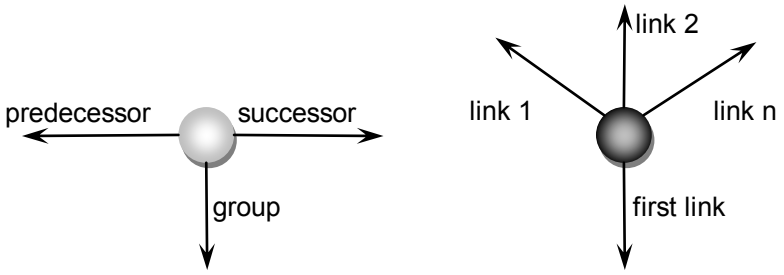
In this section, we will describe a technique for representing texts in the structured form with its basic data types and "learning by repetition" algorithm that facilitates the automatic creation and maintenance of the corresponding hierarchical network. This hierarchical network is composed of two types of nodes where **groups** are used to uniquely represent the natural language structures (e.g. words, phrases, sentences, groups of sentences, paragraphs, groups of paragraphs, documents), while **links** are used to represent these natural language structures within different contexts (e.g. words in phrases, phrases in sentences, sentences in paragraphs, etc.). We will, then, give a simple example of the hierarchical network created using the described representation technique, provide an algorithm for translating texts into structured form and finally, we will just briefly describe how the hierarchical network can be interpreted to find its meaning.

4.1. Basic Data Types

Any technique suitable for the representation of texts in the structured form must satisfy the above described requirements related to context representation and unique representation of structure elements. The network, similar to RTN, which will represent texts in structured form can be built using two basic data types corresponding to two defined requirements:

Link data type (Fig. 1.a) enables the context representation and implements the ternary, sequential relation between the previous, current and successive structure element. Since each word and any other natural language structure may appear in different contexts (e.g. the same word may appear in different phrases or sentences), links are used to represent this natural language structure in all these contexts. It contains pointers to its predecessor and successor, but also to the structure element it represents in the given context. The successor of the last link in the sequence points to the group which represents this sequence. The link data type corresponds to states in RTN.

Group data type (Fig. 1.b) supports unique representation of all structure elements (characters, syllables, words, phrases, sentences, etc.). This means that any natural language structure is represented only once in a hierarchical structure that may represent many natural language documents. Instead of repeating the same structure element in different contexts, we use links to represent the corresponding group in these contexts. Each group contains a pointer to the first link of the sequence it represents, but also pointers to all the links that represent the corresponding structure element in different contexts. The group data type corresponds to transitions in RTN.



a) link b) group

Fig. 1. Basic data types

Any natural language document has inherent, hierarchical structure, which may not be readily visible when we read it, composed of characters at the lowest levels, then words, phrases, sentences, paragraphs, etc. Since the main characteristic of all natural languages is their sequentiality, we may say that any natural language document can be represented as a hierarchical structure comprised of sequences at different levels of abstraction. The hierarchical structure used to represent texts in structured form contains groups representing characters at the lowest level, followed by syllables which represent sequences of letters, words as sequences of syllables and letters, phrases as sequences of other phrases and words, sentences as sequences of phrases and words and so on. Hereby, sequences are represented using links.

Formally, the information in structured text representation can be represented by the space S defined by the triple of groups G, links L and sequences Q (composed of links from L):

$$S = \{G, L, Q\},\ g_i \in G, i = 1, r,\ l_j \in L, j = 1, s,\ q_k \in Q, k = 1, t \tag{1}$$

Initially, G contains only groups corresponding to letters, L contains only links corresponding to these atomic groups and Q is an empty set of sequences.

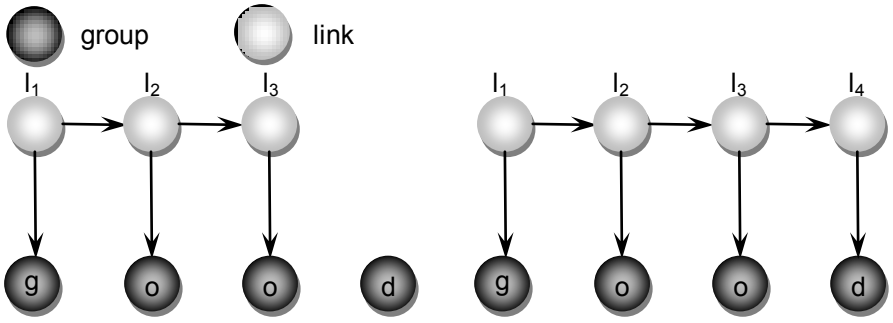
Structured text representation follows the two basic principles in knowledge representation, the principle of locality (context representation) and the principle of unique representation.

Principle of locality defines the transition T from the link l_t, which is the last link in the subsequence q_i, to the link l_u, when group g_c belonging to the same hierarchical level appears at the end of subsequence q_i:

$$l_u = T(l_t, g_c),\ q_p = q_i l_u,\ l_u \to g_c,\ q_i, q_p \in Q,\ g_c \in G,\ l_t, l_u \in L \tag{2}$$

The link l_u represents the group g_c in the subsequence q_p, which extends the subsequence q_i. If g_c is a new group, or link l_u does not exist, then the new link l_u must be created. The principle of locality enables learning of new sequences.

For instance, suppose that the system learns the word “good” and that it already represented the sequence “goo” (Fig. 2.a) and that it has to add now “d” to complete the sequence (Fig. 2.b). In our example, $l_u = l_4$, $l_t = l_3$, g_c corresponds to a group representing the letter “d”, q_i. corresponds to a sequence of letters “goo” represented by links l_1, l_2 and l_3, while q_p. corresponds to a sequence of letters “good” represented by links l_1, l_2, l_3 and l_4.



a) unfinished sequence b) finished sequence

Fig. 2. Principle of locality

This principle is related to the representation of sequences at different levels of hierarchy (words, phrases, sentences, paragraphs, etc.). It basically states that paragraphs are composed of sentences and not of letters or words.

Principle of unique representation states that each subsequence (q_x) that repeats in two different sequences (contexts, q_i, q_j) must be uniquely represented by the corresponding group (g_u):

$$
\begin{aligned}
& g_s \to q_i, q_i = q_a q_x q_b \,,\; g_s \in G \,,\; q_a, q_b, q_i, q_s, q_x \in Q \\
& g_t \to q_j, q_j = q_c q_x q_d \,,\; g_t \in G \,,\; q_c, q_d, q_j, q_t, q_x \in Q \\
& g_u \to q_x \,,\; l_p, l_q \to g_u \,,\; l_p, l_q \in L \,,\; g_u \in G \,,\; q_x \in Q \\
& g_s \to q_i, q_i = q_a l_p q_b \\
& g_t \to q_j, q_j = q_c l_q q_d
\end{aligned}
\tag{3}
$$

Hereby, subsequences q_a or q_b, q_c or q_d may be empty, i.e. they may contain no links. When a subsequence q_x repeats in two sequences (q_i, q_j), a new group g_u will be created corresponding to this subsequence, as well as two new links (l_p, l_q) representing this subsequence in two different contexts (q_i, q_j). This is an example of self-organization of the space S, which allows an automatic identification of semantic concepts, structures and relations.

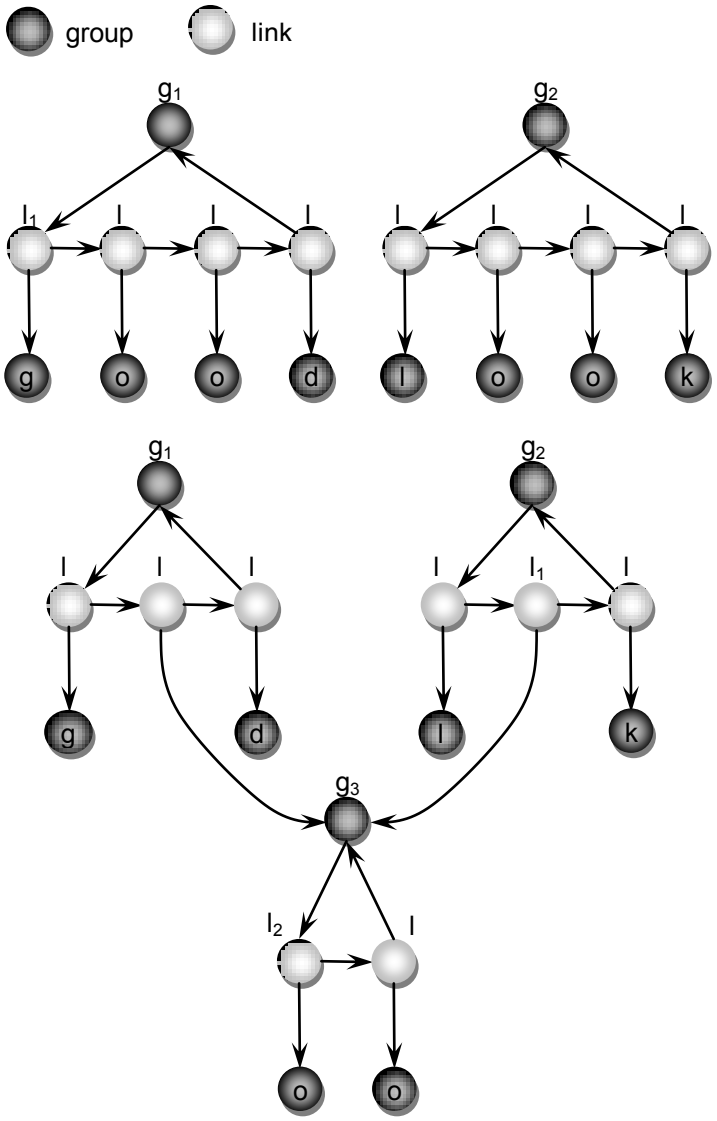


Fig. 3. Principle of unique representation

As an example of principle of unique representation we will use two words “good” and “look”. We can immediately notice that these words share the

same subsequence of letters “oo”, which should be uniquely represented (Fig. 3).

In our notation group $g_s = g_1$ uniquely represents the word “good” and the corresponding sequence of letters q_i, group $g_t = g_2$ uniquely represents the word “look” and the corresponding sequence of letters q_j where $q_a = l_1$ corresponds to letter “g”, $q_x = l_2 l_3$ to sequence of letters “oo” in word “good” and $q_x = l_6 l_7$ to the same sequence of letters in word “look”, $q_b = l_4$ to letter “d”, $q_c = l_5$ to letter “l” and $q_d = l_8$ to letter “k”. Since the same sequence of letter “oo” appears in both words, we should represent this sequence uniquely. Therefore, we create a group $g_u = g_3$ to uniquely represent this sequence and then a link $l_p = l_9$ to represent the letters “oo” in the word “good” and a link $l_q = l_{10}$ to represent the same sequence of letters in the word “look”.

4.2. Learning by Repetition

“Learning by repetition” is one of the basic forms of learning inherent to human beings. We cannot observe different phenomena unless they are repeated. In the same way, we learn to speak languages. “Learning by repetition” enables us to learn the structure, i.e. words, phrases, sentences, etc. of any natural language. Notice that “learning by repetition” doesn’t allow us to understand the meaning of language structures, but only to distinguish them.

However, when linguists speak about the structure, they usually refer to syntactic structure of a natural language statement. A dependency tree representing the structure of a natural language statement can be created by parsing this statement using the corresponding dependency grammar. “Learning by repetition” cannot be used to create a dependency tree, because it is not related to any dependency grammar and syntactic structures. The syntactic structures defined by dependency grammars are language dependent and subjective in their essence, whereas the structures (words, phrases, etc.) created by “learning by repetition” are language independent.

“Learning by repetition” facilitates unique structure representation and can be defined in the following way: if two contexts share the same subsequence, this subsequence then represents a new structure element which is shared by two contexts. For this new structure element, a new group will be created and two new links will represent this structure element in the given contexts. Strictly speaking, this definition is slightly different from the definition used in psychology where the same structure should be repeated many times to be memorized.

In practice, one of these two contexts will be a referential context, which has been already created, while the second one is a new one and has not yet been represented. Basically, we have three possible cases when “learning by repetition” can take place: the repeated subsequence appears at the beginning of the referential context (Fig 4.a), it occurs in the middle of the

referential context (Fig. 4.b), or it takes place at the end of the referential context (Fig. 5.c).

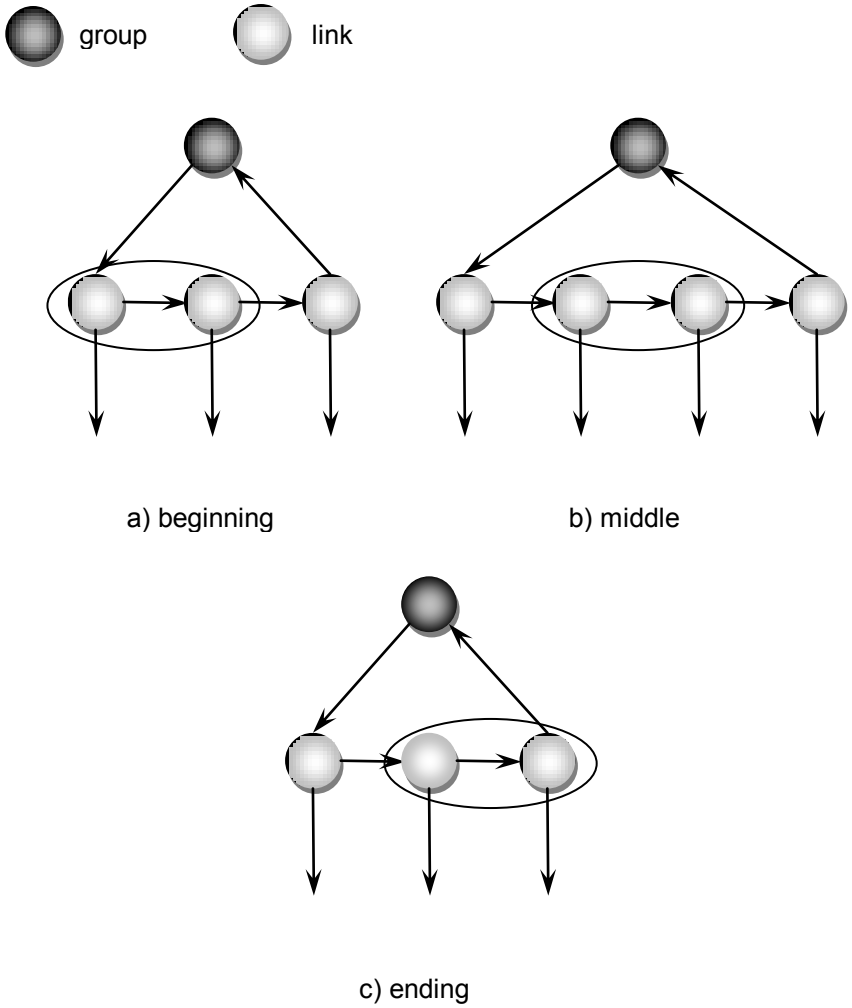


Fig. 4. Position of repeated subsequence

As an example of “learning by repetition”, we will take two sentences:

1. “Bill was in the school”
2. “Bill is in the school”.

Suppose that we have already entered all words and that we have presented the first sentence. We will have a hierarchical network as presented in Fig. 5. Notice that groups represented at the bottom of the hierarchical representation are actually not named and that these labels are used only to simplify the diagram and hide the part of structure representing the corresponding words.

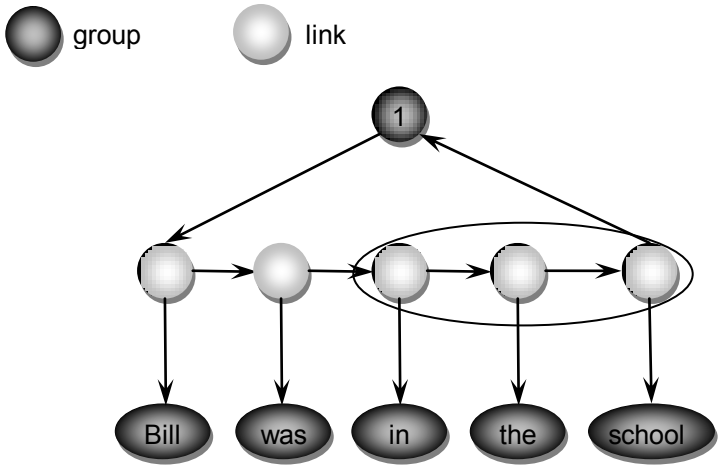


Fig. 5. Hierarchical representation of the first sentence

When the second sentence is processed by the structured text representation technique, it will discover that the ending subsequence “in the school” is repeated (Fig. 5) and reorganize the hierarchical network to represent correctly the second statement (Fig. 6).

The representation technique presented in this paper enables the creation of the self-organizing hierarchical network, which changes as new texts are fed to it. It reuses the part of information that has been already represented and adds the new one. Unlike symbolic knowledge representation technique, this knowledge representation technique is able to automatically translate any natural language text into the corresponding hierarchical structure and vice versa with no loss of information.

4.3. Example

As an example of structured text representation, we will use the same two simple sentences to represent the context correctly:

1. “Cows eat grass.”
2. “Cats eat mice.”

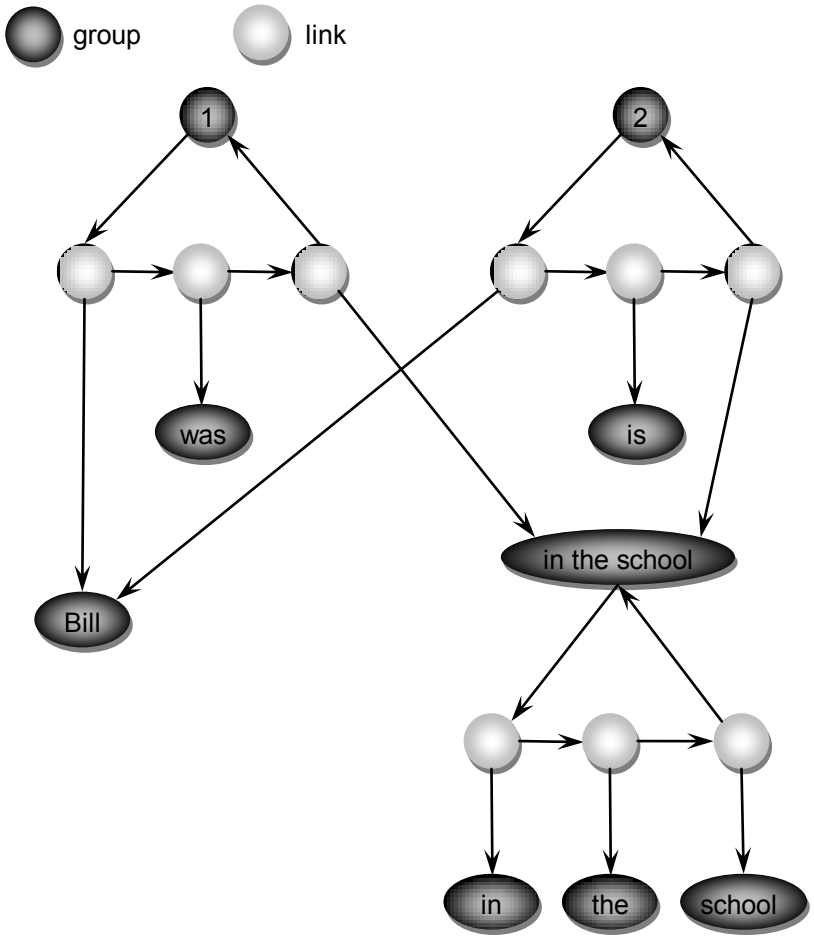


Fig. 6. Learning by repetition

At the beginning, the empty hierarchical structure is comprised only of groups representing single characters. There are no links, because no text is fed to it yet.

The learning process begins with feeding the single words. This is actually quite similar to the way babies are learning to talk. After we have fed all the words from our simple statements, we will get hierarchical network as presented in Fig. 7.

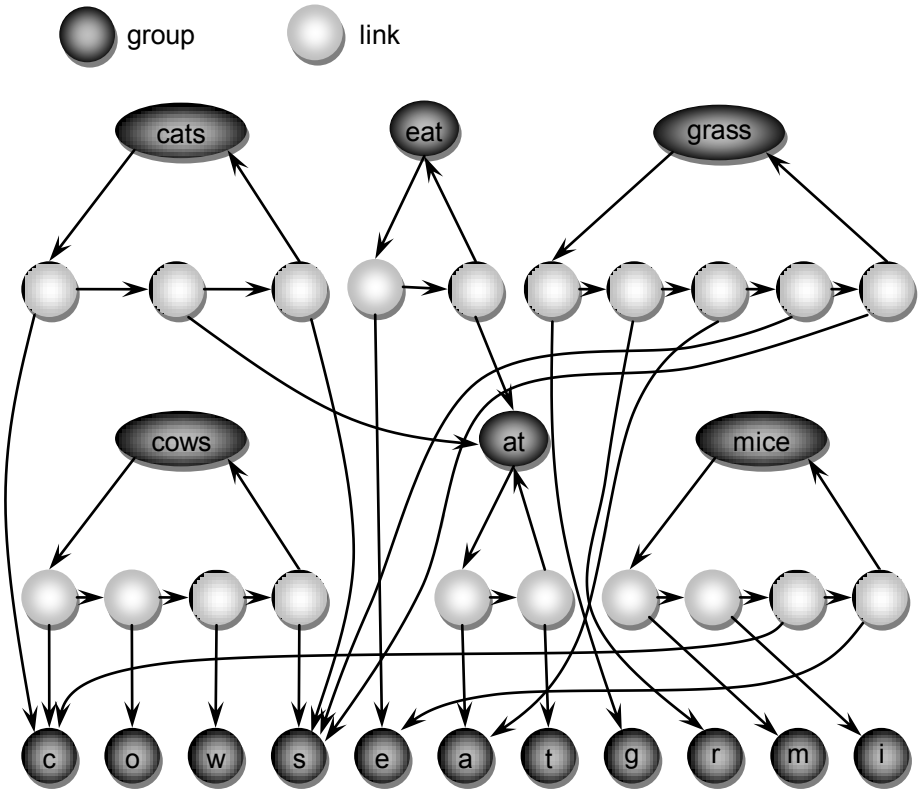


Fig. 7. Representation of single words

After both sentences are processed the hierarchical structure will have the form as presented in Fig. 8.

The same representation technique can be used to represent sentences, paragraphs, sections, etc. represented as the corresponding groups in Fig. 9.

4.4. Translation of Texts into Structured Form

Any natural language text can be translated automatically into the structured form composed of groups representing uniquely syllables, words, phrases, sentences, groups of sentences, paragraphs and group of paragraphs using the algorithm given below.

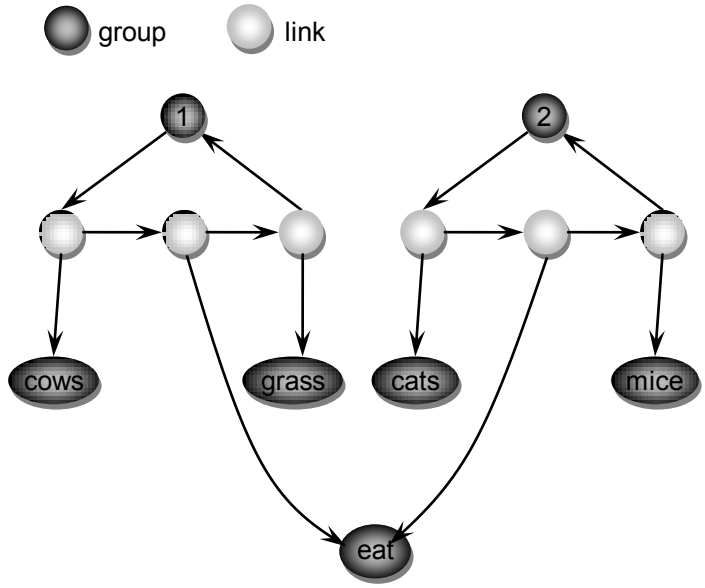


Fig. 8. Correct context representation

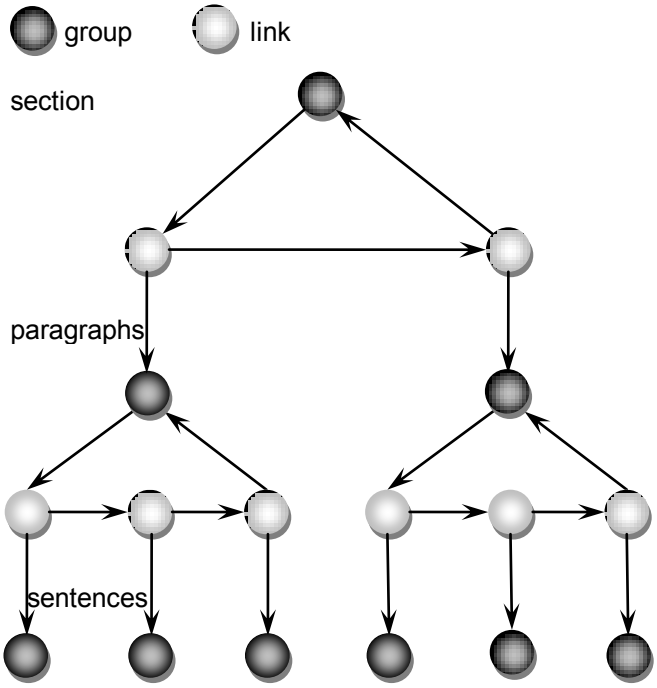


Fig. 9. Representation of sentences, paragraphs and section

```
createStructuredTextForm(fileName, currDocumentGroup)
{
        paragraphGroups = new DoubleLinkedList();
        paragraphsGroups = new DoubleLinkedList();
        textFile = new File();
        textFile.open(fileName);
        currPos = 0;
        currChar = textFile.readAt(currPos);
        // while not end of file
        while (currChar != eof)
        {
                sentenceGroups = new DoubleLinkedList();
                sentencesGroups = new DoubleLinkedList();
                // while not end of paragraph
                while (currChar != eop)
                {
                        charGroups = new DoubleLinkedList();
                        constructGroups = new DoubleLinkedList();
                        // while not end of sentence
                        while (currChar != eos)
                        {
                                convertCharToGroup(currChar, charGroup);
                                charGroups.addTail(charGroup);
                                currPos++;
                                currChar = textFile.readAt(currPos);
                        }
                        pos = charGroups.getHeadPosition();
                        while (pos != null)
                        {
                                // Identify existing syllables, words or
                                // phrases
                                identifyLongestSequence(charGroups, pos,
                                        nextPos, currGroup);
                                constructGroups.addTail(currGroup);
                                pos = nextPos;
                        }
                        if (constructGroups.Count() > 1)
                                // Create a new sentence
                                currSentenceGroup =
                                        createGroup(counstructGroups);
                        else
                                // The sentence is already defined
                                currSentenceGroup = currGroup;
                        sentenceGroups.addTail(currSentenceGroup);
                        delete charGroups;
                        delete constructGroups;
                }
```

```
                pos = sentenceGroups.getHeadPosition();
                while (pos != null)
                {
                        // Identify existing groups of sentences
                        identifyLongestSequence(sentenceGroups, pos,
                                nextPos, currGroup);
                        sentencesGroups.addTail(currGroup);
                        pos = nextPos;
                }
                if (sentencesGroups.Count() > 1)
                        // Create new paragraph
                        currParagraphGroup =
                                createGroup(sentencesGroups);
                else
                        // The paragraph is already defined
                        currParagraphGroup = currGroup;
                paragraphGroups.addTail(currParagrapghGroup);
                delete sentenceGroups;
                delete sentencesGroups;
        }
        pos = paragraphGroups.getHeadPosition();
        while (pos != null)
        {
                // Identify existing groups of paragraphs
                identifyLongestSequence(paragraphGroups, pos,
                        nextPos, currGroup);
                paragraphsGroups.addTail(currGroup);
                pos = nextPos;
        }
        if (paragraphsGroups.Count() > 1)
                // Create new document
                currDocumentGroup = createGroup(paragraphsGroups);
        else
                // The same document is already entered
                currDocumentGroup = currGroup;
        delete paragraphGroups;
        delete paragraphsGroups;
        textFile.close();
        delete textFile;
  }
```

The algorithm is described using the `createStructuredTextForm` procedure given in a pseudo-code form. This procedure has two arguments: `filename` is an input argument representing the name of the text file; `currDocumentGroup` is an output argument representing the text file in a

structured form. This hierarchically organized, structured form is composed of groups of paragraphs, paragraphs, groups of sentences, sentences, phrases, words, syllables and characters at the bottom of this hierarchy. Hereby, each of these natural language structures is uniquely represented for all text files that may be translated using the given procedure.

The procedure divides the text documents into sentences and paragraphs. It then translates one by one character found in a sentence into character groups (`convertCharToGroup`) and adds these character groups to the `charGroups` list.

The `identifyLongestSequence` procedure is used to implement the principle of unique representation based on learning by repetition. It has four arguments: the first one is a double-linked list of groups, the `pos` argument is the position in this list from where the procedure starts to identify the existing sequence, the `nextPos` argument is the position in the list where we begin the next search and the `currGroup` argument is a group representing the longest found sequence. Actually, this procedure implements three possible cases:

The group found at `pos` in linked list is not followed by the next group in the same list in any previously created sequence. The procedure returns `nextPos` as the position of the next group in the list and `currGroup` is the group found at `pos`.

All groups found in a double-linked list are already defined as a sequence represented by a single super group. The procedure returns `null` as the value of `nextPos` and the super group for `currGroup`.

Learning by repetition is applied and for the recognized subsequence of groups a new group is created. The `nextPos` points to the first group after the recognized subsequence in the linked list and the new group is returned as `currGroup`.

The function `createGroup` implements the principle of locality. The double-linked list containing groups is the only input argument and the function returns the super group representing uniquely the sequence of groups contained in the given list. For the contained groups the function first creates the corresponding links, then, it connects these links to represent a new sequence and finally creates a super group that uniquely represents the newly created sequence.

The hierarchical organization representing natural language texts serves as “a database for texts”, because it enables a structured representation of texts with “indexes” for each word, phrase, sentence, group of sentences, paragraph and group of paragraphs. However, unlike ordinary databases which require human experts to design them and create indexes manually, this “database for texts” together with the corresponding “indexes” is automatically created, with no help from human experts.

Knowing that the algorithm easily determines the repeated sentences, groups of sentences, paragraphs and groups of paragraphs, one obvious

application would be the determining of originality of text documents. Not only can the algorithm identify the repeated parts of texts, but it can also easily determine the provenance of copied parts. The other approaches sequentially compare the given text with the other texts, which can be very time-consuming, especially when there are many texts to check.

Another very useful application of the algorithm would be in the keyword search. The present day search engines provide many unproductive hits, because they search for keywords in the whole document, whereas they should try to locate the sentences or paragraphs containing these keywords. The algorithm explicitly stores the information about the context (sentences, paragraphs) and since all words and phrases are indexed in all encompassing natural language structures (sentences, paragraphs, documents) it is easy to find the given keywords in any desired context in all represented documents.

4.5. Interpretation of Meaning

Symbolic knowledge representation techniques define the meaning of represented knowledge using names, so they do not have to interpret the meaning.

On the other hand, structured text representation technique does not use the names to describe the meaning; hence, the hierarchical structures used to represent plain texts are not comprehensible to humans. These hierarchical structures have to be interpreted somehow to extract the meaning from them.

However, from the practical point of view, it seems that the emphasis should not be on the meaning but on the closely related issue such as relevance. In semantic knowledge representation, the role of a domain expert is very important, because he/she decides what objects, relations, features, etc. are relevant for the given domain of application. This relevance is actually defined having in mind the application domain.

It seems that human beings determine the relevance of things based on the appearance of differences in similar contexts. For instance, in distinguishing the twin brothers, we do not rely on similarities in their appearance, but on the small differences we can notice.

In case on natural languages, our attention is also attracted by differences. If we observe the phrases:

1. “Good morning”
2. “Good afternoon”
3. “Good evening”

we will immediately notice that the first word “Good” is repeated in all of them, but also that the second word is always different. Since it spots the difference in the similar contexts, our brain considers this difference as important and generalizes the three different words appearing at the end of these three phrases. This generalization actually gives the importance to

things and represents the basis of semantics. These three words may be taken as instances of a newly discovered, simple semantic category.

Since structured text representation requires the representation of context, the corresponding techniques should be able to automatically discover semantic categories in the given contexts. To do this, these techniques should support two types of learning: “learning by generalization” and “learning by specialization”.

“Learning by generalization” may be defined as follows: when two similar contexts differ from each other only by two different structures found in the same place in these contexts, then these two structures can be generalized. A new semantic category is discovered and two structures represent the instances of this newly discovered category.

“Learning by specialization” may be similarly defined: when a context contains a structure which appears at the same place as the structures that have been already generalized in similar contexts, this new structure can be considered as a new instance of the same semantic category.

In our example with three phrases, “learning by generalization” may be applied on the first two phrases, where words “morning” and “afternoon” appear at the same place in similar contexts and therefore, may be generalized to create a new semantic category. When the third phrase is considered, “learning by specialization” may be applied, because the word “evening” appears at the same place as the generalized words “morning” and “afternoon”, so, the word “evening” may be considered as a new instance of the same semantic category.

We can force “learning by generalization and by specialization” by using the explicit definitions and this is what we usually think when we speak about semantics:

1. “John is a boy”
2. “Bill is a boy”
3. “Tom is a boy”

The phrase “is a boy” is repeated in these three phrases, while “John”, “Bill” and “Tom” are instances of a semantic category. However, in the proposed knowledge representation techniques, this semantic category is actually not named and is not even defined as a separate structure. Only the structures corresponding to words “John”, “Bill” and “Tom” are marked as instances of a semantic category and when we ask a question:

“What is John”

we will be able to find the answer:

“John is a boy”

“Learning by generalization and by specialization” can be used to identify simple semantic categories and their instances. Instances of simple semantic

categories are words and phrases. The process of discovering simple semantic categories is actually very similar to the process of identifying symbols in symbolic knowledge representation.

However, the processing power of the proposed representation techniques should go beyond symbolic processing. These techniques should also be able to process sentences, paragraphs, sections, documents, web pages, etc.

The instances of simple semantic categories take part in more complex structures like complex phrases, sentences, paragraphs, etc. thus creating complex semantic categories or patterns. These complex semantic categories could be used to recognize natural language commands or to find some information or documents. However, the process of interpreting the meaning of represented texts is out of the scope of this paper. More information about the possible implementation of this process can be found in [13].

5. Conclusions

Although it seems that the existing knowledge representation techniques do not have much in common, almost all of them can be described as symbolic techniques. Actually, they are all designed to represent symbols, i.e. clearly separated entities (objects, phenomena) with defined features and relations that are relevant in the given domain of application. They all use names to describe the meaning of represented knowledge. So, these techniques besides providing means for knowledge representation also provide means for the interpretation of meaning. They facilitate symbolic knowledge representation and symbolic processing.

However, the symbolic knowledge representation techniques simply do not have the necessary representational power to represent texts in structured form. They do not provide the means to represent natural language structures, which are hierarchical and sequential by their nature, nor the means to process texts effectively.

Different knowledge representation formalisms used in text processing to represent various kinds of knowledge like morphological, syntactic, semantic, pragmatic or discourse are also not suitable for structured text representation. They are actually designed to represent only a specific kind of knowledge related to word structure, syntactic structure, meaning of words or discourse context.

However, it is well known in linguistics that any limited language is at the same time a regular language. So, a Finite State Automaton could be used to represent any text document, and this is exactly how texts are represented in files. The problem with text files as representation formalism is that they are not structured and thus not convenient for automatic text processing. Recursion Transition Networks (RTN) can be used to structure a graph represented by a Finite State Automaton, but the corresponding sub-graphs

must be named and that's why standard RTNs are not suitable for structured text representation.

In this paper, we have defined a novel technique for structured text representation which resembles the RPNs, but can be used to automatically translate texts into the structured form. It can be viewed as "a database for texts", where unlike the relational databases, which must be designed, this "database" is automatically created from the texts.

We have defined first two requirements that must be fulfilled by a technique to be able to represent texts in the structured form. The first requirement is related to the accurate context representation, i.e. the sequential order of natural language structures that comprise a more complex structure, while the second requirement is related to the unique representation of natural language structures in different contexts.

We have then defined two data types corresponding to two requirements for structured text representation: link data type (corresponding to RPN states) is used to satisfy context representation requirement, while group data type (corresponding to RPN transitions) satisfies the requirement for the unique structure representation.

The hierarchical structure representing text documents is created using groups and links and a special form of learning, which we call "learning by repetition". "Learning by repetition" enables learning the structure of natural languages, by identifying the repeated subsequences of structure elements. It facilitates an automatic translation of any natural language document into the structured form and vice versa with no loss of information. The created structure is self-organizing and changing as new knowledge is fed to it, whereby the old knowledge is reused and the new one is added.

Two other types of learning: "learning by generalization" and "learning by specialization" enable the interpretation of the represented knowledge. "Learning by generalization" supports the discovery of new semantic categories, while "learning by specialization" enables the definition of new instances of the existing semantic categories. These two types of learning facilitate the definition of simple semantic categories and the corresponding instances represented by words and phrases.

Complex semantic categories or patterns are represented by complex natural language structures composed of instances of simple and complex semantic categories. Complex semantic categories enable text processing needed for the understanding of natural language commands or finding the necessary complex information based on natural language queries.

Text files represent basically the same information as the proposed representation technique, but in an unstructured form. In the proposed technique information is structured in the way that each natural language structure (word, phrase, sentence, paragraph, etc.) is uniquely represented in all contexts in which it appears. The created structure facilitates an easy identification of all contexts in which some natural language structure appears giving rise to an efficient text processing and many practical applications.

There are many possible practical applications of structured text representation in areas like continuous speech recognition, question answering systems based on natural language queries, information retrieval, user interfaces based on natural language commands, machine translation, etc.

Continuous speech recognition systems are usually based on phonetic and phonological knowledge and the representation of contexts can enhance the precision of such systems by providing all candidate words that can succeed the last recognized word. The use of word context could significantly reduce the set of candidate words and considerably increase the probability that the correct word will be recognized.

As we all know, standard search engines based on keywords are not very useful when we use natural language queries. We usually get many unproductive hits. There are two reasons for such a performance: 1) search engines do not take care about the context; 2) they do not use semantic categories to abstract the complexities of natural language. All search engines implicitly take a document or web page as a context in which keywords are searched (reason for bad precision rate). However, sometimes we expect that these keywords must be found in the same sentence or the same paragraph. Standard search engines have poor recall rate when using natural language queries because the same thing can be said in many different ways. Semantic categories are coping efficiently with the richness of a natural language, but they are not supported in standard search engines. The structured text representation facilitate context representation and semantic categories, hence, it could be used to implement an efficient question answering system based on natural language queries.

Information retrieval is usually also based on keyword search, therefore, the same limitations hold as for search engines. Information retrieval based on structured text representation can improve the recall and precision rate again by using the context representation and semantic categories. For instance, instead of using combinations of keywords, patent attorneys could easily find all semantically correlated patents using the patent they would like to check.

Natural language-based user interfaces could be easily built using the structured text representation and semantic categories to cover many possible ways how humans can express natural language commands.

Machine translation could also benefit from the use of structured text representation technique, because parallel text corpora in different languages could be fed to it and the same group could then be used to represent the same word, phrase, sentence, paragraph, etc. in different languages. Thus, it would be possible to easily identify the highest level of translation (be it a paragraph, sentence, phrase or word) of a text based on the parallel text corpora represented in structured form.

The structured text representation technique presented in this text has been already successfully applied in question answering systems based on natural language queries. It was implemented first in a prototype system [12] that provides information about flight timetable for the largest European

airlines, while the second implementation was in a prototype web portal [13] providing information about flights, football matches and weather forecasts.

Acknowledgement. This work has been partially supported by the EU Seventh Framework Programme project LOD2 (Pr. No.: 257943) and partly by the Ministry of Education and Science of the Republic of Serbia (Pr. No.: TR-32010).

References

1. Date, C.J.: Database in Depth: Relational Theory for Practitioners. Sebastopol, CA: O'Reilly Media, Inc. (2005).
2. Russell, C. et al: The Object Data Standard: ODMG 3.0. San Francisco, CA: Morgan Kaufmann. (2000).
3. Sowa, J.: Knowledge Representation: Logical, Philosophical, and Computational Foundations. Pacific Grove, CA: Brooks/Cole Publishing Co. (2000).
4. Vraneš, S., Stanojević, M.: Prolog/Rex - A Way to Extend Prolog for Better Knowledge Representation. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 6, No. 1, 22-37. (1994).
5. Fensel, D., Hendler, J.A., Lieberman, H., Wahlster, W. (Eds.): Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential. Cambridge, MA: MIT Press. (2003).
6. Karp, R. et al: XOL: An XML-Based Ontology Exchange Language (version 0.4). (1999). Available: http://www.ai.sri.com/pkarp/xol/ (current February 2011).
7. Heflin, J. et al: SHOE: A Knowledge Representation Language for Internet Applications. Technical Report, CS-TR-4078 (UMIACS TR-99-71), Dept. of Computer Science, University of Maryland. (1999).
8. Kent, R. Ontology Markup LanguageVersion 0.3. (1999). Available: http://www.ontologos.org/OML/OML%200.3.htm (current February 2011).
9. Brickley, D., Guha, R.V., (Eds.) RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation. (2004). Available: http://www.w3.org/TR/rdf-schema/ (current February 2011).
10. McGuinness, D., Fikes, R., Handler, J., Stein, L.: DAML+OIL: An Ontology Language for the Semantic Web. *IEEE Intelligent Systems*, Vol. 17, No. 5, 72-80. (2002).
11. McGuinness, D., van Harmelen, F. (Eds.): OWL Web Ontology Language – Overview. W3C Recommendation. (2004). Available: http://www.w3.org/TR/owl-features/ (current February 2011).
12. Stanojević, M., Vraneš, S.: Knowledge representation with SOUL. *Expert Systems with Applications*, Vol. 3, No. 1, 122-134. (2007).
13. Stanojević, M., Vraneš, S.: NIMFA - Natural language Implicit Meaning Formalization and Abstraction, *Expert Systems With Applications*, Vol. 37, No. 12, 8172-8187. (2010).
14. O'Brien, G., Opie, J.: Radical connectionism: thinking with (not in) language. Language & Communication, No. 22, 313–329. (2002).
15. Hinton, G.E.: Mapping Part-Whole Hierarchies into Conntectionist Networks. *Artificial Intelligence*, Vol. 46, No. 1-2, 47-75. (1990).
16. Hawkins, J.: Learning Like A Human. IEEE Spectrum, 17-22. (April 2007).

17. Kharlamov, A., Raevsky. V.: Networks Constructed of Neuroid Elements Capable of Temporal Summation of Signals. In Rajapakse, J., Wang L. (Eds.): Neural Information Processing: Research and Development. Springer, 56-76. (2004).
18. Allen, J.: Natural Language Understanding, Second Edition. Redwood City, CA: The Benjamin/Cummings Publishing Company. (1994).
19. Man, W.C., Taboada M.: Intro to RST (Rhetorical Structure Theory). http://www.sfu.ca/rst/ (current February 2011).
20. Groza, T., Handschuh, S., Clark, T., Buckingham Shum, S., de Waard, A.: A short survey of discourse representation models. Proceedings 8th International Semantic Web Conference, Workshop on Semantic Web Applications in Scientific Discourse. Lecture Notes in Computer Science. Berlin: Springer Verlag. (2009).
21. Feldman, R., Aumann, Y., Finkelstein-Landau, M., Hurvitz, E., Regev, Y., Yaroshevich, A.: A Comparative Study of Information Extraction Strategies. *Lecture Notes in Computer Science*, Vol. 2276, 349-359. (2002).
22. Cunningham, H.: Information Extraction, Automatic. In K. Brown, (Ed.): Encyclopedia of Language and Linguistics, 2nd Edition. Elsevier. (2005).

Dr. Mladen Stanojević is a Senior Researcher at the Mihailo Pupin Institute. He received his MSc degree in the field of belief revision/truth maintenance and PhD in the area of natural language processing, both from the University of Belgrade. He has been working on applying techniques from artificial intelligence and natural language processing in real-world decision support systems, NLP based services and user interfaces, intelligent web applications, etc. His research interests include knowledge representation techniques, natural language processing techniques, semantic web, web services, service oriented architectures, etc. In these areas he published over 100 scientific papers. He is a member of IEEE.

Prof. Dr. SanjaVraneš is jointly appointed as a Scientific Director of the Mihajlo Pupin Institute and as a Professor of Computer Science at the University of Belgrade. From 1999 she has been engaged as a United Nations Expert for information technologies, and from 2005 as the expert evaluator and reviewer of EC Framework Programme Projects. Her research interests include semantic web, knowledge management, decision support systems, multicriteria analysis algorithms, complex event processing, etc. In these areas she published over 170 scientific papers. She serves as a reviewer of respectable international journals, like IEEE Transaction on Computer, IEEE Intelligent Systems magazine. She was a postdoctoral research fellow at the University of Bristol, England in 1993 and 1994. In the period 1999-2004 she was a Scientific Consultant at the ICS-UNIDO, International Center for Science and High Technology in Trieste, Italy. She has also served as a project leader and/or principal architect of more than 20 complex software projects. She is a member of IEEE, ACM and AAAI. She is also a member of Serbian Academy of Engineering Sciences and of National Scientific Council.

Received: September 1, 2010; Accepted: July 5, 2011.

DOI: 10.2298/CSIS100425067K

A Genetic Algorithm for the Routing and Carrier Selection Problem*

Jozef Kratica[1], Tijana Kostić[2], Dušan Tošić[3], Djordje Dugošija[3], and Vladimir Filipović[3]

[1] Mathematical Institute, Serbian Academy of Sciences and Arts,
Kneza Mihajla 36/III, 11 000 Belgrade, Serbia
jkratica@mi.sanu.ac.rs
[2] UCLA Department of Mathematics
Box 951555 Los Angeles, CA 90095-1555, USA
kostict@math.ucla.edu
[3] Faculty of Mathematics, University of Belgrade,
Studentski trg 16/IV, 11 000 Belgrade, Serbia
{dtosic | dugosija | vladaf}@matf.bg.ac.rs

Abstract. In this paper we present new evolutionary approach for solving the Routing and Carrier Selection Problem (RCSP). New encoding scheme is implemented with appropriate objective function. This approach in most cases keeps the feasibility of individuals by using specific representation and modified genetic operators. The numerical experiments were carried out on the standard data sets known from the literature and results were successful comparing to two other recent heuristic for solving RCSP.

Keywords: vehicle routing problems, genetic algorithm, evolutionary computation, combinatorial optimization.

1. Introduction

The delivery of goods from a warehouse to local customers is an important and practical problem of the logistics management. It has to decide which customers are to be served by the heterogeneous internal fleet and to route the vehicles of the internal fleet. A internal vehicle allows a company to consolidate several shipments, going to different destinations, in a single route. This problem with internal fleet and external carriers has also known as VRPPC (vehicle routing problem with private fleet and common carriers) in the literature ([4])

Demand of remaining customers must be served by external carriers. An external carrier usually assumes the responsibility for routing each shipment from origin to destination. The freight charged by a external carrier is usually much higher than the cost of a vehicle in internal fleet. Therefore, in some cases, it may be more economical to use external carriers instead of using an internal

* This research was partially supported by Serbian Ministry of Education and Science under the grant no. 174010.

vehicle to serve one or very few customers. Also, if total demand is greater than whole capacity of the heterogeneous internal fleet, logistics managers have to consider using an external carrier. Selecting the right mode to transport a shipment may yield significant cost savings to the company.

Routing and Carrier Selection Problem (RCSP) consists of the routing fixed number of vehicles with the limited capacity from the central warehouse to the customers with known demand. The objective is to minimize overall cost, which consists of the three parts: fixed cost of the using necessary vehicles in the internal fleet, variable transportation cost of routing every vehicle and a cost of freight for remaining customers (not served by internal fleet) charged by external carriers. RCSP is a NP-hard problem, as an generalization of the well known vehicle routing problem.

2. Related work

The routing problems are well-known combinatorial optimization problems and many approaches for solving these have been proposed. Good survey of new contributions can be found in [10] and for multi-objective case in [11].

However, only in several papers the problem with the vehicle routing when external carrier services are available, was considered. In [2] a fleet planning problem for long-haul deliveries with fixed delivery locations and an option to use an external carrier is considered. Static problem with a fixed fleet size and optional use of a outside carrier is considered in [1]. In [12] is described a methodology to address the fleet size planning and to route limited vehicles from a central warehouse to customers with random daily demands is developed. Paper [7] considered the problem where the company has only one vehicle.

In [5] selected customers were served by the external carriers, then used the modified version of heuristic proposed by [6] to construct the routes to serve the remaining customers and finally uses the local search (steepest descent heuristics) to improve the obtained solution.

The method proposed in [3], is named SRI (Selection, Routing and Improvement) heuristic, and it is composed of the following steps: (a) select customers to be served by the external carrier, (b) construct a first initial solution, (c) improve the obtained solution, (d) construct another initial solution, and (e) improve the second solution. Then, the best obtained solution is retained as a final solution. Use of two initial solutions increases chance to get good solutions within a very reasonable computation time.

Paper [4] describes metaheuirstic that uses perturbation procedure in the construction and improvement phases. It also performs exchanges between the sets of customers served by the private fleet and the common carrier. The obtained results clearly indicate that perturbation metaheuirstic is useful tool for solving RCSP.

A tabu search heuristic with a neighborhood structure based on ejection chains is described in [20]. In is empirically demonstrated that proposed algorithm slightly outperforms previous approaches reported in the literature.

3. Mathematical formulation

In our problem one central warehouse is considered. All vehicles start at the warehouse and return back to it. Also, the demands of all customers are known in advance and they cannot exceed the vehicle capacity. Each customer also has to be served by exactly one vehicle (either from internal fleet or by external carrier).

In the following an integer programming model given in [5] is presented:

$$\min \left(\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{m} c_{ijk}x_{ijk} + \sum_{i=1}^{n} e_i z_i\right) \tag{1}$$

$$\sum_{k=1}^{m} y_{1k} = m \tag{2}$$

$$z_i + \sum_{k=1}^{m} y_{ik} = 1 \qquad i = 2, n; \tag{3}$$

$$\sum_{i=1}^{n} q_i \cdot y_{ik} \leq Q_k \qquad k = 1, m; \tag{4}$$

$$\sum_{j=1}^{n} x_{ijk} = y_{ik} \qquad i = 1, n; \;\; k = 1, m; \tag{5}$$

$$\sum_{j=1}^{n} x_{jik} = y_{ik} \qquad i = 1, n; \;\; k = 1, m; \tag{6}$$

$$\sum_{i\in S}\sum_{j\in S} x_{ijk} \leq |S| - 1 \qquad for\ all\ subsets\ S \subseteq \{2, 3, ..., n\}, \;\; k = 1, n; \tag{7}$$

$$x_{ijk} \in \{0, 1\}; \;\; y_{ik} \in \{0, 1\}; z_i \in \{0, 1\}; \quad i = 1, n; \;\; j = 1, n; \;\; k = 1, m; \tag{8}$$

In relations (1)-(8) the number of customers is denoted by n and number of vehicles by m. The coefficients f_k and Q_k are fixed cost and capacity of vehicle k ($k = 1, .., m$). q_i and e_i represent demand and cost charged by external carrier (if it is used) of customer i ($i = 1, .., n$). Transportation cost of truck k traveling from customer i to customer j is represented by c_{ijk}. Variable $z_i = 1$ if customer i is served by external carrier, and 0 otherwise. Similarly, $y_{ik} = 1$ if customer i is

served by vehicle k and $x_{ijk} = 1$ if vehicle k travels from customer i to customer j.

Constraint (2) ensures that all vehicles have been assigned to customers, (3) ensures that every customer is served by internal fleet or external carriers, (4) denotes vehicle capacity constraint, while (5) and (6) ensure that a vehicle arrives to a customer and also leaves that location. Subtour breaking constraints are given in (7).

There is another ILP formulation (with polynomial number of constraints) for RCSP, proposed in [4]. Moreover, we use ILP formulation only to explain problem, but without applying it in our algorithm.

4. Proposed genetic algorithm

Genetic algorithms (GAs) are robust stochastic search techniques which imitate some spontaneous optimization processes in the natural selection and reproduction. At each iteration (generation) GA manipulates a set (population) of encoded solutions (individuals), starting from either randomly or heuristically generated one. Individuals from the current population are evaluated using a fitness function to determine their qualities. Good individuals are selected to produce the new ones (offspring), applying operators inspired from nature (crossover and mutation), and they replace some of the individuals from the current population. Genetic algorithms are easily hybridized with other methods (for example see [16, 18]).

A global description of used genetic algorithm is given by pseudo-code in Figure 1.

$Npop$ denotes the number of individuals in a population and $val(i)$ is objective value of i-th individual.

Detailed description of GAs is out of this paper's scope and it can be found in [19]. Extensive computational experience on various optimization problems shows that GA often produces high quality solutions in a reasonable time. Some of recent applications are [8, 14, 17].

In this section we shall describe a GA implementation for solving the Routing and Carrier Selection Problem - RCSP. Computational results summarized in Tables 1 and 3 are very encouraging and give further justification of GA robustness.

4.1. Representation and objective function

The representation of individuals in this GA implementation is completely different from the other GA approaches for vehicle routing problems. In other GA methods, representation of particular gene include customer number in current problem instance. Customer's number usually is not an important value in the particular problem instance. For example, if customers are permuted in problem instance, solution value will remain the same.

```
Input_Data();
Population_Init();
while not Finish() do
  for i:=1 to Npop do
       if(Exist_in_Cache(i))
          then
               val(i) := Get_Value_From_Cache();
          else
               val(i) := Objective_Function(i);
        endif
        Update_Cache_Memory();
  endfor
  Fitness_Function();
  Selection();
  Crossover();
  Mutation();
endwhile
Output_Data();
```

Fig. 1. The basic scheme of our GA implementation

In order to use problem instance's characteristics in better way, in our GA implementation each gene represents relative distance between current customer and other unserved customers. This idea emerges from fact that in the optimal route of every vehicle consecutive customers are often close to each other. This means, if we increase chances of choosing relatively close customers in route of current vehicle, that will lead genetic algorithm toward promising regions of search space.

In the initial part of the GA, for every customer, we arrange other customers in the non-decreasing order of their distances (see Example 1). Genetic code of each individual consists of the $n-2$ genes, since the customer 1 is warehouse and it has not demand. The last customer is determined by all previous. Procedure starts from the warehouse and the first vehicle. We take the gene that corresponds to current vehicle and current customer from the genetic code. If that gene has value r, we take the $r+1$th closest unserved customer (warehouse is also not counted) from the matrix described above. If that customer has demand that can be served by current vehicle, we add it to the route of the current vehicle. If its demand exceeds capacity of current vehicle, the vehicle returns to warehouse. In that case, next vehicle is used, current customer is moved to warehouse and previous procedure is repeated. Due to the fact that load of vehicles is determined in sequential way, subsequent vehicles are empty. When there are no more vehicles available, demands of remaining customers are served by external carriers.

Example 1. Instance $chu1$ from [5] has n=6 customers and m=2 vehicles:

customers:					vehicles:			Customers arranged
i	xcoord	ycoord	q	e	k	Q	f	by distance:
1	35	35	0	0	1	40	60	1: 2 3 6 4 5
2	41	49	10	90	2	30	50	2: 4 1 5 6 3
3	35	17	7	108				3: 1 5 6 2 4
4	55	45	13	132				4: 2 1 5 3 6
5	55	20	19	150				5: 3 4 1 2 6
6	15	30	26	120				6: 1 3 2 5 4

Vehicles of the internal fleet have transportation costs $1.5/per mile (number of miles is rounded to integer). Suppose that the genetic code is 3 1 0 1. The first gene 3 means that the 4th closest (unserved) customer to open warehouse is chosen (customer 4). The second gene 1 is applied to customer 4 and its 2nd closest unserved customer is chosen (customer 5, since open warehouse is not count into the account). The third gene 0 denotes that the closest customer is chosen (customer 3). Since the 2nd closest unserved customer is 2 (1 and 5 are served) and its demand exceeds the capacity of vehicle 1 ($13+19+7+10 = 49 > 40$), vehicle must go to the warehouse (route of the 1st vehicle is: 1 4 5 3 1). We continue with the next (second) vehicle. The 2nd closest unserved customer of the warehouse is 6 (2 is closest, 3 is previously served). Since the demand of last customer 2 exceeds the capacity of vehicle 2 ($26 + 10 > 30$) and there are no more vehicles, customer 2 is served by an external carrier. Then, the route of vehicle 2 is: 1 6 1. This solution is optimal (see [3] or [5]) and has total cost of 387.5 .

4.2. Genetic operators

Our GA implementation experimented with tournament and fine-grained tournament selection - FGTS (described in [9]). The FGTS depends on the real parameter F_{tour} - the desired average tournament size that takes real values. Actually, the average tournament size should be as close as possible to F_{tour}. It is implemented using two types of tournaments. During one generation, tournaments are held with different number of competitors. The first tournament type is held k_1 times and its size is $\lfloor F_{tour} \rfloor$. The second type is performed k_2 times with $\lceil F_{tour} \rceil$ individuals participating ($\lfloor x \rfloor = r$ and $\lceil x \rceil$ = s $\Longleftrightarrow r \leq x \leq s$ and $r, s \in Z,\ x \in R$) that implies $F_{tour} \approx \frac{k_1 \cdot \lfloor F_{tour} \rfloor + k_2 \cdot \lceil F_{tour} \rceil}{k_1 + k_2}$.

The crossover operator is applied on a selected pair of parents producing two offspring. The one-point crossover is applied by randomly choosing crossover points and simply exchanging the segments of the parents' genetic codes. Crossover points are chosen on gene borders to prevent disruption of good genes. For example, if every gene has length 4, only possible crossover points are 4, 8, 12, etc.

The standard simple mutation operator is performed by changing a randomly selected gene in the genetic code of the individual, with a certain mutation rate. Since the number of genes in this GA implementation is $n-2$, the mutation rate

is $\frac{0.4}{n-2}$. Extensive computational experience on various optimization problems shows that this rate is appropriately chosen.

4.3. Caching GA

The running time of the GA is improved by caching (see [13]). The evaluated objective functions are stored in the hash-queue data structure, with the corresponding genetic codes. When the same genetic code is obtained again during the GA, the objective value is taken from the hash-queue table, instead of computing the objective function. The Least Recently Used (LRU) strategy is applied for caching GA. The number of cached function values is limited to 5000 in this implementation.

4.4. Other GA aspects

The population numbers 150 individuals and in the initial population (the first generation) is randomly generated. This approach provides maximal diversity of the genetic material and better gradient of objective function. A steady-state generation replacement with elitist strategy is used. In this replacement scheme only N_{nonel} = 50 individuals are replaced in every generation, while the best N_{elite} = 100 individuals are directly passed in the next generation preserving highly fitted genes. The elite individuals do not need recalculation of objective value since each of them is evaluated in one of the previous generations.

Duplicated individuals are removed from each generation. Their fitness values are set to zero, so that selection operator prevents them from entering into the next generation. This is very effective method for saving the diversity of genetic material and keeping the algorithm away from premature convergence. The individuals with the same objective function but different genetic codes in some cases may dominate the population. If their codes are similar, the GA can lead to local optimum. For that reason, it is useful to limit their appearance to some constant. In this GA application this constant is set to 40.

5. Computational results

In this section the computational results of the GA method and comparisons with existing algorithms are presented. All tests were carried out on an Intel 1.4 GHz with 256 MB memory. The algorithms were coded in C programming language. We tested our GA method on all RCSP instances proposed in [5] and [3].

The finishing criterion of GA is the maximal number of generations N_{gen} = 5000. The algorithm also stops if the best individual or best objective value remains unchanged through N_{rep} = 2000 successive generations. Since the results of GA is nondeterministic, method was applied 20 times on each problem instance.

The Table 1 summarizes the GA result on instances described above and is organized as follows:

- the first three columns contain the test instance name, the number of customers and vehicles respectively;
- the fourth column contains the optimal solution, named Opt for all solutions that are proved to be optimal. For instances chu5 and new5 optimal solution is not known because CPLEX in [3] was stopped after 150 hours. For this reason best known solution of CPLEX for these instances is noted with $*$.;
- the best solution obtained by GA in 20 runs, named GA_{best}, is given in fifth column;
- the average running time (t) used to reach the final GA solution for the first time is given in the sixth column, while the seventh and the eighth column (t_{tot} and gen) show the average total running time and the average number of generations for finishing GA, respectively;
- in the last two columns $eval$ represents the average number of the objective function evaluations, while $cache$ displays savings (in percent) achieved by using the caching technique.

Routes of internal vehicles and which customers are served by external carriers are presented in Table 2. The first column display instance's name, while the next two columns, presents information about optimal and GA solutions, respectively.

Next, in Table 3, we compare results of the GA method with Chu heuristic from [5] and SRI from [3]. The first two columns in Table 3 display instance's name and optimal solution. Next three columns contain results of GA: best GA solution in 20 runs, average time t in seconds needed to detect the best GA solutions and t_{tot} represents the total time (in seconds) needed to reach finishing criterion. Next two columns taken from [5] represent best solution and running time of the his heuristic. Since running times of SRI heuristic is not reported in [3], last column represents only solution values obtained by SRI heuristic.

Table 1. GA results

$Inst$	n	m	Opt	GA_{best}	t (sec)	t_{tot} (sec)	gen	$eval$	$cache$ (%)
chu1	6	2	387.5	opt	0.002	0.910	2001	3025	97.0
chu2	11	2	586.0	opt	0.075	1.357	2107	71695	32.1
chu3	16	3	823.5	opt	0.199	1.667	2261	85229	24.7
chu4	23	2	1389.0	1407.0	0.784	2.923	2742	125115	8.9
chu5	30	3	1441.5^*	1461.0	2.222	4.780	3556	167732	5.7
new1	6	2	423.5	opt	0.002	0.903	2005	2947	97.1
new2	11	2	476.5	opt	0.022	1.316	2032	65566	35.6
new3	16	3	777.0	opt	0.647	2.113	2838	105460	25.8
new4	23	2	1521.0	1545.0	0.449	2.603	2422	111690	8.0
new5	30	3	1609.5^*	b.k.	1.652	4.372	3189	149655	6.3

Optimal solutions in Tables 1 and 3 are noted by opt, instances that are not tested by $n.t.$ and best known solutions by $b.k.$ Although, Table 3 does not

Table 2. Routes of vehicles

Inst	*Optimal solution*	*GA solution*
Chu1	Route 1:1-3-5-4-1 Route 2:1-6-1 External carrier:2	Route 1:1-4-5-3-1 Route 2:1-6-1 External carrier:2
Chu2	Route 1:1-4-3-10-11-5-1 Route 2:1-2-9-8-7-1 External carrier:6	Route 1:1-4-3-10-11-5-1 Route 2: 1-7-8-9-2-1 External carrier:6
Chu3	Route 1:1-14-16-6-3-2-7-1 Route 2:1-8-12-15-9-1 Route 3:1-4-10-11-13-1 External carrier:5	Route 1: 1-7-2-3-6-16-14-1 Route 2: 1-8-12-15-9-1 Route 3: 1-13-11-10-4-1 External carrier:5
Chu4	Route 1:1-8-22-5-6-9-10-14-12-13-1 Route 2:1-7-2-3-4-17-16-15-18-23-21-20-19-1 External carrier:11	Route1:1-19-20-23-21-18-15-16-17-4-3-2-7-14-12-13-1 Route 2: 1-22-5-6-9-10-8-1 External carrier:11
Chu5	Route 1:1-21-23-3-6-5-4-20-1 Route 2:1-16-17-14-8-18-10-15-9-13-12-11-24-19-1 Route 3:1-27-28-29-30-26-25-2-7-1 External carrier:22	Route 1: 1-20-21-4-5-6-2-7-25-26-30-28-29-27-1 Route 2: 1-19-24-15-9-18-10-8-14-17-16-13-12-11-1 Route 3: 1-13-11-10-4-1 External carrier:22
New1	Route 1:1-6-4-1 Route 2:1-2-5-1 External carrier:3	Route 1: 1-6-4-1 Route 2: 1-2-5-1 External carrier:3
New2	Route 1:1-2-5-8-11-6-10-1 Route 2:1-7-4-9-1 External carrier:3	Route 1: 1-8-2-5-11-6-10-1 Route 2: 1-7-4-9-1 External carrier:3
New3	Route 1: 1-9-11-14-3-5-1 Route 2: 1-13-8-16-12-6-1 Route 3: 1-2-10-4-15-1 External carrier:7	Route 1: 1-5-3-14-11-9-1 Route 2: 1-6-12-16-8-13-1 Route 3: 1-2-10-4-15-1 External carrier:7
New4	Route 1: 1-16-2-9-7-21-18-5-8-1 Route 2: 1-23-6-20-19-4-22-10-13-3-12-15-14-17-1 External carrier:11	Route 1: 1-17-14-15-12-3-13-10-4-22-19-20-6-5-23-1 Route 2: 1-8-18-21-7-9-2-16-1 External carrier:11
New5	Route 1:1-23-26-4-22-16-19-5-7-9-24-14-11-1 Route 2: 1-18-12-29-2-17-28-1 Route 3: 1-25-10-8-15-30-21-27-13-20-6-1 External carrier:3	Route 1:1-23-26-4-22-16-19-5-7-9-24-14-11-1 Route 2: 1-25-10-8-15-30-21-27-13-20-6-1 Route 3: 1-1-18-12-29-2-17-28-1 External carrier:3

Table 3. GA compared with Chu and SRI heuristic

$Inst$	Opt	GA			Chu		SRI
		Sol	$t[s]$	$t_{tot}[s]$	Sol	$t[s]$	Sol
chu1	387.5	opt	0.002	0.910	opt	3.14	opt
chu2	586.0	opt	0.075	1.357	631	4.58	opt
chu3	823.5	**opt**	0.199	1.667	900.0	5.88	826.5
chu4	1389.0	1407.0	0.784	2.923	1681.5	8.42	**opt**
chu5	1441.5^*	1461.0	2.222	4.780	1917	11.06	**1444.5**
new1	423.5	opt	0.002	0.903	n.t.	n.t.	opt
new2	476.5	opt	0.022	1.316	n.t.	n.t.	opt
new3	777.0	**opt**	0.647	2.113	n.t.	n.t.	804.0
new4	1521.0	**1545.0**	0.449	2.603	n.t.	n.t.	1564.5
new5	1609.5^*	b.k.	1.652	4.372	n.t.	n.t.	b.k.

contain complete comparisons on all instances, because instances $new1$-$new5$ are proposed in [3], after publication of the paper [5]).

The data from Table 3 show that the GA method reached optimal solution (or best known solution for $chu5$ and $new5$ instances) in 7/10 cases, SRI in 6/10 cases and Chu heuristic only in 1/5 cases. For more clear comparison, solutions in Table 3, that are strictly better than solutions of other methods are bolded and underlined. We can see that GA is strictly better than other methods in 3 cases ($chu3$, $new3$, $new4$), SRI is strictly better than other methods in 2 cases ($chu4$, $chu5$), while Chu heuristic is never strictly better than other methods. From these results, it is quite obvious that GA and SRI outperform Chu heuristic, GA also obtain better solutions than SRI. It is obvious that GA produces high quality solutions in the reasonable time.

Our GA approach is also tested on large-scale instances from [4], both homogeneous and heterogeneous, and results are presented in Table 4 and Table 5. In that case our GA algorithm is hybridized with local search used in [20]. Local search is not applied on each individual since it is very time consuming. The strategy of applying local search from [15] is reapplied for this problem.

For all large-scale instances the optimal solution is not known, so the best solutions reported in the literature are used instead of optimal ones.

6. Conclusions

We present new heuristic, based on a genetic search framework, for solving the Routing and Carrier Selection Problem. Arranging unserved customers in non-decreasing order by their distances from current customer directs GA to promising search regions. Computational experiments on existing RCSP instances demonstrate the robustness of the proposed algorithms with the respect to the solution quality and running times. Comparisons with results from the literature show the appropriateness of proposed algorithm.

Table 4. GA results on large homogeneous RCSP instances

$Inst$	$Best$	GA		
		Sol	$t[s]$	$t_{tot}[s]$
CE-01	1119.47	1158.98	48.778	49.666
CE-02	1814.52	1893.66	90.213	91.234
CE-03	1921.1	1987.75	104.078	111.434
CE-04	2515.5	2668.87	191.656	204.088
CE-05	3097.99	3279.64	308.912	342.056
CE-06	1207.47	1233.20	46.605	47.316
CE-07	2004.53	2086.17	91.102	91.969
CE-08	2052.05	2130.82	107.776	113.816
CE-09	2429.19	2558.70	204.949	224.706
CE-10	3393.41	3598.36	314.969	358.665
CE-11	2330.94	2383.34	125.611	137.418
CE-12	1952.86	2042.84	105.744	111.045
CE-13	2858.94	2929.02	125.086	163.665
CE-14	2214.14	2338.22	101.846	103.830
G-01	14160.77	14910.52	479.625	629.328
G-02	19208.52	20258.91	1699.072	4568.416
G-03	24592.18	25941.17	7691.849	13529.265
G-04	34607.12	36083.77	9697.439	22360.709
G-05	14249.82	14875.44	1002.598	1803.100
G-06	21498.03	22440.03	3231.161	4826.779
G-07	23513.06	24621.42	6638.286	11098.164
G-08	30073.56	31326.38	7311.450	12532.019
G-09	1323.57	1368.47	2032.363	3236.873
G-10	1590.82	1646.20	5633.333	7682.187
G-11	2166.66	2235.24	9246.969	17381.100
G-12	2490.01	2578.12	18287.615	32100.689
G-13	2268.32	2347.49	772.484	1113.649
G-14	2693.35	2796.74	1487.419	2454.822
G-15	3157.31	3283.07	3264.514	5083.658
G-16	3637.52	3804.04	6351.355	11131.333
G-17	1631.49	1898.36	329.982	372.932
G-18	2691.61	3079.03	613.516	851.768
G-19	3452	3940.71	878.972	1110.866
G-20	4272.98	4823.76	1085.466	1606.512

Table 5. GA results on large heterogeneous RCSP instances

$Inst$	$Best$	GA		
		Sol	$t[s]$	$t_{tot}[s]$
CE-H-01	1191.7	1203.27	45.697	49.168
CE-H-02	1790.67	1860.84	92.357	94.371
CE-H-03	1917.96	1988.73	101.455	109.335
CE-H-04	2475.16	2622.24	191.363	206.061
CE-H-05	3143.01	3314.16	308.520	340.371
CE-H-06	1204.48	1210.75	50.987	51.531
CE-H-07	2025.98	2108.23	89.202	90.408
CE-H-08	1984.36	2057.75	109.998	119.157
CE-H-09	2438.73	2601.96	212.156	233.717
CE-H-10	3267.85	3415.40	326.473	382.491
CE-H-11	2303.13	2381.52	121.292	139.536
CE-H-12	1908.74	1954.80	104.721	109.677
CE-H-13	2842.18	2883.67	124.298	143.939
CE-H-14	1907.74	1988.79	104.464	111.559
G-H-01	14174.27	14812.40	460.866	661.852
G-H-02	18537.7	19395.20	2216.556	4757.013
G-H-03	25177.92	26523.43	5994.606	14040.338
G-H-04	34589.11	36261.53	12965.626	23744.678
G-H-05	15411.82	16254.20	572.518	889.646
G-H-06	19859.3	20717.86	2509.686	5350.663
G-H-07	23481.28	24727.21	5475.465	10955.261
G-H-08	27334.84	28605.47	13105.465	23568.062
G-H-09	1329.27	1386.03	1904.543	3259.721
G-H-10	1554.96	1622.14	4642.931	8750.368
G-H-11	2191.23	2266.04	6885.852	14759.343
G-H-12	2482.92	2580.32	14375.062	32527.158
G-H-13	2231.88	2330.81	839.558	1256.318
G-H-14	2682.85	2809.86	1043.280	2687.650
G-H-15	3123.6	3285.70	2379.560	5963.082
G-H-16	3621.85	3780.43	5750.238	10786.368
G-H-17	1664.08	1932.18	344.216	379.060
G-H-18	2708.73	3062.08	571.672	729.860
G-H-19	3443.59	3892.96	717.370	1004.026
G-H-20	4306.53	4865.32	930.398	1450.434

Our future research will be directed to parallelization of the presented GA, incorporation it in exact methods and its applying in solving similar routing problems.

References

1. Agarwal, Y.K.: Vehicle routing with limited fleet and common carrier option. TIMS/ORSA Joint National Meeting, Boston (1985)
2. Ball, M.O., Golden, A., Assad, A., Bodin, L.D.: Planning for truck fleet size in the presence of a common-carrier option. Decision Sciences 14, 103–120 (1983)
3. Bolduc, M.C., Renaud, J., Boctor, F.: A heuristic for the routing and carrier selection problem. European Journal of Operational Research 183, 926–932 (2007)
4. Bolduc, M., Renaud, J., Boctor, F., Laporte, G.: A perturbation metaheuristic for the vehicle routing problem with private fleet and common carriers. Journal of the Operations Research Society 59, 776–787 (2008)
5. Chu, C.: A heuristic algorithm for the truckload and less-than-truckload problem. European Journal of Operational Research 165, 657–667 (2005)
6. Clarke, G., Wright, J.W.: Scheduling of vehicles from a central depot to a number of delivery points. Operations Research 12, 568–581 (1964)
7. Diaby, M., Ramesh, R.: The distribution problem with carrier service: a dual based penalty approach. ORSA Journal on Computing 7, 24–35 (1995)
8. Djurić, B., Kratica, J., Tošić, D., V., F.: Solving the maximally balanced connected partition problem in graphs by using genetic algorithm. Computing and Informatics 27(3), 341–354 (2008)
9. Filipović, V.: Fine-grained tournament selection operator in genetic algorithms. Computing and Informatics 22, 143–161 (2003)
10. Goel, A., Gruhn, V.: A general vehicle routing problem. European Journal of Operational Research 191(3), 650–660 (2008)
11. Jozefowiez, N., Semet, F., Talbi, E.G.: Multi-objective vehicle routing problems. European Journal of Operational Research 189(2), 293–309 (2008)
12. Klincewicz, J., Luss, H., M.G., P.: Fleet size planning when outside carrier service are available. Transportation Science 24, 169–182 (1990)
13. Kratica, J.: Improving performances of the genetic algorithm by caching. Computers and Artificial Intelligence 18, 271–283 (1999)
14. Kratica, J., Kovačević-Vujčić, V., Čangalović, M.: Computing strong metric dimension of some special classes of graphs by genetic algorithms. Yugoslav Journal of Operations Research 18(2), 143–151 (2008)
15. Kratica, J., Milanović, M., Stanimirović, Z., Tošić, D.: An evolutionary based approach for solving a capacitated hub location problem. Applied Soft Computing 11(1), 1858–1866 (2011)
16. Li, J., Pan, Q., Xie, S.: A hybrid variable neighborhood search algorithm for solving multi-objective flexible job shop problems. ComSIS 7(4), 907–930 (2010)
17. Liu, H.: Generative 3d images in a visual evolutionary computing system. ComSIS 7(1), 111–125 (2010)
18. Mao, Q., Zhan, Y.: A novel hierarchical speech emotion recognition method based on improved ddagsvm. ComSIS 7(1), 211–221 (2010)
19. Mitchell, M.: An introduction to genetic algorithms. MIT Press, Cambridge, Massachusetts (1999)

20. Potvin, J., Naud, M.: Tabu search with ejection chains for the vehicle routing problem with private fleet and common carrier. Journal of the Operations Research Society 62, 326–336 (2011)

Jozef Kratica was born in 1966 in Belgrade, Serbia. He received his B.Sc. degrees in mathematics (1988) and computer science (1988), M.Sc. in mathematics (1994) and Ph.D. in computer science (2000) from University of Belgrade, Faculty of Mathematics. In 2002 he joined Mathematical Institute as a researcher. As a delegation leader participated on the International Olympiads in Informatics (IOI'90 Minsk - Belarus, IOI'93 Mendoza - Argentina). His research interests include genetic algorithms (evolutionary computation), parallel and distributed computing and location problems.

Tijana Kostić was born in 1984 in Belgrade. She received bronze medal on the 41st International Mathematical Olympiad in Seoul 2000. She received her B.Sc. (2006) and M.Sc. (2007) degree in numerical mathematics from University of Belgrade, Faculty of Mathematics. Now, she is a Ph.D. student and Teaching/Research Assistant at UCLA, Department of Mathematics, USA. Her research interests include genetic algorithms and combinatorial optimization.

Dušan Tošić was born in 1949 in Knjaževac, Serbia. He received his B.Sc. degree in mathematics (1972), M.Sc. in mathematics (1977) and Ph.D. in mathematics (1984) from the University of Belgrade, Faculty of Mathematics. Since 1985 he has been professor of computer science at the Faculty of Mathematics. His research interests include parallel algorithms, optimization and evolutionary computation, numerical solving of the differential equations and teaching computer science.

Djordje Dugošija was born in 1947 in Sremska Mitrovica, Serbia. He received his B.Sc. degree in mathematics (1970), M.Sc. in mathematics (1974) and Ph.D. in mathematics (1986) from the University of Belgrade, Faculty of Mathematics. Since 1986 he has been professor of mathematical programming and discrete mathematics at the Faculty of Mathematics. He was the leader of the national team on many International Mathematical Olympiads. His research interests include operations research, combinatorial optimization and evolutionary computation.

Vladimir Filipović was born in 1968 in Podgorica, Montenegro. He received his B.Sc. degree (1993), M.Sc. (1998) and Ph.D. (2006) in computer science from University of Belgrade, Faculty of Mathematics. In 2006, he becomes assistant professor of computer science at the Faculty of Mathematics. His research interests include genetic algorithms, parallel algorithms and operational research.

Received: April 25, 2010; Accepted: October 21, 2011.

DOI: 10.2298/CSIS100804037L

An XML-algebra for Efficient Set-at-a-time Execution

Maxim Lukichev[1], Boris Novikov[1], and Pankaj Mehra[2]

[1] Saint-Petersburg University
Saint-Petersburg, Russia
maxim.lukichev@gmail.com, borisnov@acm.org
[2] Whodini,Inc.
CA, USA
pankaj@whodini.com

Abstract. The importance of XML query optimization is growing due to the rising number of XML-intensive data mining tasks. Earlier work on algebras for XML query focused mostly on rule-based optimization and used node-at-a-time execution model. Heavy query workloads in modern applications require cost-based optimization which is naturally supported by the set-at-a-time execution model. This paper introduces an algebra with only set-at-a-time operations, and discusses expression reduction methods and lazy evaluation techniques based on the algebra. Our experiments demonstrate that, for queries with complex conditional and quantified expressions, the proposed algebra results in plans with much better performance than those produced by the state-of-the-art algebras. For relatively simple queries, the proposed methods are expected to yield plans with comparable performance.

Keywords: native XML databases, XML query optimization, query algebras.

1. Introduction

High-level declarative query languages are one of the most important tools offered by database management systems. These languages have great expressive power and are easier to use than conventional programming languages. Modern query execution engines contain sophisticated query optimizers that transform a declarative query into an efficient sequence of low-level operations. The overall optimization process can be presented as a two-phase process, where the first phase is *logical optimization* and the second one, *cost-based optimization* [7]. Logical optimization focuses on rewriting the query into logically equivalent forms that are hopefully more efficient. Cost-based optimizations first creates detailed plans for each rewritten query and then selects the best one based on available statistics of the data.

An *algebra* of operations specifies equivalent operator reorderings. Relational database management systems use common relational algebra, however there is no widely accepted algebra for XML databases. The reasons for this are

partly historical, and partly to do with the complexity of the XQuery language. Known XQuery algebras can be classified according to the optimization phase that they target, as either *rule-based* (RB) or *cost-based* (CB).

RB algebras target the logical optimization phase. The operations of such algebras support both node-at-a-time (NT) and set-at-a-time (ST) execution models [10]. The former type sequentially pass execution for each input item to subsequent operations. Operations of the latter type take as arguments sets of tuples, just as relational operators do.

CB algebras require their operations to be cost-estimation friendly. Traditional cost estimation techniques [8] work only for ST operations for which the cardinality of the result can be estimated from the cardinalities of the arguments. For these techniques to apply, conditional predicates (if any) in the ST operations should test for simple value-based conditions, such as value equality or a structural relationship in case of XML (*e.g.* parent-child).

The XQuery language includes quantified and conditional expressions and *where*-clauses, each of which could involve potentially expensive predicates. Lazy evaluation of the query, *i.e.* performing only those operation evaluations that are necessary for producing the result, hence turns out to be an important technique. While NT-strategy naturally supports lazy evaluation, ST requires entire sets of input to be explicitly pre-evaluated, that in turn can lead to unnecessary computations.

In this paper, we propose algebraic transformations that generate more efficient ST plans by pushing selective operations closer to the leafs of the query plan. Unlike in the relational algebra, these algebraic transformations for XQuery are non-trivial, as they require additional logical-plan transformations in order to preserve correctness. An algebra that facilitates such transformations is the focus of this paper. Our algebra, XAnswer, covers all the XQuery constructs except recursive functions.

We do not discuss cost estimation techniques in detail, instead focusing on logical transformations of the inherently more efficient ST plans. We experimentally demonstrate performance gains for queries with complex conditional and quantified expressions by executing ST-operations in the order discovered using the proposed transformations.

The remainder of the paper has following structure. In the section 3 we define the XAnswer algebra and its main operations. In the section 4 we describe our rules for expression construction and their normalization. The expression construction rules map XQuery to XAnswer. Since the proposed rules produce redundant operations, in the section 5 we discuss transformations that reduce redundancy and enable lazy evaluation. The section 6 presents our experimental validation.

2. Related work

XML-query optimization has been the subject of recent intensive research. Early approaches [1, 2] are based on the algebra for XML Query introduced

in [18] and utilize rewriting rules over expressions of XQuery Core [6]. Such approaches, while are feasible for relatively simple queries, yield poor performance on queries with nested expressions because of use of the use of NT-style operations.

Later approaches fall into two groups: *tree-based* and *tuple-based*. These approaches differ in the kind of elements their operations are defined over. The former use tree-based algebras [4, 9] which operate on ordered sets of trees. The latter use relational-like structures, sets of tuples.

The operations used in tree-based approach generate trees for intermediate results and use pattern matching over those trees.

While most of the tuple-based approaches require the sets of tuples to be ordered, others such as the query algebra [14] avoid this limitation. While some algebras for tuple-based approach assume that tuples contain only atomic values [14], others permit values to be sequences [13, 15, 19], and some others [11, 16] allow values to be sets of tuples.

Tuple-based algebras enable traditional optimization techniques known from relational databases. Significant performance gains can be obtained by using *query unnesting* [11, 13], which is special transformation to substitute NT-operations with combinations of ST-operations. This gain is due to the ability to exploit hash join for implementing ST-operations, whereas for NT-operations a nested-loop join is likely the only possible implementation. One of the most aggressive unnesting schemes is used in Galax [13]. For query translation it uses special NT-operations *MapConcat*, *MapFromItem* and *Cond* for conditional expressions (*if-then-else*). It also defines an ST-operation *GroupBy*. Query unnesting is based on the idea of introducing the *GroupBy* operation where possible (actually, inside operations corresponding to *for*-clauses). Then, using other rules, *GroupBy* and *MapConcat* are swapped to replace *MapConcat* with cartesian product (or join) when possible.

Recent algebras [13, 15] are unable to unnest certain classes of nested expressions, instead being forced to utilize NT-operations. These operations appear in selections with complex predicates, as well as in quantified (*some* or *every*) or conditional expressions.

Let's consider a FLWOR with *if-then-else* clause:

```
for \$i in A/B
let \$b := \$i/D
return if \$i/c then \$b/T else \i/F
```

Translation of this query with NT-style condition, like in earlier work, will lead to the use of inefficient nested loop join and the lack of support for lazy evaluation will cause redundant execution of $\$i/D$ even if $\$i/c$ is *false*.

3. Algebra overview

In this section we introduce *XAnswer*, our tuple-based algebra, in which all operations can be *set-at-a-time (ST)*. Our algebra, has some similar components to XAT [19] and Galax. Just like XAT and Galax, its operations are defined over ordered sets of tuples. Each tuple is a set of items that can be either a single value (XML atomic value, XML-node [17]) or a sequence of single values. We call this structure an $envelope$ and define it formally:

Definition 1 *Envelope ($< he|be|re >$) is a triple of header* he*, body* be *and result attribute* re*. Header* he *is an unordered set of unique-whitin-this-header names called attributes (A). Result attribute* re *is an attribute from* he*. Body* be *is an ordered set of tuples (τ), where each tuple is a set of pairs (A, v) for each attribute* A *of the header. v is either a single value or a sequence of single values.*

In what follows, we will denote tuples as $\tau(e)$, where e is a sequence of values of corresponding tuple pairs. Envelopes are denoted as $< h|\tau(e_1)\ldots|r >$, where h is the header, r is the result attribute, $\tau(e_i)$ forms envelope body, and e_i denotes the sequence of values corresponding to the $i-th$ attribute within h. The signature $< || >$ is used for the empty envelope.

Envelopes are *similar* ($\simeq$) if they have identical headers and their bodies differ only in order of tuples. An envelope *includes* ($\succeq$) another envelope if they have identical headers and if the the body of the first contains all the tuples of the latter.

We continue with the description of the operations that form an algebraic basis. Next, we briefly introduce additional operations that simplify notations or support more efficient implementation.

3.1. Basic operations

Basic operations are presented in the Table 1. To uniquely identify new attributes, we use the function $nextId()$ to generate the attribute's names. An operation for attribute renaming is trivial, and we do not list it among the operations in the table.

Along with relational-like operations that appear in other tuple-based algebras [13, 14, 19], XAnswer adds the *union* operation. Unlike its relational counterpart, XAnswer's *union* does not remove duplicates.

In XAnswer, we introduced a left-outer-join operation instead of expressing it using selection, cross product, and union operators. This is so because envelopes' bodies are ordered and may contain duplicates. Using the left-outer-join along with selection and union, it is possible to express other set operations, namely intersection and subtraction.

To enable query unnesting, previous tuple-based algebras [13, 19] use tuples grouping. They evaluate a nested query in ST-fashion instead of NT, group

Table 1. Main algebraic operations

Operation name	Input	Output
Unary operations		
Function execution (f_f)	$< h \mid \tau(e_1) \dots \tau(e_n) \mid r >$	$< h, i = nextId() \mid \tau(e_1, f(e_1)) \dots \tau(e_n, f(e_n)) \mid i >$
Selection (σ_{pr})		$< h \mid \tau(e'_1) \dots \mid r >$, where $(e')_i \subseteq (e)_i$ and $pr(e_i) = true$
Projection ($\pi_{h'}$)		$< h' \mid \tau(e'_1 \mid_{h'}) \dots \mid r \mid_{h'} >$, where $h' \subseteq h$
Sort ($sort_{h'}$)		$< h' \mid \tau(e'_1) \dots \mid r >$, where $(e')_i = Sort_{h'}(e)_i$
Index ($index_i$)		$< h, i \mid \tau(e_1, 1) \dots \tau(e_n, n) \mid r >$
Nest ($nest_{h'}$)		See example table 2 and comments in section 3.1
Unnest ($unnest_{h'}$)		See example table 2 and comments in section 3.1
Duplicate (dup_{h_i})		$< h, nextId() \mid \tau(e_1, e_1 \mid_{h_i}) \dots \mid r >$, where $h_i \in h$
Binary operations		
Union ($\bigcup$)	$< h \mid \tau(e_1) \dots \mid r >$, $< h \mid \tau(e'_1) \dots \mid r >$	$< h \mid \tau(e_1) \dots \tau(e_n), \tau(e'_1) \dots \tau(e'_m) \mid r >$
Cross product ($\times$)	$< h \mid \tau(e_1) \dots \mid r >$, $< h' \mid \tau(e'_1) \dots \mid r' >$	$< h, h' \mid \tau(e_1, e'_1) \dots \tau(e_1, e'_n), \tau(e_2, e'_1) \dots \mid r' >$
Left outer join ($\bowtie^l_{pr}$)		$< h, h' \mid \tau(e_1, e''_1) \dots \tau(e_1, e''_n), \tau(e_2, e''_1) \dots \mid r' >$, where if $(pr(e_i, e'_i) = true)$ then $e''_i = e'_i$ else $e''_i = ()$

the results of the query into sequences, and append the sequences to corresponding tuples. In [13], this is done using a complex operation called *group by*. We define a lower-level operation $nest$ along with an $unnest$ operation that ungroups grouped tuples. The semantics of these operations is shown in the Table 2. While $nest$ and $unnest$ appear in [19], previously proposed tuple groupings are different from ours. Our $nest$ and $unnest$ operations are not complementary (see Table 2). Note that if h is empty in $nest_h$, all tuples fall into one group, and the resulting envelope has a single tuple with sequences of values for each attribute.

Table 2. Input and output of algebraic operations

(a) $nest_{A,B}$

Input			Output		
A	B	C	A	B	C
a_1	b_1	c_1	a_1	b_1	c_1
		c_2			c_2
a_1	b_2	c_3			
a_1	b_1	c_2	a_1	b_2	c_3

(b) $unnest_{A,B}$

Input			Output		
A	B	C	A	B	C
a_1	b_1	c_1	a_1	b_1	c_1
		c_2	a_1	b_1	c_2
		c_2	a_1	b_1	c_2
a_1	b_2	c_3	a_1	b_2	c_3

(c) dup_B

Input		Output		
A	B	A	B	B'
a_1	b_1	a_1	b_1	b_1
a_2	b_2	a_2	b_2	b_2

(d) $quant^{every}_{true,B}$

Input		Output	
A	B	A	B
a_1	$true$	a_2	true
	$false$		
	$true$		
a_2	$true$		

For duplication of attribute values we use *duplicate* operation (dup_a). The input and output are presented in the Table 2. The operation is used in plan reduction described in the section 5.2 to replace relatively expensive joins.

3.2. Additional operations

In order to simplify notation, we introduce specific leaf algebraic operations, *leaf path step* ($LPS_p := unnest_{re(f)}(f_p(< || >))$) and *leaf constructor* ($LC_c := unnest_{re(f)}(f_c(< || >))$). LPS is used to extract values stored in a document. LC is used to extract XML-values (constants) that are not present in a document (e.g. they occur in the text of a query). In LC_c operation, c is a sequence constructor (e.g. `(1,2,3)`) or an element constructor (e.g. `<a>text</a>`). In LPS_p operation, p is an XPath expression (e.g. `/a/b//c`).

Structural join is the traditional join operation with a structural predicate that specifies the relation between two nodes (e.g. parent-child). If structural predicate is applied to sequences of values, it evaluates to true if there is at least one value satisfying the predicate in each sequence. Unary path step and structural join are used in path expressions mapping.

XAnswer has a basic function operation (f_f). The pattern f specifies the overall structure of the element and places where corresponding tuple values should be substituted. In our notation, we add a special operation called *element constructor*(θ_p), where p is an element pattern which describes the schema of an element.

The purpose of *quantify* ($quant^q_{c,h_i}$) operation is to efficiently implement lazy evaluation (section 5.3). The operation is a specific kind of selection used for controlling the pipeline. A sample input and output of the quantify operation are presented in Table 2.

Addition operation is a special complex operation included in the algebra to avoid unnecessary steps in physical plan. The operation appears during the plan reduction described in the section 5.2. *Addition* is a combination of projection, nest and join: $A \bigoplus_p^v B = nest_{he(A)}(\pi_{he(A) \bigcup v}(A \bowtie_p^l B))$, where p is a predicate and $v \subseteq he(B)$.

4. Plan construction

The plan construction process consists of two steps, normalization and translation. During the process FLWOR expressions are broken into single for- and let-clauses [6], predicates are moved from xpath to where-clauses [5, 11, 13] and complex expressions are removed by introducing new variables [11].

4.1. Normalization

The proposed normalization differs from previous work mostly in the normalization of nested, quantified and conditional expressions.

Below we outline some of our logical transformations. We break up complex expressions via introduction of new variables in such a way that only let-clauses contain them. Any nested FLWOR expression starts with a for-clause. Positional predicates are

```
let $k := for $i in expr
                (for $j in ...)
          return cond
(: if "some" :)
let $r := $k = true
(: if "every" :)
let $r := not($k = false)
return $r
```

Fig. 1. Quantifiers normalization

moved to where-clauses. Quantified expressions are replaced with FLWORs (see Figure 1).

4.2. Translation

An algebraic expression is constructed in a top-down fashion. The Table 3 illustrates some of the expression construction rules. We assume that P is a subexpression that was constructed on a previous step and E is an expression corresponding to the bound expression ($expr$) of a let- or a for-clause, $he(E)$ or $re(E)$ means header or return attribute extraction from the resulting envelope of the expression E.

Table 3. Construction rules

	XQuery	XAnswer
1	`for $i in` $expr$	$P \times \pi_{re(E)}(E)$.
2	`let $i :=` $expr$	$(P \times nest_{()}(\pi_{re(E)}(E))$.
3	`for $i in` $expr(v)$	$\pi_{he(P)\cup re(E)}(index_i(P) \bowtie_{i=j} E(unnest_v(index_j(P))))$.
4	`let $i :=` $expr(v)$. $Expr$ is xpath expression that has a reference to previously defined variable	$\pi_{he(P)\cup re(E)}(nest_{he(P)\cup i}(index_i(P) \bowtie^l_{i=j} \pi_{re(E)\cup j}(E(unnest_v(index_j(P))))))$.
5	`let $i :=` $expr(v_1 \dots v_n)$. $Expr$ is not xpath and has references to previously defined variables	$\pi_{he(P)\cup re(E)}(nest_{he(P)\cup i}(index_i(P) \bowtie^l_{i=j} E(index_j(P))))$.
6	$expr(v_1 \dots v_n)$. $Expr$ is logical or algebraic expression operating with variables $v_1 \dots v_n$.	$f_{ep,v_1 \dots v_n}(E)$.
7	`where` $expr$.	$\sigma_{re(E)=true}(E(P))$.
8	`order by` $v_1 \dots v_n$.	$sort_{v_1 \dots v_n}(P)$.
9	`/axis::node-test`.	$P \bowtie_{::axis} LPS_{node-test}$.
10	$E_1 op E_2$. Op is one of *intersect, union, except*.	$P_1 op P_2$, $op \in (\cup, \cap, \setminus)$.
11	`if` $expr_c$ `then` $expr_t$ `else` $expr_f$.	See section 5.3.

Note that, if $P =< \,||\, >$, in the rules *1,2,9* only right operand of $\bowtie$ and $\times$ should be considered as a resulting subplan.

Example 1

Below we demonstrate sample plan construction process. Let's denote $h_A = re(LPS_A)$ and $h_B = re(LPS_B)$.

Step	Rule	Input	Output
0		for $v in A let $m := $v/B	$P =< \| \| >$
1	1	`for $v in A` ; $P =< \| \| >$	$P = \pi_{re(E)}(E)$
2	9	`A` ; $P =< \| \| >$	$E = LPS_A$
3	1	`for $v in A` ; $P =< \| \| >$; $E = LPS_A$	$P = \pi_{re(LPS_A)}(LPS_A)$
4	5	`let $m := $v/B` ; $P = \pi_{re(LPS_A)}(LPS_A)$	$\pi_{he(P)\cup re(E\|_{P'})}(nest_{he(P)\cup i}(index_i(P) \bowtie^l_{i=j} \pi_{re(E\|_{P'})\cup j}(E\|_{P'}))), P' = unnest_v(index_j(P))$
5	9	`$v/B`; $P' = unnest_v(index_j(P))$	$E\|_{P'} = P' \bowtie_{::child} LPS_B$
6	5	`let $m := $v/B`; $P = \pi_{re(LPS_A)}(LPS_A)$; $E\|_{P'} = P' \bowtie_{::child} LPS_B$	$P = \pi_{h_A\cup h_B}(nest_{h_A\cup i}(index_i(T) \bowtie^l_{i=j} \pi_{h_B\cup j}(unnest_v(index_j(T)) \bowtie_{::child} LPS_B))), T = \pi_{h_A}(LPS_A)$
Optimized: $\pi_{h_A \bigcup h_B}(nest_{h_A\cup i}((index_i(LPS_A)) \bowtie^l_{child} LPS_B))$			

5. Optimization

Efficient physical plans are derived by applying optimizing transformations to the constructed algebraic plans. The logical transformations of algebraic plans also enables further physical plan optimization.

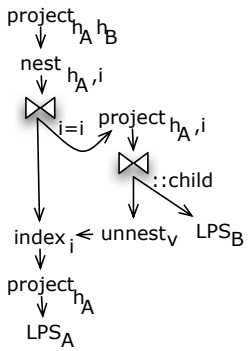


Fig. 2. The plan from Example 1

We explain all our transformations with the help of a directed graph in which vertices (nodes) represent algebraic operations and edges connect those operations to their operands (the Figure 2). In Example 1 above, $index_i(\pi_{h_A}(LPS_A))$ and $index_j(\pi_{h_A}(LPS_A))$ produce envelopes which differ only in headers. Such envelopes can be converted to each other with the *HR* operation. In such cases we consider the two subplans identical and leave only one in the graph (see Figure 2).

Definition 2 *An* operation block *is a subgraph that has no more than one node (*output node*) with incoming external edges, and has all outgoing external edges (if any) leading to only one external node. The targets of the outgoing edges are called* input nodes*.*

Below we consider an expression $B_1(B_2)$, where B_1 and B_2 are operation blocks, and all *input nodes* of B_1 have outgoing edges lead to the *output node* of B_2.

Definition 3 *An operation block B is* non-reducing *if an envelope En passed to the block is not empty and $\pi_{he(En)}(B(En)) \succeq En$. An operation block B is* non-modifying *if $\pi_{he(En)}(B(En)) \simeq En$.*

Non-reducing operation blocks keep all tuples of the input *envelope*. *Non-modifying* operation blocks are non-reducing blocks that do not add new tuples to the input *envelope*. Non-reducing and non-modifying properties can be naturally generalized from operation blocks to unary operations, join and left outer join (for joins, the left operand is considered as the input in the sense of Definition 3). Non-modifying operations include *duplication*, *function*, *sort*, etc. Non-reducing operations additionally include *left outer join*. *Join* and *selection* are reducing operations because tuples of the (left) operand may be eliminated.

Cycles appear in the graph as a result of constructing parts of the plan corresponding to for- (*for-cycles*) and let-clauses (*let-cycles*) that refer to previously defined variables. *Let-cycle* are non-modifying due to use of *nest* and *left outer join* (see rule 5 of Table 3). *For-cycles* are reducing operation blocks.

Definition 4 *Two* operation blocks B_1 *and* B_2 *are* independent *if $B_2(B_1(P)) \simeq B_1(B_2(P))$.*

If a block operates with attributes that were added to the header of the resulting envelope by another block, it *depends* on the other block. Independent blocks never depend on each other.

Optimizing transformations listed in the following sections preserve the value of $\pi_{re(P)}(P)$. This is achieved by using an *HR* operation to change the *return attribute* if needed. For compactness we omit usages of an *HR* operation.

5.1. Blocks pushdown

Optimizing transformations include reordering, removal, and replacement of operations. The goal of optimizing reordering is to make the intermediate results created during the plan execution smaller. With respect to this, it is important to perform selective operations as early as possible. In graph terms, it means that these operations should be placed closer to the leaves of the graph (i.e. pushed down). Often, selective operations can be pushed down only along with a containing operation block. We present a blocks pushdown algorithm that enables selective operations pushdown.

We refer to *operation blocks* corresponding to *for-, let-, where-, etc. clauses* as *clause-blocks*. Two *clause-blocks* are independent if one clause of normalized query does not refer to a variable introduced in another.

1. Suppose $P = B_s(B_r)$ is a plan, where B_s and B_r are operation blocks and B_s is a block to push down.

2. Locate a *clause-block* B_m, $B_r = B_e(B_m(B_b))$: B_s depends on B_m. If there is no such B_m, then $B_e = B_r$.
3. Recursively perform the algorithm on B_m, obtaining B'_m.
4. Push down B_s through B_e in order to obtain: $P = B'_e(B_s(B'_m(B'_b)))$.

5.2. Redundant operation reduction

In this section we present logical optimizations that remove redundant operations or group several operations into complex ones. The optimizations include:

- removal of unnest operations;
- introduction of addition operations;
- removal of unnecessary index operations; and
- cycle removal.

Unnest removal is trivial and can be applied directly: $unnest_{h'}(P) \Rightarrow P$ if P does not contain $nest_{h''}$ and $h'' \subset h'$.

The **addition operation** is also introduced directly based on its definition (see section 3.2). The addition operation groups several operations (*nest, join, projection*) into a single one and allows further physical plan optimizations.

Removal of unnecessary index operations is a two-step operation that consists of preliminary reordering of index operations and subsequent index operations removal itself. After unnest removal and introdution of addition operation *let-blocks* are identified by the presence of addition operation. Such blocks are non-modifying operation blocks due to use of *left outer join* and *nest*. Consequently, if an index operation is performed before such block, indexes are retained in the result, and the following transformation is valid:

$$index_i(\pi_{he(P)\cup re(B)}B(index_j(P))) \Rightarrow \pi_{he(P)\cup re(B)\cup i}(B(index_i(index_j(P)))).$$

Subsequent application of this transformation produces a plan that contains sequences of *index* operations. The sequences are replaced with a single *index* operation.

Cycle removal is performed on for- and let-cycles that appear (as implied by the construction rules) whenever the respective bound expressions reference previously defined variables. The bound expressions can be one of three kinds: an xpath expression, a FLWOR expression, or a conditional expression.

If the bound expression of a for- or a let-clause is an xpath. Xpath expressions are mapped using *structural join* and *LPS* operations. Assume B_x is a block corresponding to some xpath. It consists of *structural join* and *LPS* operations. While *structural joins* are reducing operations, they have left outer versions ($\bowtie^l$) that are non-reducing. B_x^l is obtained from B_x by replacing *structural joins* with left outer joins.

We propose the following transformation for *let-cycles*:

$$\pi_{he(P)\cup re(B_x)}(nest_i\left(index_i(P) \bowtie^l_{i=j} (B_x\,(unnest_v(index_j(P))))\right)) \Rightarrow$$
$$\pi_{he(P)\cup re(B^l_x)}(nest_i\left(B^l_x(unnest_v(index_i(P)))\right)).$$

The transformation for *for-cycle* is:

$$\pi_{he(P)\cup re(B_x)}(index_i(P) \bowtie_{i=j} B_x\,(unnest_v(index_j(P)))) \Rightarrow$$
$$\pi_{he(P)\cup re(B_x)}(B_x(unnest_v(dup_v(P)))) \quad .$$

The topmost *join* operation in the original plans of the cycle is used to set up a correspondence between variable values of the query context with the new values obtained within the clause. *Unnest* operation can break the correspondence. To keep it, we use *dup* operation instead of *join* and then perform required unnesting on the introduced attribute. If the *unnest* operation is removed in optimization process, *dup* can also be eliminated.

If the bound expression of a let-clause is a nested FLWOR. The optimization is based on the fact that the topmost join $\bowtie^l_{i=j}$ that organizes a cycle can be removed if it has a non-reducing operation block as a right-hand operand.

After query normalization, each nested FLWOR starts with *for* and ends with *where*, *order by*, and *return*. The last three clauses are mapped with, respectively, σ, $sort$ and π operations. Let an operation block B_1 corresponds to *order by* and *return*, and B_2 corresponds to the sequence of *let-* and *for-clauses*. A plan for complete *let-clause* with a nested *FLWOR* is:

$$\pi_{he(P)\cup re(B_1)}(nest_i(index_i(P) \bowtie^l_{i=j} B_1(\sigma_p(B_2(index_j(P)))))). \qquad (1)$$

$\bowtie^l$ in this plan can be pushed down over B_1. The transformation also modifies *sort* operation to preserve original order.

Let $B_2 = B_{F_1}(\ldots B_{F_n})$, where B_{F_i} are *for-* or *let-blocks*.

There are two cases: $\forall i\ B_{F_i}$ is not dependent on P and $\exists i\ B_{F_i}$ is dependent on P. In case B_{F_i} is not dependent on P according to normalization rules, B_{F_1} is a *for-block*. According to construction rules and independence with P, $F_1 = P \times B_x$, where B_x is a block corresponding to a xpath expression. Since B_{F_2} is also independent of P, it is easy to see that $B_{F_2}(P \times B_{F_1}) = P \times B_{F_2}(B_{F_1})$. After application of this transformation sequentially for expression (1), combining σ and $\times$ into $\bowtie^l_p$, and removing outer $\bowtie^l_{i=i}$, we obtain:

$$\pi_{he(P)\cup re(B_1)}(nest_i(B_1(index_i(P) \bowtie^l_p B_{F_n}(\ldots(B_{F_1}))))).$$

In case B_{F_i} is dependent on P, we use the fact that *let-cycles* are non-modifying blocks (see section 5) and *for-cycles* are reducing due to their topmost *join* operations.

Let's denote $B^l_2 = B_{L_1}(\ldots B_{L_n})$, where $B_{L_i} = B_{F_i}$ if B_{F_i} is a *let-block* and $B_{L_i} = B^l_{F_i}$ if B_{F_i} is a *for-block*. $B^l_{F_i}$ is obtained by replacing its topmost *join* with *left outer join* in B_{F_i}. We perform (1) to the following:

$$\pi_{he(P)\cup re(B_1)}(nest_i(B_1(\sigma_p(B^l_2(index_i(P)))))).$$

Cycle removal is applied recursively for inner cycles.

If the bound expression of a let-clause is a conditional expression. If the bound expression is a conditional expression, transformations similar to those described for xpath expression case can be applied because the corresponding operation block is non-reducing. We omit details due to space limitations.

5.3. Transformations for lazy evaluation

Lazy evaluation is preferable in quantified expressions, positional predicates, and conditions of where, if-then-else, typeswitch clauses. In XAnswer, these types of expressions are mapped with ST-operations. When a physical plan is executed, the operands of the ST-operations are pre-evaluated, thus potentially making expensive unnecessary computations. To reduce such computations, certain support, e.g. pushing down of selective operations closer to leaves, is required in the process of algebraic plan modification.

Transformations of quantified expressions and positional predicates. Positional predicates allow physical plan execution to avoid the calculation of the whole operand of the operation. For quantified expressions, if one satisfying (or not satisfying - depending on whether “some” or “every” is used) value is found, there is no need to evaluate further the operand of the quantified operation. *Position* and *quant* operations are introduced into algebraic plans to allow pipelined evaluation at the physical level.

According to normalization and construction rules, positional predicates are mapped using selections and index operations. If $h_i = re(index_i)$ and p is a positional predicate $\sigma_{p(h_i)}$ is replaced with $position_{p(h_i)}$.

A similar mapping is used for quantified expressions. Unlike the case of positional predicates, the selection is performed over nested values. According to the proposed optimizations, the nestings are included in *addition* operations. Then, if $h_i = re(\bigoplus)$, an expression $\sigma_{p(h_i)}$ is substituted with $quant_{p(h_i)}$.

Transformations of conditions. Consider a where-clause *where A op B*. A and B are logical expressions or terminals (variable references). Op is a logical operator and or or. We assume without loss of genrality that A and B are terminals; if not, the process described below can be applied recursively. B_A and B_B are operation blocks corresponding to A and B respectively. In case of *if-then-else* we use B_C, B_T and B_F to denote corresponding operation blocks for the *condition*, *then-*, and *else-branches*. We assume that other blocks in the plan are independent with A and B. Without this assumption lazy evaluation does not bring performance benefits and there is no much point in applying this transformation.

Transformation of conditions is preceded by sequential pushdown of B_A, B_B (*where-clause*) or B_C, B_T, B_F (*if-then-else-clause*) using the algorithm described in section 5.1.

Logical “and” in a where-clause. Selection decomposition is:

$$\sigma_{A\&B}\left(B_B(B_A(P))\right) = \sigma_A\left(\sigma_B(B_B(B_A(P)))\right).$$

Due to normalization, B_A and B_B are independent blocks.

Blocks B'_A, B'_B, P' (possibly empty) are obtained by pushdown of B_A and B_B according to the algorithm in section 5.1: $B_B(B_A(P)) \Rightarrow B'_B(B_B(B'_A(B_A(P'))))$. B'_B has blocks depending on it, all blocks are independent with B'_A. We also assume that B'_B contains *sort* operation that restores original order of tuples. The transformation result is:

$$B'_B(\sigma_B(B_B(B'_A(\sigma_A(B_A(P')))))).$$

Logical "or" in a where-clause. Since $A \cup B = A \cup (B \backslash A)$, the selection decomposition is:

$$sort_i[\pi_{he(P)\cup i}\,(\sigma_A(B_B(B_A(index_i(P)))))$$
$$\cup\pi_{he(P)\cup i}\,(\sigma_{\neg A\&B}(B_B(B_A(index_i(P)))))].$$

Block pushdown then gives us:

$$B'_B(\pi\,(\sigma_A(B_A(P'))) \cup \pi\,(\sigma_B(B_B(B'_A(\sigma_{\neg A}(B_A(P')))))).$$

An algebraic expression for *if-then-else clause case* is:

$$sort_i[HR_{(re(B_T)\to k)}(\sigma_C(P'')) \cup HR_{(re(B_F)\to k)}(\sigma_{\neg C}(P''))],$$

where $P'' = B_F(B_T(B_C(index_i(P))))$.

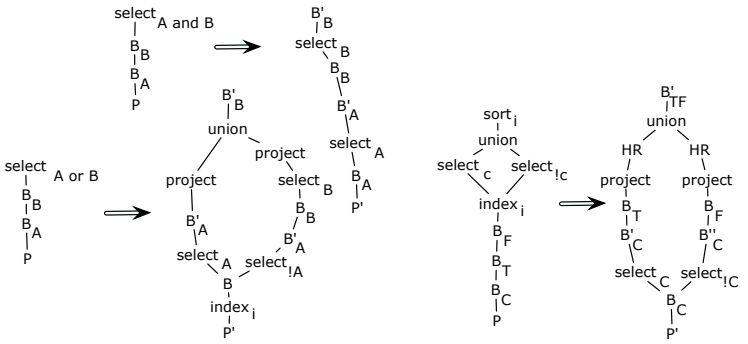
Blocks pushdown results in:

$$B'_{TF}\,(HR(\pi(B_T(B'_C(\sigma_C(B_C(P'))))))) \cup HR(\pi(B_F(B''_C(\sigma_C(B_C(P'))))))\,.$$

The blocks B'_C, B''_C, B'_{TF}, and P' are obtained by pushing down B_C, B_T, and B_F blocks. Only B_T is not independent with B'_C and B_F is not independent with B''_C. As with B'_B in the previous case, B'_{TF} contains *sort* operation that restores original order of tuples.

Similar optimization, whose description we omit here, is used for *typeswitch-clauses*.

The proposed optimizations are presented on the Figure 3.



(a) *Where-clause* decomposition **(b)** *If-then-else* decomposition

Fig. 3. Optimizations for lazy evaluation

6. Experiments

Our experimental implementation of XAnswer is based on XML DBMS eXist [1]. EXist storage provides DLN-indexes [12] as unique ids for all nodes and all

structural operations can be evaluated using just the ids. Thus, envelopes were implemented as arrays of tuples that contain either atomic values (e.g. string, integer), ids, or sequences of the ids or atomic values.

The code was written in Java and experiments were conducted under Windows XP on Intel Dual Core T2300 @1.6 GHz with 2Gb of main memory. The XMark benchmark [3] was used as data set.

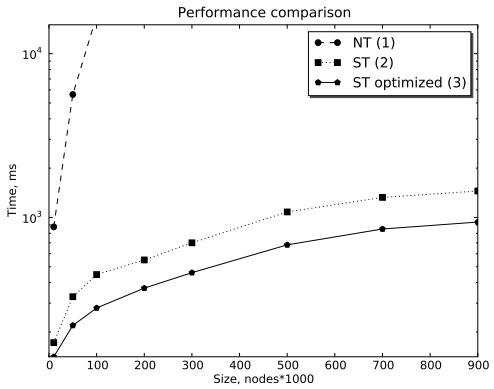


Fig. 4. Experiment E_1

Known approaches can perform unnesting for relatively simple queries with nested FLWORs. That kind of optimization allows switching from NT to ST-execution, which can bring dramatic performance gain, since it allows the hash join algorithm to be used instead of nested loops. Using normalization techniques similar to ours, known approaches achieve performance comparable to XAnswer. However, known approaches cannot perform unnesting when the query contains quantifiers, where-clauses, or conditional clauses with nested FLWORs. Moreover, many such cases require lazy evaluation.

To compare other approaches with ours, we implemented NT-style plan creation for conditional clauses as well as ST-operations of XAnswer. In experiments we focused only on such queries over the XMark's data set. We present details for only two experiment sets. The first set (E_1) evaluates performance of naive NT-execution (the default plan generated by eXist), ST (initial XAnswer plan), and optimized-ST (optimized XAnswer plan) on a typical XMark query (such as query Q8) where earlier approaches can perform unnesting.

The second set (E_2) evaluates performance of naive NT-execution (the default plan generated by eXist), partial NT (having NT-style condition operation and ST for clauses without conditions and quantifiers), ST (XAnswer plan without lazy evaluation), and optimized-ST (optimized XAnswer plan) for a query with a conditional clause that requires lazy evaluation.

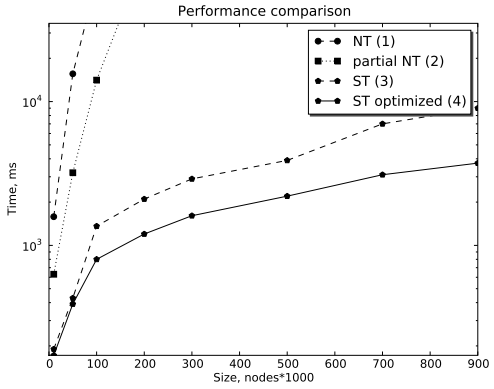


Fig. 5. Experiment E_2

Query 1 A query example from the set E_2

```
let $doc := doc('doc.xml')
for $ca in $doc//closed-auction
let $b :=
  for $p in $doc//person
  where $p/@id = $ca/buyer/@person
  return $p
return
  if ($ca/price >200)
  then (
    let $c := \b/country
    let $n := $b/name
    return <rb>{$c,$n}</rb>)
  else (
    let $item :=
      for $it in $doc//regions//item
      where $it/@id=$ca/itemref/@item
      return $it
    let $l := $item/location
    let $n :=$item/name
    return <ri>{$l,$n}</ri>)
```

Figure 4 demonstrates performance comparison for E_1. The huge performance gain of ST-plan compared to NT is observed due to use of hash join over input sets instead of evaluating inner expression in a loop. Since the query has a relatively simple where-clause that does not require lazy evaluation, the ST-optimized plan does not demonstrate major performance gain. The increase is mostly achieved through removal of a cycle and use of addition operation, which reduces amount of hash operations in the physical plan.

Figure 5 demonstrates performance evaluation for E_2. Note, that the time axis has a logarithmic scale. Since partial NT-plan utilizes ST-execution for the first nested FLWOR, the performance gain compared to naive NT is similar to the one obtained in E_1. But NT-style condition leads to evaluation of nested expressions in a loop, which causes difference in performance with the ST plan. The gain of the optimized-ST is due to the smaller sizes of intermediate results. The sizes are smaller because evaluation of the first nested FLWOR *then*- and *else*- branches happens only when teh condition has corresponding values: for example FLWOR and *then* are evaluated only when $price > 200$ in line 7 of the code in Query 1.

7. Conclusion

Heavy query workloads in modern applications require algebras that support cost-based optimization. Set-at-a-time execution is a natural way to support cost estimation. While set-at-a-time execution appeared in earlier work on efficient XQuery processing techniques, known algebras combine it with node-at-a-time to cover the complete set of XQuery operations.

In this paper we presented an algebra with only set-at-a-time operations that supports compilation of complete XQuery except recursive functions. We introduced optimization rules eliminating unnecessary operations and enabling lazy evaluation. It was experimentally shown that for complex queries with conditional and quantified expressions the proposed methods yield plans with much better performance than those that can be obtained with known algebras.

References

1. Exist DBMS Home Page. http://exist-db.org/, exist DBMS Home Page
2. Sedna DBMS Home Page. http://modis.ispras.ru/sedna/, sedna DBMS Home Page
3. XMark Benchmark Home Page. http://monetdb.cwi.nl/xml/, xMark Benchmark Home Page
4. Chean, et al.: From tree patterns to generalized tree patterns: On efficient evaluation of xquery. In: VLDB. pp. 237–248 (2003)
5. Deutsch, A., et al.: The next framework for logical xquery optimization. In: VLDB '04: Proceedings of the Thirtieth international conference on Very large data bases. pp. 168–179. VLDB Endowment (2004)
6. Fankhauser, P.: Xquery formal semantics state and challenges. SIGMOD Rec. 30(3), 14–19 (2001)
7. Ioannidis, Y.: Query optimization. ACM Comput. Surv. 28(1), 121–123 (1996)

8. Ioannidis, Y.: The history of histograms. In: VLDB '2003: Proceedings of the 29th international conference on Very large data bases. pp. 19–30. VLDB Endowment (2003)
9. Jagadish, H., et al.: Tax: A tree algebra for xml. In: DBPL '01: Revised Papers from the 8th International Workshop on Database Programming Languages. Springer-Verlag, London, UK (2002)
10. Jagadish, V., et al: Timber: A native xml database. The VLDB Journal 11(4), 274–291 (2002)
11. May, N., et al.: Nested queries and quantifiers in an ordered context. Proc. ICDE Conference, Boston, USA pp. 239–250 (2004)
12. Meir, W.: Index-driven xquery processing in the exist xml database (2006)
13. Re, C., et al.: A complete and efficient algebraic compiler for xquery. In: ICDE '06: Proceedings of the 22nd International Conference on Data Engineering. p. 14. IEEE Computer Society, Washington, DC, USA (2006)
14. Sartiani, C., Albano, A.: Yet another query algebra for xml data. In: IDEAS '02: Proceedings of the 2002 International Symposium on Database Engineering and Applications. pp. 106–115. IEEE Computer Society (2002)
15. Wang, S., et al.: Optimization of nested xquery expressions with orderby clauses. In: ICDEW '05: Proceedings of the 21st International Conference on Data Engineering Workshops. IEEE Computer Society (2005)
16. Weiner, A., Mathis, C., Haerder, T.: Towards cost-based query optimization in native xml database management systems. In: Proceedings of the SYRCoDIS 2007 Colloquium on Databases and Information Systems (2008)
17. Extensible Markup Language (XML) 1.0 (Second Edition) (6 October 2000), w3C Recommendation
18. XML Query Algebra (15 February 2001), w3C Recommendation
19. Zhang, X., Pielech, B., Rundesnteiner, E.: Honey, i shrunk the xquery!: an xml algebra optimization approach. In: WIDM '02: Proceedings of the 4th international workshop on Web information and data management. pp. 15–22 (2002)

Maxim Lukichev is Sr. Software Engineer at Whodini, Inc. Previously Maxim was a researcher at HP Labs Russia and was a member of Information Management Research group at St.Petersburg University. Maxim received his Masters and Ph.D. in Comp.Sci. from St.Petersburg University. His research interests are in the area of query optimization, indexing and data structures, semantic data and information extraction.

Boris Novikov is a professor and a head of Operations Research Laboratory and Information Management Research Group at St.Petersburg University. He also works as a database expert for the industry. Boris received his PhD in Comp.Sci.in 1981, and Dr. Sci. degree in 1994 from St.Petersburg University. His research interests include design of database systems, transactions, indexing and data structures, query processing and optimization, performance tuning, and management of distributed heterogeneous information resources.

Pankaj Mehra is SVP and CTO at Whodini, Inc. Pankaj also serves as an industry advisor to CodeX, a joint program of Stanford Law School and Computer

Science. He is on the editorial board of IEEE Internet Computing magazine. Previously Pankaj was a Chief Scientist of HP Labs Russia, and before that faculty member at IIT Delhi. Pankaj received his B Tech from IIT Delhi, and his Ph.D. in Computer Science from University of Illinois at Urbana-Champaign.

Received: August 4, 2010; Accepted: July 1, 2011.

DOI 10.2298/CSIS100929039E

A Decision Support Method for Evaluating Database Designs

Erki Eessaar[1] and Marek Soobik[1]

[1] Department of Informatics, Tallinn University of Technology,
Raja 15, 12618 Tallinn, Estonia
Erki.Eessaar@ttu.ee, mareksoobik@gmail.com

Abstract. It is possible to produce different database designs based on the same set of requirements to a database. In this paper, we present a decision support method for comparing different database designs and for selecting one of them as the best design. Each data model is an abstract language that can be used to create many different databases. The proposed method is flexible in the sense that it can be used in case of different data models, criteria, and designs. The method is based on the Analytic Hierarchy Process and uses pairwise comparisons. We also present a case study about comparing four designs of SQL databases in case of PostgreSQL™ database management system. The results depend on the context where the designs will be used. Hence, we evaluate the designs in case of two different contexts – management of measurements data and an online transaction processing system.

Keywords: database design, decision support, Analytic Hierarchy Process, object-relational database, SQL.

1. Introduction

Database design process consists of conceptual design, logical design, and physical design according to a well-known methodology [25]. During conceptual database design, one has to capture an accurate representation of reality. During logical design, one has to describe the design of a database in terms of a data model (for instance, the underlying model of SQL database language) but without taking into account the database management system (DBMS), based on which the database will be implemented. During physical design, one has to design a database by taking into account the DBMS where the database will be implemented.

Date [8, p. 287] is in position that "Database design is still largely subjective in nature". Normalization theory [6] and the principle of orthogonal database design [10] are the main examples of the use of scientific principles in the context of relational database design.

The goal of the paper is to present a systematic and structured method for the evaluation of database designs and for the selection of the best database

design from a set of designs. The designs, which one can evaluate, are the results of logical or physical design. The method is a *multi-criteria decision support method* that helps database developers to make more informed decisions during database development and hence improve the quality of databases.

Simsion [22, p. 301] reports the results of a constrained exercise, according to which different data modelers produce "a wide range of different workable logical data models in response to a common 'conceptual' data model". Therefore, we need a method, based on which to compare the logical data models and select or reject them. Teorey, Yang, and Fry [25] suggest a relation refinement step before physical design, the goal of which is to find more efficient and adaptable database schemas without the loss of data integrity. The result of this step would be a set of alternative logical structures that should be considered during the physical design. We need a method that would help us to select the best design from this set during physical design.

In addition, this kind of method is necessary because new data models (in the sense of abstract language) provide more and more features. For instance, Soutou [23] writes that if one uses the underlying data model of the SQL-92 standard, then it is possible to use two different designs to represent one-to-many relationships. On the other hand, if one uses the underlying object-relational data model of the SQL:1999 standard, then it is possible to use twelve different designs to represent one to many relationships [23]. It complicates the work of database designers because the number of alternative designs increases and designers have to select the best design from this set. Feuerlicht, Pokorný, and Richta [12] discuss the use of object-relational features that are specified in the SQL:2003 standard. They conclude that "numerous design options exist that need to be evaluated in the context of specific application requirements" [12, p. 986]. They also note that there is a lack of database design methodologies that help designers to make informed decisions about design choices.

In our previous work [11], we proposed a set of criteria that one could use to evaluate database designs and evaluated two SQL database designs (the regular design and the universal design) in terms of different criteria. In this paper, we extend the work by proposing a generalized evaluation method and present the results of an evaluation of four different SQL database designs.

The rest of the paper is organized as follows. Firstly, we describe related work in the field of decision support in case of database design. Secondly, we present an evaluation method of database designs. We also present a metamodel of the method that can be used as a basis to create a software system that assists users of the method or can be used to record the results of existing evaluations. Thirdly, we use the method to evaluate four SQL database designs in case of PostgreSQL™ 8.3.6 DBMS. Finally, we draw conclusions and point to the future work.

2. Related Works

The idea of using decision support methods during database design is not new. March [16] explains the techniques of mathematical clustering, iterative grouping refinement, mathematical programming, and hierarchic aggregation, which can be used to determine efficient physical organization of data for a database. The method is usable in case of hierarchical or network data models. On the other hand, in this paper, we propose a method, which can be used in case of any data model and can be used to select the best physical as well as logical organization of data in a database.

There exist selection methods for specific types of database objects. For instance, Theodoratos and Bouzeghoub [26] propose a general algorithm for selecting a set of materialized views (snapshots) for a data warehouse so that the selected set satisfies all the given constraints and minimizes the operational cost. They use "AND/OR dag representation for multiple queries and views". [26, p. 7] On the other hand, our proposed method is not limited with a specific type of database objects (like materialized views) or specific type of databases (like data warehouses).

Svahnberg et al. [24] describe a decision support method that is based on the Analytic Hierarchy Process (AHP) [21] and can be used to compare software architecture candidates. In this paper, we apply an AHP-based method to compare database designs. Svahnberg et al. [24] use only subjective judgments of a set of professional software developers to compare alternatives in terms of criteria. On the other hand, our method proposes the use of *measurements* to compare alternatives in terms of criteria. The goal is to increase the objectivity of evaluation results. Park and Lim [18] present an AHP-based method for comparing user interface designs. It is similar to our method because they use the results of usability measurements to compare designs pairwise in terms of usability criteria.

Vaidya and Kumar [29] describe application of AHP in the field of selection, evaluation, benefit cost analysis, allocation, planning and development, priority and ranking, decision making, and forecasting. Some applications (like software selection or evaluation of quality of software systems) are from the field of software engineering. None of the applications is used in the context of database design. However, the research of Vaidya and Kumar [29] demonstrates that AHP is suitable and widely used decision support method that could be used to evaluate database designs as well.

Chaudhuri and Narasayya [4] review the state of the art in the field of self-tuning DBMSs. Similarity with our method is that both of them are used to find the best design. In case of our method, the evaluators are database designers who must take into account the results of evaluation while designing a database. On the other hand, self-tuning DBMSs have system-defined capabilities that take into account rules, external cost models, or the database statistics and automatically make modification in the internal schema of the database. The systems use performance of database operations and data size as the main criteria. Our proposed method can be

used in case of logical design as well as physical design and permits the use of wider range of criteria.

3. An Evaluation Method of Database Designs

3.1. The Analytic Hierarchy Process

The Analytic Hierarchy Process (AHP) [21] allows us to make decisions by modeling a complex problem as a hierarchical structure. The levels of this model are from top to bottom goal, objectives, and alternatives. The goal is the overall objective. Objectives correspond to criteria that one has to take into account by comparing alternatives. There can be multiple levels of criteria. Alternatives are objects, between which the choice will be made. The process consists of comparing objectives pairwise to find the relative importance of the criteria in terms of the goal. In addition, one has to compare alternatives pairwise in terms of each criterion. For the pairwise comparisons one has to use a nine-point scale [21]. Comparison of elements *i* and *j* answers a question, which of them is more important and how much more important it is [24]. The results are combined to calculate the final score of each alternative. The alternative with the highest final score is the best in terms of the goal. For instance, the goal might be to select the most suitable software architecture for a particular context [24].

3.2. Description of the Method

In this paper, we propose an AHP-based method for evaluating database designs. It consists of the following consecutive steps.

1. Description of the goal of the evaluation.
2. Description of the context where the designs will be used.
3. Selection of alternatives (database designs or pairs of database designs and platforms). A platform is a version of a DBMS that is used to implement a database.
4. Selection of criteria, based on which the alternatives will be evaluated.
5. Selection of software measures, based on which the measurements will be made. If a criterion *c* is associated with a software measure, then the results of measurements of alternatives will be used to make pairwise comparisons of the alternatives in terms of *c*. If a criterion *c* does not have a suitable software measure, then it is possible to use subjective opinions of one or more evaluators to make pairwise comparisons of alternatives in terms of *c*. We prefer criteria with associated software measures because the results of measurements help us to increase the objectivity of comparison results.

6. Specification of tasks in a database, based on which the measurements will be made. In this context each task is a problem that has to be solved in a test database to measure the alternatives. Tasks may have subtasks. Tasks have different solutions in case of different alternatives. Let *e* be an evaluation and A is the set of alternatives that are evaluated during *e*. All the measurements for evaluating alternatives in A in terms of a criterion *c* must be based on the same task. Otherwise the measurement results are not comparable. In addition, all the measurements for evaluating alternatives in A in terms of a criterion *c* must use the same protocol to perform measurements. For instance, if we count the number of physical lines of code in case of a criterion *c*, then we must use the same rules to format the code in case of all the alternatives in A.
7. If there are one or more tasks that are specified as the result of the previous step, then it is necessary to implement the designs in a test database (or in more than one test database if the alternatives are pairs of designs and platforms) and generate test data. If possible, one should use a public and well-known specification that contains requirements to the test data as a basis of implementing the test database. If the specification is public, then it is easier to repeat the experiments. If the specification is well-known, then it has probably been carefully evaluated.
8. Performing the tasks and measuring the results based on the test database (or databases).
9. Pairwise comparison of the criteria and calculation of the relative importance of the criteria. This step is still subjective in nature but explicitly defined context should simplify it. If a criterion *c* at the level N has one or more sub-criteria *c1',...,cn'* at the level N+1, then it is also necessary to calculate the relative importance of the sub-criteria in terms of *c*.
10. Pairwise comparison of the alternatives based on each criterion that is on the lowest level of the hierarchy of criteria.
11. Calculation of the final scores of the alternatives in terms of the goal by using AHP.

Fig. 1 and Fig. 2 present a metamodel of the method. We use UML class diagrams to present the metamodel. The method does not produce absolute judgments of the alternatives. Instead, it produces judgments that depend on a particular context, criteria, measures, tasks, solutions, and alternatives.

Each database and database management system (DBMS) consists of external, conceptual, and internal levels according to the ANSI/SPARC architecture of DBMSs [6]. External and internal levels are closest to the users and physical storage, respectively. Conceptual level is between these two and represents the entire information content of the database [6]. Each level has one or more corresponding schemas in a database. We use the concept "database design" quite loosely to denote a specification of a set of elements that belong to the external schemas, the conceptual schema, or the internal schema of a database. We do not prescribe the size (the number of elements) of designs that one can evaluate by using the method.

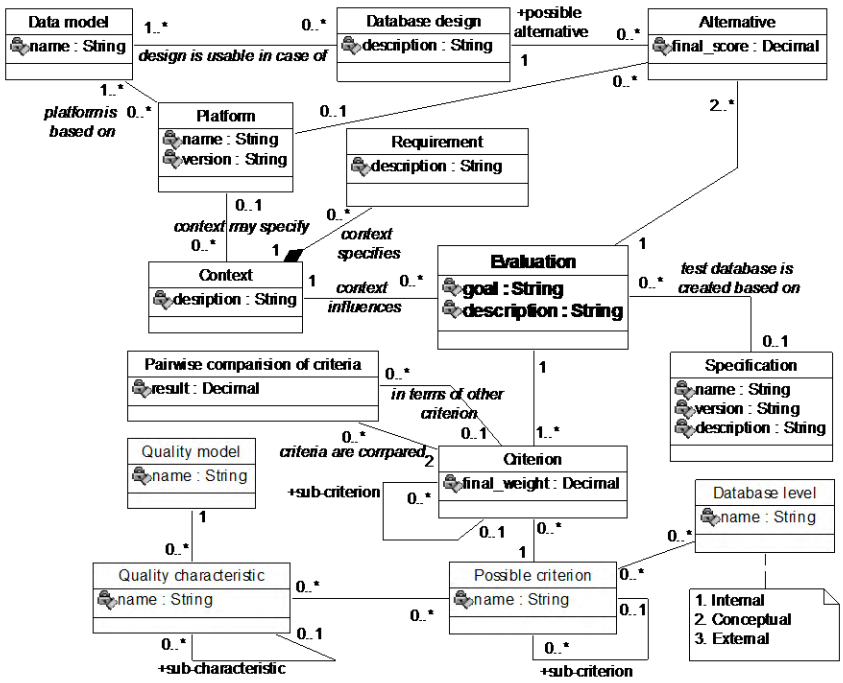


Fig. 1. A fragment of a metamodel of an evaluation method of database designs

Each database design is usable in case of one or more data models and platforms (versions of a DBMS). An example of data model is the underlying object-relational data model of the SQL:2003 standard [17]. We denote it as the OR_{SQL} data model. Each platform has one or more associated data models, based on which the database languages of the platform have been created.

Each evaluation has the goal (for instance, find the best design from the set of given designs). Each evaluation must take into account the context where the database, and hence the designs, will be used. Each context is a set of requirements to the database. For instance, a context could specify that up to 1000 users use the database at the same time, queries should be answered within one second, and there are at least 10 different roles of database users. Based on these requirements one can conclude that in this context the best design 1) must facilitate implementation of concurrency control, 2) must support high performance of database operations, and 3) must allow administrators to grant/revoke permissions in the database as easily as possible.

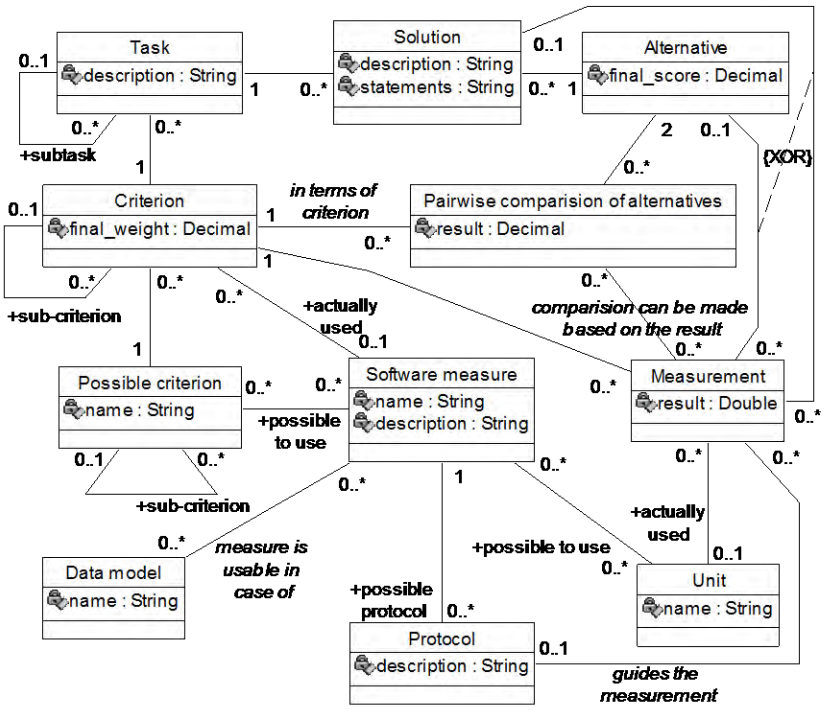


Fig. 2. A fragment of a metamodel of an evaluation method of database designs

Each evaluation has one or more associated criteria and at least two associated alternatives. The number of alternatives is limited by the amount of time and effort that is needed to develop them. According to Forman and Selly [13, p. 38], "humans are not capable of dealing accurately with more than about seven to nine things at a time" and therefore the number of alternatives and the number of highest-level criteria should not be bigger than nine. Criteria form a hierarchical structure. For instance, criterion "Performance of data manipulation operations" can have a sub-criterion "Performance of data manipulation operations that search data about one entity". The number of immediate sub-criteria of a criterion should also not be bigger than nine. Sets of alternatives, criteria, and measures that are actually used during an evaluation are subsets of possible alternatives (database designs or pairs of database designs and platforms), criteria, and measures.

If one evaluates logical designs and does not use criteria that require the creation of a test database, then it is not necessary to determine the platform, based on which the database will be created. On the other hand, if one has to perform tasks (for instance, measure performance of database operations) based on a test database, then one must bear in mind that different platforms provide different means to implement the designs and it influences the results

of evaluation. It is possible that all the designs, which are considered during an evaluation, will be implemented by using the same platform. In this case, the information about the platform is a part of the context. It is also possible that the alternatives are pairs of platforms and database designs.

Some measurements do not require the creation of a test database. For instance, one can calculate the schema size of a SQL database without actually implementing the schema in a database.

Behkamal, Kahani, and Akbari [2] propose to find criteria for evaluating software systems based on quality models, like the one defined in the ISO 9126 standard. It is also possible in case of the proposed method. For instance, criteria "Access control", "Integrity constraints", and "Performance of data manipulation operations" correspond to the ISO 9126 quality model sub-characteristics "Security", "Changeability", and "Timebased efficiency", respectively. In addition, we suggest the use of database levels as a basis to systematically search and select the criteria. Each criterion corresponds to zero or more database levels, which are defined by the ANSI/SPARC architecture of DBMSs. For instance, criterion "Performance of data manipulation operations" corresponds to the internal level because it depends on the stored record types and their physical sequence, indexes etc. All these elements are described by the internal schema of a database.

During each evaluation one has to compare criteria pairwise to find their relative importance (weights) in terms of the goal. The requirements, which are associated with the context of an evaluation, determine the relative importance of the criteria in case of the evaluation. In addition, one has to perform measurements to find values of software measures in case of different alternatives. Each software measure is usable in case of zero or more data models. For instance, Piattini et al. [19] present a set of measures that are usable in case of the OR_{SQL} data model. Let us assume that a software measure *m* is associated with a criterion *c*. The results of the measurements based on *m* will be used during the pairwise comparison of alternatives in terms of *c*. If a possible criterion has more than one associated software measure, then evaluator uses one the measures for the evaluation.

4. Application of the Evaluation Method

In this paper, we demonstrate the use of the method by comparing four database designs, the underlying data model of which is the OR_{SQL} data model. We implement all the designs based on open source PostgreSQL™ 8.3.6 DBMS [20]. The goal of the evaluation is to find the best design in case of two different contexts.

In this study, we use a subset of the database that is proposed in the TPC Benchmark™ C [28] to create the test database. The entire database is for the wholesale supplier company that has a number of geographically distributed sales districts and warehouses. The conceptual model of the subset of the database specifies entity types *Customer*, *Order*, *Order_line*,

and *Stock*. The semantics of data does not affect the measurements that we plan to use during this evaluation.

4.1. Contexts

The relative importance of criteria depends on the context in which the database will be used. In this study, we evaluate designs in terms of two *hypothetical* contexts.

Context 1. It describes a system that deals with the management of measurement data. The system has a small number of users – up to two users register data and up to five users perform complex statistical queries. Publication of its data may cause material or moral harm. The system must answer queries within minutes. However, if the system does not answer queries, then it does not cause remarkable consequences. All the unauthorized modifications of data must be detectable. In the future there may be additional types of measurements, the resulting data of which the system has to manage. The system must prohibit registration of seemingly incorrect data and therefore it must be as easy as possible to enforce integrity constraints at the database level.

Context 2. It describes a customer management system of a big retail company, which is an example of an online transaction processing system. The system has thousands of users. Most of the queries, which are executed in the database, help users to find information about a particular customer. Performance of data modification operations is less important. Publication of the data in the database disrupts functioning of the company or violates the privacy of people. The system must answer queries within seconds. If it does not answer queries, then it causes disruption of the functioning of the company. It must be possible to identify the source of all the data in the database. The requirements to the database are fixed for the next three years. There are relatively few business rules, based on which one has to create integrity constraints in the database.

4.2. Alternatives

Next, we explain the four database designs, which are the alternatives in our evaluation.

The regular design. According to this design one has to create a separate base table (table in short) based on each entity type (*Customer*, *Order*, *Order_line*, and *Stock* in our case) that is specified in the conceptual data model. We call the design regular because it is widely used and seems natural. Fig. 3 describes tables that are created according to the regular design.

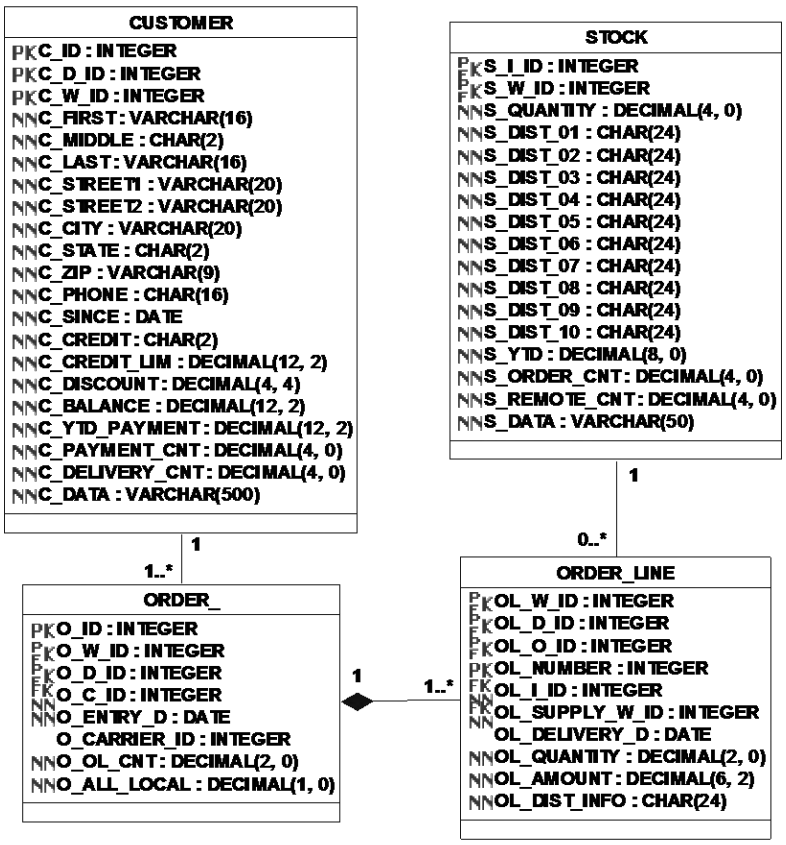


Fig. 3. An example of the use of the regular design (adapted from [28])

The universal design. This is a highly generic database design, according to which all the data in a database is represented in terms of *Object_types*, *Objects*, *Attributes*, *Attribute_values*, and *Relationships*. For instance, Hay [14] and Blaha [3] refer to this kind of design as "Universal Data Model" and "Softcoded Values", respectively. Data about *Object_types* and *Attributes* defines legal *Attribute_values* that can be associated with *Objects*. Table *Attribute_value* has a set of columns, the specification of which has the general form: <<*data_type_name*>>_ *data_type_name*. These columns allow us to record values that have different types. The number of these columns and their data types depend on a platform (version of a DBMS) where this database is created. Fig. 4 describes tables that are created according to the universal design. We consider this design because it gives an impression of great flexibility.

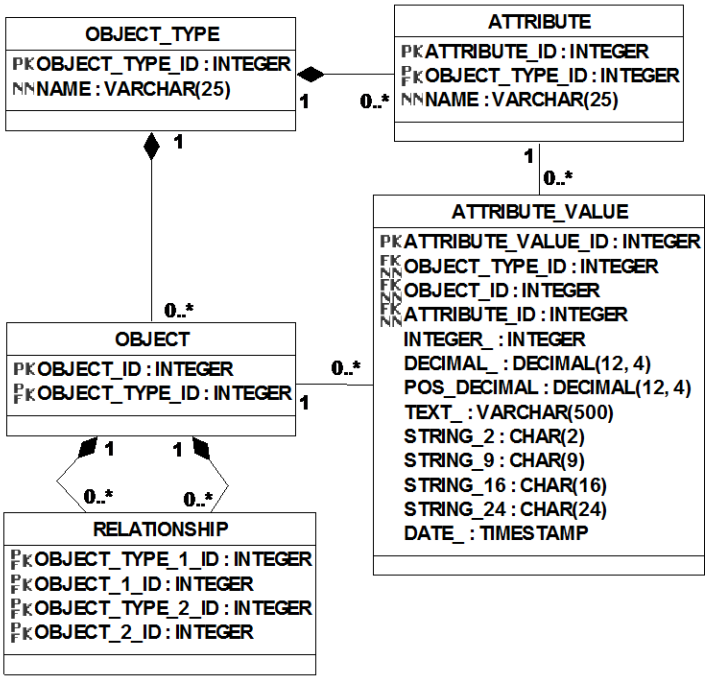


Fig. 4. The universal design

The entity-attribute-value with classes and relationships (EAV/CR) design. It is a variation of the universal design. Each supported data type should have exactly one corresponding table for recording attribute values with this type according to the EAV/CR design [5]. This is different from the universal design where is one generic table *Attribute_value* that has a column for each supported data type. Fig. 5 describes some tables that are created according to the EAV/CR design. We do not present all the tables that we have created based on different data types on Fig. 5. We consider the EAV/CR design because it gives an impression of great flexibility and is a widely used variation of the universal design.

The sixth normal form (6NF) design. Table T is in 6NF if and only if it cannot be nonloss decomposed at all (other than the identity projection of T)[7]. Date [7] also notes that the identity projection of a table T is the projection over all of its columns. Let us assume that a conceptual data model of a database specifies entity type E with attributes *a1, ...,an*. There is one table for each attribute *a1, ...,an* in the database, which is created according to the 6NF design. All tables, which are created according to this design, consist of columns that form the key plus at most one additional column that is not part of the key.

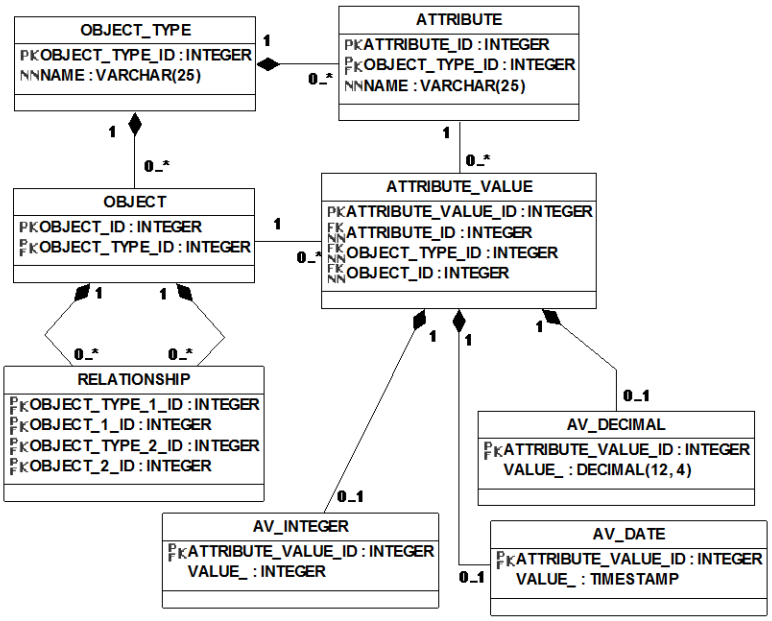


Fig. 5. The EAV/CR design

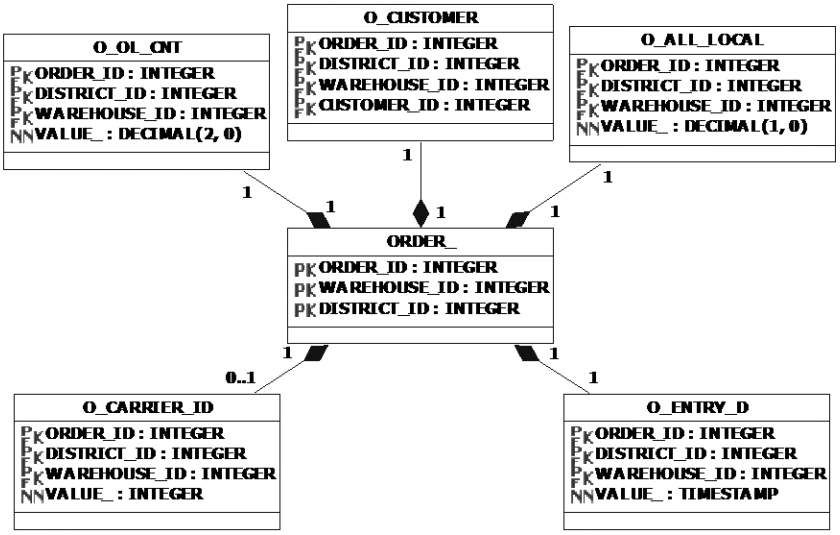


Fig. 6. An example of the use of the 6NF design

Date, Darwen, and Lorentzos [9] propose to use a similar design to record temporal data. The design also allows us to prevent the use of NULLs to present missing information [8]. If an attribute does not have a value, then there is no corresponding row in the table that is created according to the attribute. Fig. 6 describes some tables that are created according to the 6NF design.

4.3. Criteria

We wanted to evaluate the designs from different perspectives and to use measurements for that purpose. Therefore, we selected the criteria in a way that each level of the ANSI/SPARC database architecture has at least one corresponding criterion and each criterion has at least one associated software measure. Next, we present the names of the selected criteria together with their corresponding level.

1. External level: Complexity of queries, Access control.
2. Conceptual level: Schema size, Integrity constraints.
3. Internal level: Performance of data manipulation operations, Data size, Concurrency control.

We stress that the proposed criteria are not the only possible criteria to evaluate OR_{SQL} database designs.

Next, we describe for each level its associated criteria and introduce the software measures that we will use to evaluate database designs in terms of these criteria. In case of each selected measure *m* – the smaller is the measurement result in case of an alternative *a*, the better is *a* in the context of the criterion that is associated with *m*.

In this paper “small number of entities” means between two and five (endpoints included) and “large number of entities” means more than five.

External level. *Complexity of queries*. At the external level different views on a database are specified. One has to form queries to create views and hence the complexity of these queries is very important criterion at the external level. It is possible to represent queries as graphs where nodes are table aliases and arcs represent join and semijoin operations [27]. We evaluate the complexity of queries by calculating *Coefficient of Network Complexity* CNC=(A×A)/N where N is the number of nodes and A is the number of arcs [15]. We decided to find *the total complexity* of three different types of queries: 1) query that finds data about one entity, 2) query that aggregates data about small number of entities, and 3) query that aggregates data about large number of entities. The total complexity of a set of queries Q is the sum of complexities of queries that belong to Q. We decided to calculate the total complexity to avoid the distortion of the results that is caused by the selection of queries that favor one or another design.

Access control. We decided to count the physical lines of source code that are needed to grant SELECT (read) privileges to roles in order to evaluate designs in terms of access control. The criterion "Access control" has two sub-criteria in our evaluation.

1. Complexity of granting SELECT privileges to all the columns that correspond to attributes of one entity type in a conceptual data model.
2. Complexity of granting SELECT privileges to the columns that correspond to a proper subset of attributes of one entity type in a conceptual data model.

The style of writing code influences the count of physical lines. Therefore, we have to follow rules to format the code. For instance, we follow the rules that FROM and WHERE clauses start from a new line in case of a SELECT statement and the lines of code cannot be longer than 60 symbols.

Conceptual level. *Schema size*. We use software measure Schema Size to evaluate the designs in terms of schema size criterion. We assume that smaller Schema Size value means simpler database structure and hence better maintainability of the database. Piattini et al. [19] define Schema Size measure "as the sum of the tables size (TS) in the schema" [19, p. 7] and Table Size measure "as the sum of the total size of the simple columns (TSSC) and the total size of the complex columns" [19, p. 6]. The size of each simple column is one. All the columns are simple columns in case of the designs in this evaluation. Therefore, in this case Table Size of a base table T is equal to the total number of columns in T.

Integrity constraints. Each type of integrity constraints in SQL databases has a corresponding sub-criterion of criterion "Integrity constraints" in our hierarchical decision model. Some of these sub-criteria have additional sub-criteria.

1. Complexity of enforcing CHECK constraints. The sub-criteria:
   1) Complexity of enforcing constraints that involve one attribute of one entity type and 2) Complexity of enforcing constraints that involve more than one attribute of one entity type.
2. Complexity of enforcing UNIQUE constraints.
3. Complexity of enforcing NOT NULL constraints.
4. Complexity of enforcing FOREIGN KEY constraints. The sub-criteria:
   1) Number of Foreign Keys (NFK), 2) Referential Degree (RD), and
   3) Depth of Referential Tree (DRT).

In case of CHECK, UNIQUE, and NOT NULL constraints, we count the physical lines of source code that are needed to implement these constraints. In case of foreign key constraints, we use three schema level measures for evaluating object-relational database designs. Measure NFK is the number of foreign keys in the database schema [1]. Measure RD is the average referential degree over all the base tables. Baroni et al. [1, p. 34] define RD measure of a single table as "number of foreign keys in a table divided by the number of attributes of the same table". Measure DRT is defined as the longest referential path between tables in the database schema [19].

Internal level. *Performance of data manipulation operations* (*Performance of operations* in short). Criterion "Performance of data manipulation operations" has five sub-criteria in our evaluation. They correspond to different types of operations that one could perform in a database.

1. Performance of a query that finds data about one entity.

2. Performance of a query that aggregates data about a small number of entities.
3. Performance of a query that aggregates data about a large number of entities.
4. Performance of an operation for inserting data about one entity to a database.
5. Performance of an operation for modifying data about a small number of entities.

We measure performance in milliseconds. In case of each pair of a design and a sub-criterion, we perform the same task repeatedly and calculate median of the results.

Data size. The criterion "Data size" has two sub-criteria in our evaluation.

1. Size of base tables.
2. Size of indexes.

We measure the size by using PostgreSQL™ system-defined function *pg_relation_size*. The function has one parameter, the expected value of which is the name of a base table or an index. The function returns the size of the base table or index in bytes.

Concurrency control. Criterion "Concurrency control" allows us to evaluate how much effort is needed for locking data in a database to prevent concurrent data changes that cause inconsistency of data. We perform the task of changing attribute values of one entity to evaluate designs in terms of "Concurrency control". We count the physical lines of source code that are needed to implement the locking of data of one entity.

4.4. Implementation of Databases and Generation of Test Data

We implemented all the designs in different schemas of a PostgreSQL™ 8.3.6 database to prevent name conflicts of schema objects. Firstly, we created user-defined functions for generating test data to the tables of the regular design: *Customer* (30000), *Order_* (30000), *Order_line* (300008), and *Stock* (10000). In the brackets is the number of generated rows for a particular table. For all the tables, except *Stock*, we generated the same amount of test data as required in the TPC BENCHMARK™ C document [28]. For table *Stock*, we generated 10 times less data than was required in the document. TPC BENCHMARK™ C document also presents additional requirements to data that we took into account. Each warehouse must provide services to 10 districts and each district must have 3000 customers. If there is one warehouse, then there must be 1×10×3000 customers. Each customer must have one or more orders and each order must have between 5 and 15 (endpoints included) order lines. For each order, we randomly found the number of order lines by taking into account the constraint.

After we finalized the generation of test data for the regular design, we copied the same data to the tables that were created based on three other designs.

One can download the files with the statements that can be used to create the test database and generate test data from the following address:

http://staff.ttu.ee/~eessaar/files/Db_designs.zip

The computer, where we performed the experiments, had the following characteristics: Intel Core 2, T5600 1.83GHz, 2GB RAM, Windows XP Professional.

4.5. Relative Importance of Criteria

One can make some assumptions about the relative importance of the criteria by considering the contexts.

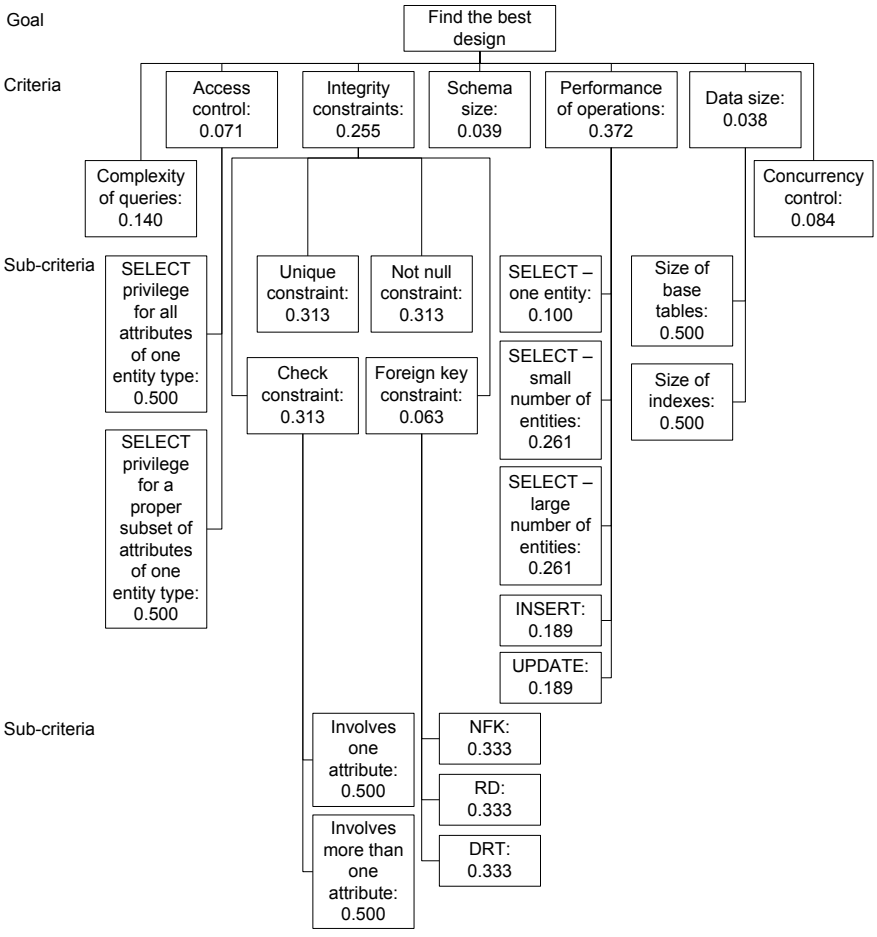


Fig. 7. Relative importance of the criteria in case of context 1

1. Access control is more important in case of *context 2* due to the requirements to confidentiality.
2. Statistical queries are more important in case of *context 1*. On the other hand, queries that help users to find information about a particular entity are more important in case of *context 2*.
3. Integrity constraints are more important in case of *context 1* due to the need to prevent registration of seemingly incorrect data.
4. Performance of data manipulation operations is more important in case of *context 2* due to the requirements to availability.
5. Concurrency control is more important in case of *context 2* due to the large number of concurrent users.

We calculated the relative importance (weights) of the criteria in case of *context 1* (see Fig. 7) and *context 2* (see Fig. 8) by comparing criteria pairwise in terms of the context.

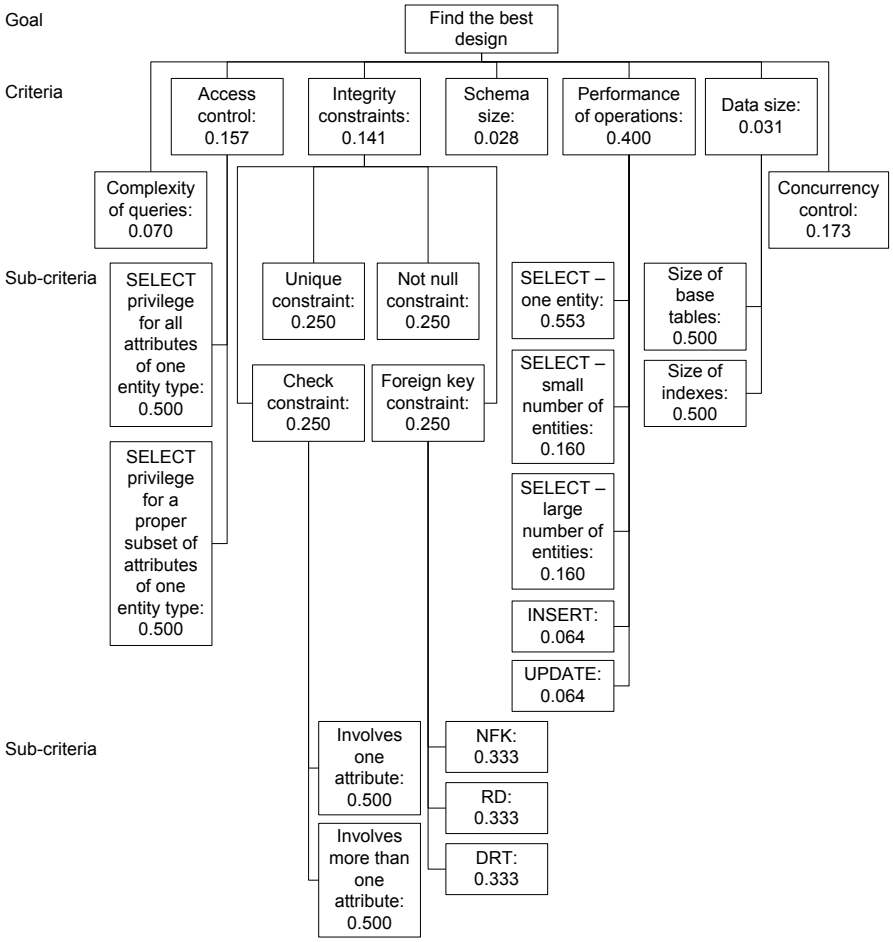


Fig. 8. Relative importance of the criteria in case of context 2

The pairwise comparison of criteria was performed by one expert. However, the proposed method does no rule out the use of more than one expert.

4.6. Evaluation of Alternatives in Terms of Criteria

In this section, we present the results of measurements that we performed to compare database designs. We present the results of measurements for all the criteria because they are needed to fully understand the final results of the study and they give detailed numerical information about the properties of the selected designs. We also explain the tasks, based on which we performed the measurements. In case of access control criterion, we present AHP comparison matrices that we created based on the measurement results. The proposed method requires the creation of such matrices in case of all the criteria but we present only some to illustrate their use.

We also performed consistency analysis of all the comparison matrices by calculating their consistency ratio. The ratio was always less than 0.10 that is positive evidence for informed judgment [21].

Access control. Table 1 presents the results of measurements in case of access control. We performed two tasks to evaluate the access control.

1. SELECT privilege for all attributes of one entity type. The task was to grant to role *role1* a SELECT privilege for reading data that corresponds to entity type *Stock*.
2. SELECT privilege for a proper subset of attributes of one entity type. The task was to grant to role *role1* a SELECT privilege for reading data that corresponds to attributes *C_LAST*, *C_ID*, *C_W_ID*, and *C_D_ID* of entity type *Stock*.

We counted the physical lines of code that were needed to implement the access control based on the requirements that were specified in the tasks.

Table 1. Measurement results in case of access control criterion (physical lines of code)

Sub-criterion	Regular design	Universal design	EAV/CR design	6NF design
SELECT privilege for all attributes of one entity type	1	9	40	16
SELECT privilege for a proper subset of attributes of one entity type	4	10	17	2

Table 2 and Table 3 present pairwise comparison matrices in case of the sub-criteria of access control. We used the results from Table 1 as the basis to perform the comparisons. For instance, the regular design is strongly better than the universal design in terms of sub-criterion "SELECT privilege for all

attributes of one entity type" according to Table 2. The higher is the score (column *Score*), the better is the design in terms of this criterion. The regular design is the best in case of sub-criterion "SELECT privilege for all attributes of one entity type" because it is possible to grant access by using one GRANT statement. In case of the 6NF design one has to use multiple grant statements and in case of the universal design and the EAV/CR design one firstly has to create views and then to grant SELECT privileges based on the views. The 6NF design is the best in case of sub-criterion "SELECT privilege for a proper subset of attributes of one entity type" because each attribute has a corresponding table. Therefore, one has to use as many GRANT statements as there are attributes in the subset. In case of other designs it is necessary to create views and to grant SELECT privileges based on the views.

Table 2. The results of pairwise comparison of designs in terms of granting SELECT privilege for all attributes of one entity type

Design	Regular design	Universal design	EAV/CR design	6NF design	Score
Regular design	1	5	8	6	0.636
Universal design	0.20	1	5	3	0.213
EAV/CR design	0.13	0.20	1	0.33	0.049
6NF design	0.17	0.33	3	1	0.103

Table 3. The results of pairwise comparison of designs in terms of granting SELECT privilege for a proper subset of attributes of one entity type

Design	Regular design	Universal design	EAV/CR design	6NF design	Score
Regular design	1	4	6	0.33	0.290
Universal design	0.25	1	3	0.20	0.107
EAV/CR design	0.17	0.33	1	0.14	0.051
6NF design	3	5	7	1	0.552

Complexity of queries. Table 4 presents the results of measurements in case of complexity of queries. We found the total complexity of three queries. The task has three subtasks.

1. SELECT – one entity. The task was to create query "Find all the data about an order, the identifier of which is 4, the associated warehouse of which has identifier 1, and the associated district of which has identifier 5".
2. SELECT – small number of entities. The task was to create query "Find the number of customers who have the last name BARCALLYESE, who use warehouse, the identifier of which is 1, and who are in district, the identifier of which is 7".
3. SELECT – large number of entities. The task was to create query "Find the number of stocks, the quantity of which in a warehouse is less than 15. The query must only consider stocks that have been ordered with orders, the

identifiers of which are between 2981 and 3001. In addition, the query must only consider stocks that 1) are associated with warehouse, the identifier of which is 1 and 2) that are associated with district, the identifier of which is 5".

Table 4. Measurement results in case of complexity of queries criterion (Coefficient of Network Complexity)

Task	Regular design	Universal design	EAV/CR design	6NF design
SELECT – one entity	0	21	21	3.2
SELECT – small number of entities	0	7.1	7.1	0
SELECT – large number of entities	0.5	11	11	1.3
Total complexity	0.5	39.1	39.1	4.5

In case of the regular design, we have to perform the smallest number of join and semijoin operations and therefore this design is the best in terms of this criterion.

Schema size. Table 5 presents the results of measurements in case of schema size. The schema size depends strongly on the context in case of the regular design and the 6NF design. On the other hand, the schema size of the universal design and the EAV/CR design changes only if database developers decide to change the set of data types, the corresponding values of which can be recorded in the database as attribute values. One has to note that this decision may depend on the context too.

Table 5. Measurement results in case of schema size criterion

Regular design	Universal design	EAV/CR design	6NF design
56	24	60	174

Integrity constraints. Table 6 presents the results of measurements in case of integrity constraints. Firstly, we calculated the number of foreign keys (NFK), referential degree (RD), and depth of referential tree (DRT). We performed two tasks to evaluate designs in terms of CHECK constraints.

1. CHECK (involves one attribute). The task was to implement constraint "Credit check of each customer must be GC or BC".
2. CHECK (involves more than one attribute). The task was to implement constraint "Each order must satisfy the condition S_REMOTE_CNT <= S_ORDER_CNT".

To evaluate designs in terms of UNIQUE constraint, the task was to implement constraint "In case of each Stock the value of attribute S_DATA must be unique". To evaluate designs in terms of NOT NULL constraint the task was to implement constrain "Each customer must have a last name". In case of CHECK, UNIQUE, and NOT NULL constraints, we counted the

physical lines of code that were needed to implement the constraints that were specified in the tasks.

For instance, in case of NOT NULL constraint the best design is the regular design because one can implement the integrity constraint by creating a declarative NOT NULL constraint. In case of other designs one has to use complex trigger procedures.

Table 6. Measurement results in case of integrity constraints criterion

Sub-criterion	Regular design	Universal design	EAV/CR design	6NF design
NFK	3	6	24	45
RD	0.08	0.30	0.39	0.26
DRT	3	3	3	3
CHECK (involves one attribute)	3	27	27	3
CHECK (involves more than one attribute)	3	71	65	42
UNIQUE	2	9	9	2
NOT NULL	2	102	101	73

Performance of data manipulation operations. Table 7 presents the results of measurements of performance of data manipulation operations (in milliseconds). We performed five tasks to evaluate the performance. The first three tasks were the same as in case of complexity of queries. In addition, we performed two tasks to measure the performance of database operations that modify data in a database: 1) insert a new order to the database (a new *Order_* entity) and 2) update the quantity of a stock, the identifier of which is 1, and that is in a warehouse, the identifier of which is 1.

Data size. Table 8 presents the results of measurements of data size. We found the size of base tables and indexes (in megabytes) in a test database. PostgreSQL™ creates automatically an index based on each primary key. We also created additional indexes on foreign keys if the corresponding columns where not indexed due to the primary key constraint. The regular design is the best in case of both sub-criterion.

Table 7. Measurement results in case of performance of data manipulation operations criterion (in milliseconds)

Sub-criterion	Regular design	Universal design	EAV/CR design	6NF design
SELECT – one entity	0.137	15046.219	14877.789	0.395
SELECT – small number of entities	79.474	3751.741	974.636	11.843
SELECT – large number of entities	213.064	4286.012	4086.631	339.235
INSERT	26.828	460.726	160.427	95.409
UPDATE	25.649	17339.214	8057.793	9.495

Table 8. Measurement results in case of data size criterion (in megabytes)

Sub-criterion	Regular design	Universal design	EAV/CR design	6NF design
Size of base tables	53	296	264	148
Size of indexes	18	288	288	87

Concurrency control. Table 9 presents the results of measurements in case of concurrency control. The task was to lock all the data that corresponds to one entity that has type *Stock* to modify the data and to prevent concurrent modification of the data. We counted the physical lines of code that were needed to implement the concurrency control task. The best design is the regular design because all the data about an entity is in one row and UPDATE statement automatically locks the entire row. In case of other designs, we had to lock data in more than one row by using explicit SELECT ... FOR UPDATE statements for that purpose. The 6NF design got the worst result due to the number of different tables that contain data about one entity.

Table 9. Measurement results in case of concurrency control criterion (physical lines of code)

Regular design	Universal design	EAV/CR design	6NF design
0	23	23	49

4.7. The Results of the Comparison

Table 10 and Table 11 summarize the results of evaluation of four designs.

Table 10. The results of evaluation of designs in case of context 1

Design/ Criterion	Regular design	Universal design	EAV/CR design	6NF design
Complexity of queries	0.0879	0.0087	0.0087	0.0348
Access control	0.0330	0.0114	0.0036	0.0234
Schema size	0.0086	0.0223	0.0061	0.0019
Integrity constraints	0.1411	0.0228	0.0215	0.0700
Performance of operations	0.1584	0.0173	0.0312	0.1654
Data size	0.0241	0.0026	0.0032	0.0080
Concurrency control	0.0490	0.0146	0.0146	0.0058
Relative goodness of a design	*0.5021*	*0.0998*	*0.0889*	*0.3092*

For instance, in case of *context 1* the relative importance of access control criterion is 0.071 (see Fig. 7). It has two sub-criteria, both of which have the relative importance 0.500 in terms of the main criterion. The score of the regular design is 0.636 and 0.290 in case of these sub-criteria (see Table 2

and Table 3, respectively). Therefore, the score of the regular design in case of access control in *context 1* is (0.636×0.500 + 0.290×0.500)×0.071= 0.0330.

Table 11. The results of evaluation of designs in case of context 2

Design/ Criterion	Regular design	Universal design	EAV/CR design	6NF design
Complexity of queries	0.0440	0.0044	0.0044	0.0174
Access control	0.0725	0.0251	0.0079	0.0513
Schema size	0.0061	0.0158	0.0043	0.0013
Integrity constraints	0.0751	0.0166	0.0136	0.0359
Performance of operations	0.1893	0.0177	0.0280	0.1652
Data size	0.0200	0.0022	0.0026	0.0067
Concurrency control	0.1008	0.0300	0.0300	0.0119
Relative goodness of a design	*0.5078*	*0.1117*	*0.0908*	*0.2897*

Last rows of Table 10 and Table 11 present the final scores of designs in case of different contexts (row *Relative goodness of a design*). We found them by summarizing the scores of alternatives in case of each design. The bigger is the final score, the better is the design in terms of the goal.

The best design, from the set of given designs, in case of both contexts is the regular design. It has the highest scores in case of almost all the criteria, except performance of data manipulation operations in case of *context 1* and schema size. In many cases, this design has much higher scores than other designs. The second best design in case of both contexts is the 6NF design. In case of *context 1* it is the best in terms of performance of data manipulation operations. The 6NF design has the worst results among the alternatives in case of schema size and concurrency control. The third and fourth best design in case of both contexts is the universal design and the EAV/CR design, respectively. The results are a surprise in case of *context 1* because some systems for managing data of clinical measurements use the universal design or the EAV/CR design [5]. We do not claim that the regular design is always the best and the EAV/CR design is always the worst because it depends on the context, criteria, relative importance of the criteria, measures, tasks, and alternatives with which the designs will be compared. For instance, Chen et al. [5] consider performance criterion in case of the EAV/CR design and mention also database and query maintainability as the advantages of using the EAV/CR design. In this paper, we used more criteria to evaluate database designs. We think that a wider range of criteria gives a better overview of advantages and disadvantages of a particular database design.

5. Conclusions

Often it is possible to use more than one database design to solve the same problem. Therefore, there should be a way to evaluate the suitability of designs.

In this work, we proposed a systematic and structured decision support method, which is based on the Analytic Hierarchy Process. It enables us to compare different database designs against each other while taking into account the requirements for the database. The comparison is based on the results of measurements.

We presented a case study of evaluating four designs of SQL databases to prove the usefulness of the proposed method. We compared the regular design, the universal design, the entity-attribute-value with classes and relationships (EAV/CR) design, and the sixth normal form (6NF) design in case of PostgreSQL™ DBMS 8.3.6 based on two quite different contexts. One of the contexts describes a scientific information system for managing the results of measurements. Another context describes a typical online transaction processing system. We found the relative goodness of each database design for both contexts. The goal of the case study was to demonstrate the use of the method and not to make absolute conclusions about the goodness of the designs. Although in case of both contexts the best was the regular design it is possible that the results could be different in case of different contexts, criteria, measures, tasks, solutions, or alternatives.

An important property of the proposed method is its flexibility – it can be used in case of different data models, criteria, and alternatives.

Additional results of our work are a set of possible criteria that one can use to evaluate the designs of SQL databases. We also found software measures that correspond to the criteria. It is possible to reuse all of that during the future evaluations of SQL database designs.

On the basis of the results it is concluded that the proposed method can be effectively used to evaluate database designs.

Future work must include more empirical studies about the use of the proposed method. It is necessary to use it in case of various data models and database designs. For example, one could create some designs that are like the first and most probable regular design, but with some common design mistakes. After that one could evaluate the method by comparing the regular design with the set of newly created designs.

In addition, it is crucial to further investigate criteria that can be used to evaluate database designs. There should also be a software system that supports and partially automates the use of the method.

References

1. Baroni, A.L., Calero, C., Abreu, F.B., Piattini, M.: Object-Relational Database Metrics Formalization. In Proceedings of the Sixth International Conference on Quality Software. IEEE Computer Society, Beijing, China, 30-37. (2006)
2. Behkamal, B., Kahani, M., Akbari, M.K.: Customizing ISO 9126 Quality Model for Evaluation of B2B Applications. Information and Software Technology, Vol. 51, No. 3, 599-609. (2009)
3. Blaha, M.: Patterns of Data Modeling. CRC Press, Boca Raton London New York. (2010)
4. Chaudhuri S., Narasayya, V.: Self-Tuning Database Systems: A Decade of Progress. In Proceedings of the 33rd International Conference on Very Large Data Bases. ACM, University of Vienna, Austria, 3-14. (2007)
5. Chen, R.S., Nadkarni, P., Marenco, L., Levin, F., Erdos, J., Miller, P.L.: Exploring Performance Issues for a Clinical Database Organized Using an Entity-Attribute-Value Representation. Journal of the American Medical Informatics Association, Vol. 7, No. 5, 475-487. (2000)
6. Date, C.J.: An Introduction to Database Systems (8th ed.). Pearson/Addison Wesley, Boston. (2003)
7. Date, C.J.: The Relational Database Dictionary. A comprehensive glossary of relational terms and concepts, with illustrative examples. O'Reilly, USA. (2006)
8. Date, C.J.: SQL and Relational Theory. How to Write Accurate SQL Code. O'Reilly, USA. (2009)
9. Date, C.J., Darwen, H., Lorentzos, N.A.: Temporal Data and the Relational Model. Morgan Kaufmann, USA. (2003)
10. Date, C.J., McGoveran, D.: The Principle of Orthogonal Design. Database Programming & Design, Vol. 7, No. 6. (1994)
11. Eessaar E., Soobik, M.: A Comparison of the Universal and the Regular Database Design. In: Haav, H.-M., Kalja, A. (eds.): Databases and Information Systems V - Selected Papers from the Eighth International Baltic Conference, DB&IS 2008. Frontiers in Artificial Intelligence and Applications, Vol. 187. IOS Press, Amsterdam Berlin Oxford Tokyo Washington, DC, 289-300. (2009)
12. Feuerlicht, G., Pokorný, J., Richta, K.: Object-Relational Database Design: Can Your Application Benefit from SQL:2003? In: Barry, C., Conboy, K., Lang, M., Wojtkowski, G., Wojtkowski, W. (eds.): Information Systems Development: Challenges in Practice, Theory, and Education, Vol. 2. Springer US, 975-987. (2009)
13. Forman, E.H., Selly, M.A.: Decision by Objectives. World Scientific Publishing Company. (2002)
14. Hay, D.C.: Data Model Patterns: Conventions of Thought. Dorset House Pub, New York. (1996)
15. Latva-Koivisto, A.M.: Finding a Complexity Measure for Business Process Models. Research Report. Helsinki University of Technology, Systems Analysis Laboratory, Finland. (2001)
16. March, S.T.: Techniques for Structuring Database Records. ACM Computing Surveys, Vol. 15, No. 1, 45-79. (1983)
17. Melton, J.: ISO/IEC 9075-2:2003 (E) Information technology – Database languages – SQL – Part 2: Foundation (SQL/Foundation), Aug. 2003. [Online]. Available: http://www.wiscorp.com/SQLStandards.html (current December 2004)
18. Park, K.S., Lim, C.H.: A Structured Methodology for Comparative Evaluation of User Interface Designs Using Usability Criteria and Measures. International Journal of Industrial Ergonomics, Vol. 23, Issues 5-6, 379-389. (1999)

19. Piattini, M., Calero, C., Sahraoui, H., Lounis, H.: Object-Relational Database Metrics. L'Object, vol. March 2001.
20. PostgreSQL. The world's most advanced open source database. [Online]. Available: http://www.postgresql.com (current September 2010)
21. Saaty, T.L.: How to Make a Decision: The Analytic Hierarchy Process. Interfaces, Vol. 24, No. 6, 19-43. (1994)
22. Simsion, G.: Data Modeling. Theory and Practice. Technics Publication, LLC, New Jersey. (2007)
23. Soutou, C.: Modeling Relationships in Object-Relational Databases. Data & Knowledge Engineering, Vol. 36, No. 1, 79-107. (2001)
24. Svahnberg, M., Wohlin, C., Lundberg, L., Mattsson, M.: A Quality-Driven Decision Support Method for Identifying Software Architecture Candidates. International Journal of Software Engineering and Knowledge Management, Vol. 13, Part 5, 547-573. (2003)
25. Teorey, T.J., Yang, D., Fry, J.P.: A Logical Design Methodology for Relational Databases Using the Extended Entity-Relationship Model. ACM Computing Survey, Vol. 18, No. 2, 197-222. (1986)
26. Theodoratos, D., Bouzeghoub, M.: A general framework for the view selection problem for data warehouse design and evolution. In Proceedings of the 3rd ACM international Workshop on Data Warehousing and OLAP. ACM, New York, NY, 1-8. (2000)
27. Tow, D.: SQL Tuning. O'Reilly, USA. (2003)
28. TPC BENCHMARK™ C – Standard Specification, Revision 5.9, June 2007. [Online]. Available: http://www.tpc.org/tpcc/spec/tpcc_current.doc (current June 2007)
29. Vaidya O.S., Kumar, S.: Analytic Hierarchy Process: An Overview of Applications. European Journal of Operational Research, Vol. 169, No. 1, 1-29. (2006)

Erki Eessaar received the BSc and MSc degrees in informatics from Tallinn University of Techology, Estonia in 1999 and 2001, respectively. He received PhD in Engineering (in informatics) from Tallinn University of Techology, Estonia in 2006. He is currently Associate Professor in Department of Informatics in Tallinn University of Technology. He teaches courses in database design and programming of databases. His research interests are in relational- and object-relational data models, metamodeling, metamodeling systems, metadata, patterns, repositories, database design, information system design, software measures, and model-driven development.

Marek Soobik received the BS and MS degrees in informatics from Tallinn University of Techology, Estonia in 2007 and 2009, respectively. He is currently working as a software developer at itestra GmbH. His research interests are in relational- and object-relational data models, database design, and information system design.

Received: September 29, 2010; Accepted: July 20, 2011.

DOI 10.2298/CSIS101110041I

An Enterprise Information System Agility Assessment Model

Rabah Imache[1], Said Izza[2], and Mohamed Ahmed-Nacer[3]

[1]University of Boumerdes,
Faculty of Sciences, Computer Science Department,
LIMOSE, Boumerdès, Algeria.
rimache@gmail.com
[2]CIAPACA,
25 Rue de la Petite Duranne, 13857, Aix en Provence, France
Izza.said@laposte.net
[3]University of Sciences and Technologie Houari Boumediene,
computer science department, LSI, Bab-Ezzouar, Algeirs, Algeria.
anacer@mail.cerist.dz

Abstract. The enterprise strategy is influenced by the environment changes: socio economic, legislative, technology, and the globalization. This makes its Information System more complex and competition increasingly fierce. In order for an enterprise to ensure its place in this hard context characterized by rapid and random changes of the internal and external environments, it must have fast adapting policy of its strategy and drive quickly important changes at all levels of its Information System in order to align it to its strategy and vice versa; that"s, it must always be agile. Therefore, agility of the Enterprise Information System can be considered as a primary objective of an enterprise. This paper deals with agility assessment in the context of POIRE project. It proposes a fuzzy logic based assessment approach in order to measure, regulate and preserve continuously the Information System agility. It also proposes a prototype implementation and an application of the proposed approach to a tour operator enterprise.

Keywords: Enterprise Information Systems (EIS), fuzzy logic, continuous improvement, urbanization, governance, reactivity, POIRE framework, agility dimensions, agility evaluation, regulation and preservation.

1. Introduction

The work of an enterprise as a system might be considered in terms of goals and objectives such as revenues, profits, market share, budgets, and all enterprises face similar challenges: growth, value, focus, change, future, knowledge, and time [38], and security problems and/or attacks. Information system is often considered as the heart of any organization; hence, the performance of the enterprise depends on the efficiency of its information

system. The enterprise strategy is influenced by the socio economic, legislative and technology changes. Moreover, the globalization of the economy makes the enterprise information systems more complex and competition increasingly fierce. So the enterprise, in order to ensure its survival and its sustainability, it must be agile permanently; that's, an enterprise must have fast adapting policy of its strategy and drive quickly important changes at all levels of all its dimensions in order to align them to its strategy and vice versa. This can be achieved by first getting an urbanization plan [28], [40] and continuously, the enterprise must be driven according to a governance framework [3], relying on an appropriate set of best practices and/or standards. Actually, the information system is becoming a tool of strategy for most of organizations.

To bring to its full potential, any enterprise requires various categories of applications at its various levels; such as: (1) computer-aided design (CAD) systems that are used for design of manufacturing products; (2) enterprise resource planning (ERP) systems that are used to manage the marketing and sales, inventory control, procurement, distribution, human resources,… ; (3) engineering document management (EDM) systems that are used to manage and leverage the enterprise digital design data investment; (4) materials requirements planning systems (MRP) that are used to manage manufacturing processes, (5) manufacturing execution system (MES) that are used to manage and monitor work-in-process on the factory floor; (6) computer-aided management and manufacturing (CAMM) systems that are used for mechanical, electrical or electronic engineering, and analysis and manufacturing; (7) product data management systems (PDM) that are used to manage the product's data; (8) product life cycle management (PLM) systems that are used to manage the entire life cycle of a product; (9) enterprise asset management (EAM) systems that are used to manage maintenance operations on capital equipment and other assets and properties [44].

A general characterization of enterprise applications in today's context is that they are HAD (Heterogeneous, Autonomous and Distributed) systems [2], [16]. Heterogeneous means that each enterprise application implements its own data and process model using different languages, interfaces, and platforms; which may result in different levels of heterogeneity: technical, syntactic and semantic. Autonomous refers to the fact that enterprise applications run independently of any other enterprise application. Distributed means that, enterprise applications locally implement their data model which they generally do not share with other enterprise applications.

Furthermore, today's industrial information systems have other specific characteristics:

- A strong automation of processes: this implies a strong dependence on the computing. This requires to extract all the knowledge embedded by the automated systems in order to make possible or to favor knowledge transfer, learning or simply maintenance;
- A strong evolution of systems: the systems are strongly changing because of the permanent evolution of the enterprise business that

implies that these systems must be relatively autonomous and loosely coupled to get more flexible information systems;

- An imperative cooperation: systems have to operate together to achieve business enterprise objectives. Each of these systems manages specific information using specific processes.

In such systems, we can conceive that the representation of the information and the processes is specific. On the other hand, it is imperative that these systems cooperate in a flexible way to allow exchanges, facilitate the reuse of their services and facilitate their modernization. Moreover, it is imperative that the enterprise will be agile in order to correctly face changes.

Nowadays, the above agility requirement constitutes a major preoccupation of organizations, which look for more flexibility and reactivity to respond to diverse changes. The main reasons of these constantly evolution and changes are generally the organization evolution, the evolution of regulations in force, business evolution, IT evolution, and cost containment [33]. As a consequence of all these evolutions, it becomes necessary to structure, develop and integrate the EIS in an agile way to facilitate its evolution and adaptation with respect to the enterprise strategy within the scope of the EIS governance on the basis of best practices and/or standards.

However, in practice, information systems have experienced anarchic growth and their complexity increases with time. Moreover, sustaining high performance in a flat world characterized with an irregular competitive landscape raises questions, such as**:** what is driving enterprise transformation, and what will happen if we do not respond? What will the future enterprise information system look like, how it will be different from the actual one? And how to get the future information system that ensures enterprise efficiency and sustainability?

The purpose of this work is to suggest an enterprise information system fuzzy logic agility assessment model and ensures its preservation and regulation within an acceptable range with time and with respect to environments random changes. It will offer an easy and simple framework for enterprises willing to handle changes permanently and efficiently in order to be competitive in the market. Hence, this paper is organized as follows: section 2 deals with the related work. Section 3 presents the details of our proposition. Section 4 describes the implementation of the software tool. Section 5 deals with the case study. And finally section 6 outlines some conclusions and perspectives.

2. Related Work

The concept of agility originated at the end of the eighties and the early nineties in the manufacturing area in the Unites States. Agile Manufacturing was first introduced with the publication of a report by Goldman [14] entitled "21st Century Manufacturing Enterprise Strategy". Since then, the concept was extended to supply chains and business networks [1], [42], and also to

enterprise information systems [31], [32] and also to software development [9].

Despite the age of the concept, there is no consensus yet on a definition of agility. According to Conboy and Fitzgerald [10], most of the agility concepts are adaptations of elements such as flexibility and leanness, which originated earlier. In developing their definitions, they draw on the concepts of flexibility and leanness to define agility as the continual readiness of an entity to rapidly or inherently, proactively or reactively, embrace change, through high quality, simplistic, economical components and relationships with its environment. According to Dessouza [12], being agile, generally, results in the ability to (1) sense signals in the environment; (2) process them adequately; (3) mobilize resources and processes to take advantage of future opportunities; and (4) continuously learn and improve the operations of the enterprise. In the same idea, Goranson [15] interpreted agility as creativity and defined the enterprise agility as the ability to understand the environment and react creatively to both external and internal changes. In the same way, Houghton and his collaborators [17] interpret agility of information systems as the ability to become vigilant. Agility can also be defined in terms of characteristics of the agile enterprise [41]: (1) sensing, (2) learning, (3) adaptability, (4) resilience, (5) quickness, (6) innovation, (7) flexibility, (8) concurrency, and (9) efficiency. Recently, [36] studied the advantage of positioning agility in order to help enterprises to better align their agile practices with stakeholder values.

As we can see, agile is a quality for both enterprises and information systems. The question which must be answered is: are agile enterprises and agile information systems distinct, or do they signify the same thing? The answer depends on two perspectives. On the one hand, the information management perspective, we can consider them as one and the same, because the concept of agile information system is used to denote an agile enterprise, or in a general manner an agile business. On the other hand, the technological perspective, we can consider them as different, because agile information systems is used to denote only instantiations of technological solutions that help the processing of information; in this case technology (i.e., the agile information system) constitutes only a component of the agile enterprise. In this paper, we mainly focus on the first perspective because it is more comprehensive and integrated. Moreover the information system is considered as the mirror of the enterprise.

From the point of view measuring agility, there are some works that treat the agility issues within enterprises and they mainly concern the strategizing of IS for agility [13], the identification of the capabilities of agility [39], the identification of the agility sources [30], and the proposition of conceptual agility framework [35], and the measurement of the agility [45].

Galliers [39] studied the agility in the strategy point of view by suggesting a framework for IS strategizing, and mentions that there are three main points for strategizing agility: (1) the exploitation strategy: concerns the environmental and organizational analysis, the enterprise information and knowledge systems, the standardized procedures and rules, and the information services; (2) the exploration strategy: it is related on the

alternative futures of information systems, the existing communities of practice, the flexibility of project teams, the existence of knowledge brokers, and the possibility of cross-project learning; and (3) the change management strategy: it depends on the ability to incorporate the ongoing learning and review. In fact this framework concerns mainly the global approach of identifying the strategizing elements of the IS and it does not deal with agility evaluation and preservation. However, this work is a part of the POIRE framework, we proposed.

Sambamurthy et al. [39] distinguish three interrelated capabilities of agility: (1) operational agility: is the ability to execute the identification and implementation of business opportunities quickly, accurately, and cost-efficiently; (2) customer agility: is the ability to learn from customers, identify new business opportunities and implement these opportunities together with customers; and (3) Partnership agility: is the ability to leverage business partner's knowledge, competencies, and assets in order to identify and implement new business opportunities. These capabilities of agility are included in our approach which identifies five capabilities of agility with respect to POIRE conceptualization. Moreover, [39] do not study the agility assessment and regulation which are a main part of our work.

Concerning the identification of agility sources, Martenson [30] argues that systems can be agile in three different ways: (1) by being versatile, (2) by reconfiguration, and (3) by reconstruction. Being versatile implies that an information system is flexible enough to cope with changing conditions as it is currently set up. If current solutions are not versatile enough, reconfiguration will be needed; this can be interpreted as pent-up agility being released by a new configuration. If reconfiguration is not enough, reconstruction will be needed; this means that changes or additions have to be made to the information system. Furthermore, [30] proposed a framework that discusses how agility is produced and consumed. This is closely related to the level of agility that can be interpreted as a result of an agility production process to which resources are allocated. These agility levels are then used in order to consume agility when seizing business opportunities. Additionally, he outlines that when consuming agility within a business development effort, in many situations agility is reduced. This means that we are confronted to negative feedback that indicates how much enterprise's agility is reduced by this business development effort. [30] identifies three main sources of agility in the following order: versatility, reconfiguration and reconstruction. In our case, we deal mainly with versatility and reconfiguration. Reconstruction is not recommended for the sake of continuity of service. Moreover, Martenson [30] mentioned the fact that when consuming agility, it is reduced; but he does not study the agility regulation and preservation. This point is considered in our approach in which a cyclic life cycle and a cyclic methodology for agility assessment, regulation and preservation are proposed.

An important agility framework, which concerns the management perspective, is that proposed by Oosterhout et al. [35]. In this framework, we begin with the analyses of the change factors, where a required response of the enterprise is related to the enterprise's IT capability. Then, an enterprise's

agility readiness is determined by its business agility capabilities. These latter are the reasons behind the existence or non existence of agility gaps. If there is a mismatch between the business agility needs and the business agility readiness, there is a business agility gap. This has implications for the business agility IT strategy. This framework concerns mainly agility analysis, but not agility production, regulation and preservation; however, this work is included in our framework without any contradiction.

Another important work is by Lui and Piccoli [29] who studied the agility in the socio-technical perspective and proposed a theoretical framework. In this latter, the information system is considered as composed of two sub-systems: a technical system and social system. The technical subsystem encompasses both technology and process. The social subsystem encompasses the people who are directly involved in the information systems and reporting structure in which these people are embedded. To measure information system agility using the socio-technical perspective, Lui and Piccoli [29] use the agility of the four components: technology agility, process agility, people agility, and structure agility. Hence, they argue that the agility is not a simple summing of the agility of the four components, but it depends on their nonlinear relationship and suggest the use of fuzzy logic measurement for IS agility. This framework is theoretical, but coherent; whereas our framework is practical and detailed.

Furthermore, Winsley and Stijin [46] mention the importance of preservation of agility through audits and people education. This latter aspect is important because most of organizations continually need education for continuous agility. Even though this work is theoretical, it is taken into account in our approach.

Finally, Tsourveloudis et al. [45] proposed a fuzzy logic knowledge-based framework to evaluate the manufacturing agility. The value of agility is given by an approximate reasoning method taking into account the knowledge that is included in fuzzy IF-THEN rules. By utilizing these measures, decision-makers have the opportunity to examine and compare different systems at different agility levels. For this purpose, the agility is evaluated accordingly to four aspects: (1) production infrastructure, (2) market infrastructure, (3) people infrastructure and (4) information infrastructure. [45] showed the importance of fuzzy logic in agility evaluation and consider four aspects of the IS, which are included in our POIRE framework. Moreover, our work is in line with this framework since it is based on the use of fuzzy logic evaluation.

Although all these works are important, they are either theoretical or address partially the EIS. Our work is in line with the existing approaches and is mostly close to those proposed by Lui and Piccoli [29] and Tsourveloudis et al. [45].We propose to extend these last researches to the evaluation of the agility of enterprise information system. Hence, we suggested a detailed and practical framework which, after identifying the need for agility of an urbanized EIS, will allow evaluating and regulating the global agility by maintaining the agility of each dimension within an acceptable range. Moreover, it includes the configuration management which gives the possibility of keeping track of the

EIS evolution, and the concept of good governance of EIS which are sources of agility production.

3. Proposition: the Poire Framework

This section describes the details of our proposition that are proposed in the context of the POIRE project. It extends the work of [23] in which the scope and principles of POIRE framework are presented.

3.1. Main principles

POIRE project concerns the information system agility in the context of large and complex enterprises with the aim to define agile best-practices. Within this project, agility assessment constitutes an important point that allows evaluating EIS agility maturity in order to determine the pertinent points that must be improved. Our agility approach is based on two main principles: urbanization and continuous improvement.

3.1.1. Urbanization

Information systems have several dimensions which can be analyzed with the typologies of the enterprise, and a complexity reflecting the human organization they must serve. Urbanization is necessary for two reasons: (1) maintain and manage at best a heritage until its effective obsolescence, and (2) have an agile information system able to evolve quickly and efficiently, according to the changing needs [28]. Hence, in our work, we consider urbanization as one of the agility production sources.

Due to their complexity, information systems are compared to urban systems or cities; hence the need of their urbanization. The main triggers of urbanization are: organizational changes, demand business, the IT market evolution, technological developments, the search for interoperability, and agility. Hence, urbanization makes an information system best suited to serve the enterprise strategy and anticipate changes in the business environment.

In order to reach these objectives, first, we have to define what should be the target information system, the one which will best serve the strategy of the enterprise, and satisfy the business process, in short an aligned information system; second, set the construction rules allowing the system to avoid repeating shortcomings of the former information system, and to anticipate changes, in short an agile information system; finally, determine the path to follow from the actual information system to get the new one, this needs knowing well the old information system in order to set appropriate criteria to know when to start and when finishing [6]. In fact, urbanization allows obtaining predictable information systems for which we can consider their evolution with serenity [37], allows adapting the enterprise strategies to

environments „changes [11], and identifies the boundaries between its sub systems, which is not trivial [43]. It must combine three problematic sets to reach rules and a path of urbanization: the conduct of processing, necessary security and the expected optimization.

According to [26], [28] and [40], the process of urbanization is based on three main phases: (1) identify the business strategy which determines the need, (2) definition of functional and specific requirements maps, and (3) technology orientation identification. In our case, we consider urbanization as a basis to reach alignment [7] and agility production at different levels of each dimension of the suggested POIRE conceptualization of the EIS (Fig.1). First urbanization is realized for each dimension of the EIS, this reveals the dimensions" interactions, and then the process of alignment will be executed accordingly with respect to the governance directives which are defined from the enterprise strategy. Hence the process of urbanization and alignment is first top down (analysis and design of strategy) then bottom up (execution and validation). This will increase the flexibility and alignment of the EIS, hence its agility. In this context, our proposition includes a life cycle with an urbanization phase which is, actually, being developed as further work.

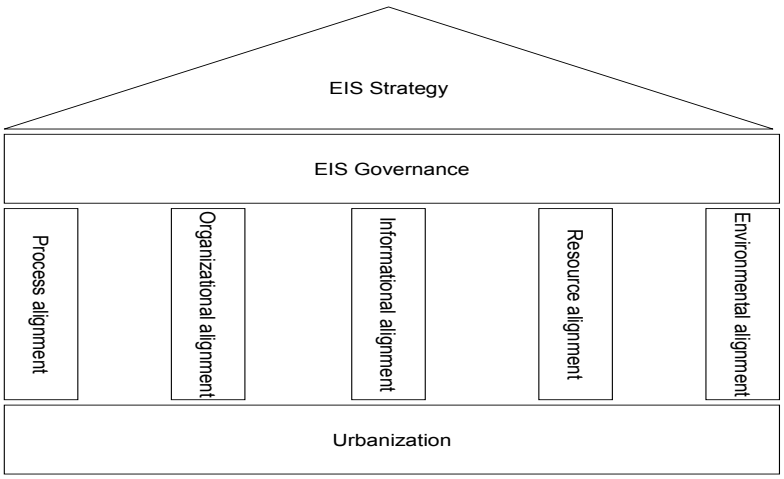


Fig.1. Urbanization and Alignment. Adapted from [7]

3.1.2. Agility and Continuous Improvement

Another aspect that"s related to enterprise and then EIS agility is the continuous improvement of the products and services" quality [8], [18]. Hence, continuous improvement is the basis for achieving EIS agility regulation and preservation. In fact, improvement must be permanent, pervasive and structured in order to continuously improve the effectiveness of the EIS in the scope of agility production and consumption. Moreover, the need for

improvement must be proactive. Indeed, one of the basic principles of agility as quality is prevention and continuous improvement to achieve the agility objectives that are defined and updated to reflect changes in business strategy. This means that continuous improvement is an endless project which aims to take into account failures and strategy orientation changes as early as possible with agile practices.

Agility encourages participation, because it is impossible to expect a total commitment of employees without creating an appropriate working environment, agility also means motivation and responsibility of each employee. Agility is the driver of competitiveness. The agility chain unites and connects all economic and social actors, so it is a matter of everyone and requires the participation of all.

To achieve the agility objectives and ensure continuous improvement of the EIS in order to meet the competition and conquer new markets, the enterprise must be driven according to a governance framework [3], relying on an appropriate set of best practices and/or standards combined with agile practices. Governance of the EIS ensures its management and it is considered as a management process based on best practices [25] allowing the enterprise to optimize its investments in order to achieve its agility objectives that are defined by the enterprise strategy. EIS governance allows [5]: (1) better decision-making: this can increase the efficiency of IS; (2) a clarification of the roles of different actors: it can create synergies; (3) better definition of responsibilities of the actors: this allows knowing the rights and duties of each employee; and (4) a better understanding of key processes related to the IS: this allows sharing the understanding of the complexity of processes and their implementation. Hence, good governance increases the degree of agility of the EIS. Standards and/or best practices allow: (1) the implementation of governance and improve controls of the IS; (2) evaluation, in the form of scale, of the level of achievement of one or more objectives; (3) manage the IS at all levels; (4) Audit of the IS; and (5) ensure the conformity of the IS.

Among the best practices and/or standards on EIS governance, we mainly find: COBIT (Control Objectives for Information and related Technology): dedicated to governance and audit of information systems [20]; ITIL (Information Technology Infrastructure Library) dedicated to optimize information technology services within the company [34]; ISO 27001 dedicated to audit and improve IS security [19]; CMMi (Capability Maturity Model Integration): dedicated to developing systems and software [4]; ISO 9001: dedicated to help organizations in developing general quality management systems [18].

Our contribution is based on an iterative life cycle for agility production and consumption, and a feedback loop regulation and preservation methodology in the scope of continuous improvement. Moreover, there is no contradiction in combining the POIRE approach with the best practices frameworks such as COBIT, ITIL, CMMI, and PRINCE.

3.2. Agility dimensions

In order to evaluate the overall agility of an EIS, first the IS is urbanized, in a top-down approach by successive refinements, into dimensions and components; then the process of agility evaluation is carried out in the bottom-up approach taking into account the mutual influences between the different components and dimensions of the EIS. In this context, POIRE framework suggested the following EIS conceptualization [23]:

Process dimension (P): This dimension deals with the enterprise behavior i.e. business processes. It can be measured in terms of time and cost needed to counter unexpected changes in the process of the enterprise. Agile process infrastructure enables in-time response to unexpected events such as correction and reconfiguration. It can be measured by their precision, exhaustively, non redundancy, utility, reliability, security, integrity, actuality, efficiency, effectiveness, and feasibility.

Organization dimension (O): This dimension deals with all the organizational elements involved in industry, i.e. structure, organization chart… It can be measured by their hierarchy type, management type, range of subordination, organizational specialization, intensity of their head quarter, redundancy, flexibility, turnover, and exploitability.

Information dimension (I): This dimension deals with all the stored and manipulated information within the enterprise. It concerns the internal and external movements of information. It can be measured from the level of information management tasks, i.e. the ability to collect, share and exploit structured data. It can be measured by their accuracy, exhaustively, non redundancy, utility, reliability, security, integrity, actuality, publication, and accessibility. Information represents a key factor for an enterprise to maintain competitiveness.

Resource dimension (R): This dimension is about the used resources within the enterprise. It can mainly concern people, IT resources, and organizational infrastructures. It can be measured by their usefulness, necessity, use, reliability, connectivity and flexibility. Concerning the people, which constitute in our opinion the main key in achieving agility within an enterprise, it can be assessed by the level of training of the personnel, the motivation/inspiration of employees and the data accessible to them. The adequateness and quality of perception, actions, decisions, and the time response represent a basis for achieving goals according to the enterprise strategy. So, people must be provided with adequate support while maintaining and strengthening their qualities through motivation and implication in the enterprise objectives.

Environment dimension (E): This dimension deals with the external factors of the enterprise, including customer service (B2C), regulations, global restrictions, and marketing feedback. It can be measured by the ability of the enterprise to identify and exploit opportunities, customize products, enhance services, deliver them on time and at lower cost and expand its market scope. It deals mainly with the interoperability; hence of the agility of the shared software and technology resources within a network of enterprises (B2B) and

with customers (B2C).Moreover, agility in the context of B2B and B2C will allow to improve permanently, the quality of the supply chain management (SCM). It can be measured by their reactivity, proactivity, and accuracy.

3.3. POIRE life cycle

For the sake of continuous improvement allowing agility production and consumption regulation and preservation, we suggest an iterative life cycle (Fig. 2) in the scope of the proposed POIRE framework.

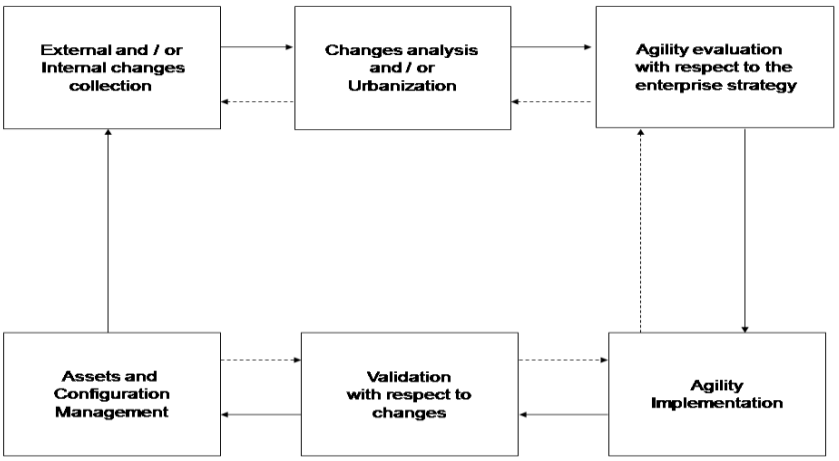


Fig.2. POIRE life cycle

First, the eve mission collects external and/or internal changes. Second, these changes are analyzed in the scope of the urbanization of the information system in order to have a better view and identify the dimensions‟ overlapping. Third the agility of the influenced components of the information system is evaluated with respect to the enterprise strategy. Forth, the necessary adjustments are implemented at each identified level of each POIRE dimension, to meet the required agility taking into account the mutual effects and/or interfaces between the EIS dimensions. Fifth, the designed changes are validated With respect to the collected changes. Finally, the assets and configuration management phase resumes the life cycle before taking into consideration any eventual external and/or internal new changes and the process of continuous improvement recycles. We notice that dot line arrows show the possible feedback in order to ensure coherence and validation.

Let us notice that the POIRE life cycle is iterative, and is based on two main principles: urbanization which allows structuring better, a priori, the

enterprise information system architecture; and continuous improvement of the EIS. This will allow obtaining an agile information system that supports the enterprise strategy and adapt to its changes (mergers, reconfigurations, new laws...) and keep its traceability by the configuration management.

3.4. POIRE metamodel

Fig.3 shows the POIRE metamodel [23]. It shows the result of the EIS decomposition, into dimensions, which are composed of factors and these factors are composed of criteria, in the context of the overall agility evaluation with respect to the appropriate metrics, in bottom-up approach.

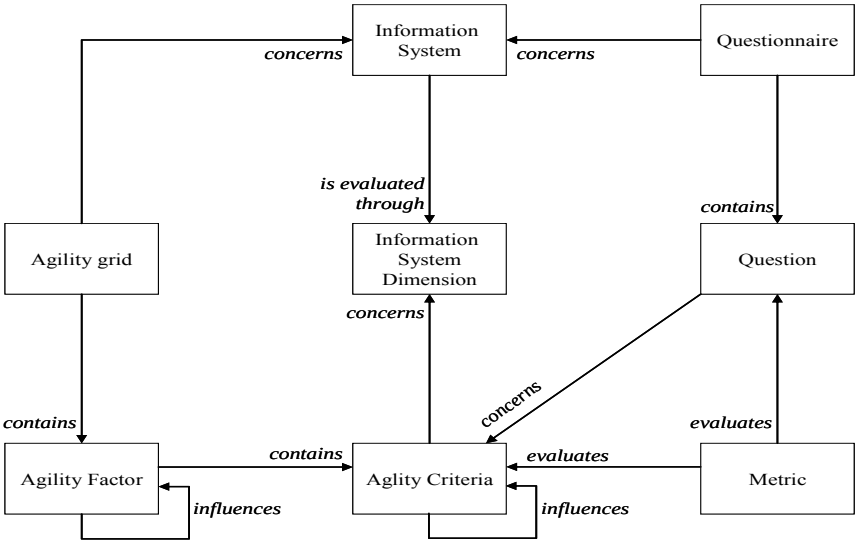


Fig.3. POIRE metamodel

An agility grid is established for each dimension of the EIS. The agility of the information system is then evaluated through the evaluation of the agility of its dimensions. This metamodel is used in order to generate the appropriate grids and questionnaires for a given EIS according to the culture of the enterprise and the context.

This is done according to a maturity grid that we proposed: this maturity grid is based on the perception of users and managers and aims to enable enterprises, especially the directions of information systems and IT managers to analyze and audit their information systems to assess their maturity and mainly the overall agility according to POIRE conceptualization, and it consists of five questionnaires, each for each POIRE dimension which are summarized in table 1 of section 3.5 below and a set of reference indicators. These indicators concern the scope of reference: listening to the influence

sources, governance: permanent alignment of the EIS to the enterprise strategy, enterprise culture: promoting the best practices, technology investment: provide sustainable development technologies, and projects and applications: optimize applications and technologies.

The development process followed for determining the criteria can be summarized as follows:

- Step 1: Construction of a fairly comprehensive list of criteria obtained after synthesis of the literature, that could possibly characterize the different concepts and / or aspects of an information system;
- Step 2: Determination of the external attributes of the evaluation: factors, which are obtained by combinations of criteria;
- Step 3: Purification of the previously obtained model by eliminating the factors or criteria considered unimportant, synonyms or polysemes;
- Step 4: Identification and normalization of metrics for each criterion. Identifying metrics for a criterion allows defining the quantitative measures for this criterion, while the normalization is to transform theses measures so that they belong to the interval [1- 5].

The obtained criteria are grouped into agility factors. The retained factors are: coherence, commodity, conformity, exploitability, flexibility, optimality, responsiveness, and security.

3.5. Agility evaluation approach

The concept of agility is somehow subjective; this makes difficult the definition of agility metrics. In addition the enterprise information system is multidimensional. So in order to evaluate the overall or global agility of an EIS, after the process of urbanization of the information system, we proceed as follows (FIG.4):

- Step 1: Collect internal and external information, this action depends on the adequateness of perception and quality of interpretation of the changes.
- Step 2: Define a set of quantitative agility factors, composed of criteria, which constitutes a questionnaire for each dimension defined in the POIRE framework, and set the criteria agility values.
- Step 3: Evaluate the agility of each dimension by combining the factors agility values.
- Step 4: Combine the obtained results in order to get the overall or global agility of the EIS.

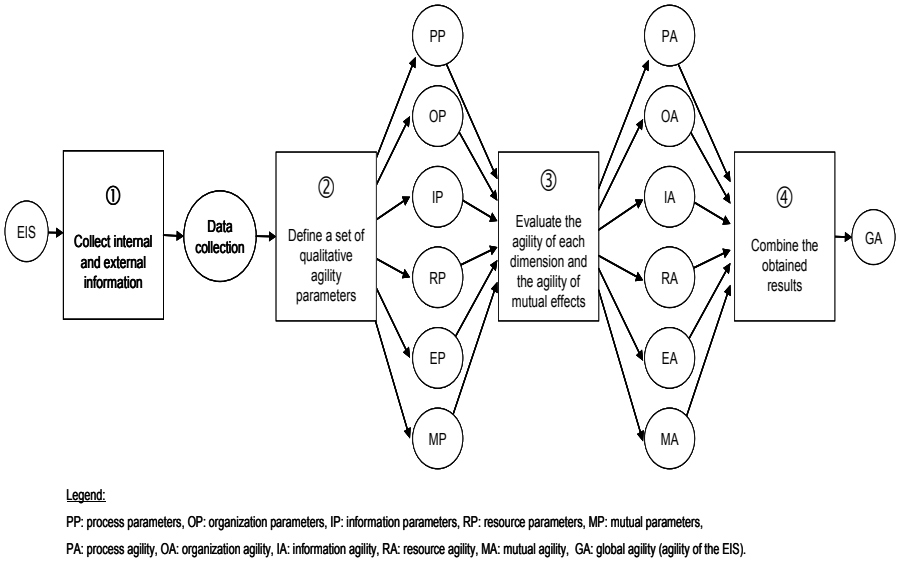


Fig.4. POIRE agility evaluation approach.

Our approach, in order to apprehend the complexity of EIS, defines three levels of complexity for agility evaluation:

- A simplified level which assumes that all the dimensions have the same weight and all the criteria of each dimension have the same weight too;
- An extended level which refines the results obtained by the simplified model, takes into account the weights of the dimensions in a given context and type of EIS;
- And the detailed level which takes into account the weights of dimensions, the weights of the factors, and the weights of the criteria. Hence, for each level of complexity we develop its corresponding assessment model and according to the complexity of a given EIS, we can use one, two or three of them sequentially.

In the present work, we deal with the high level of abstraction; that"s the simplified model corresponding to the simplified level of complexity and the extended level which takes into account the weights of the dimensions. This model assumes that all the criteria and factors of the dimensions have the same weight. It is applicable for EIS corresponding to this configuration, such as a tour operator. The calculations will be as follows:

The agility of any factor (FA_i) of any dimension is given by:

$$FA_i = \left(\sum_{j=1}^{NC} Cj \right) / NC \quad (1)$$

Where Cj : j^{th} criterion of the factor i
NC : number of criteria of the factor i
The agility of any of the five dimensions (DA) of the POIRE conceptualization is given by:

$$DA = \left(\sum_{j=1}^{NF} FA_j \right) / NF \quad (2)$$

Where FA_j : jth factor of the dimension D
NF : nombre de facteurs de la dimension DThe overall or global agility (GA) of the EIS is given by:

$$GA = \left(\sum_{j=1}^{5} \lambda_j DA_j \right) / \sum_{j=1}^{5} \lambda_j \quad (3)$$

Where GA: Global Agility of the EIS
DA: Dimension Agility, with D = [P/O/I/R/E]
λ_j: weight of dimension j which is set by managers and users.According to the preceding section, agility evaluation concerns all the dimensions of the EIS; hence, for each agility dimension, we suggest a list of pertinent criteria grouped in a list of factors, as shown in table 1 bellow for the resource dimension. Each dimension list is presented to users as questionnaire in order to set the appropriate value for each criterion. The value of agility of each criterion is expressed using the fuzzy logic variables which take values in the following set {Very low, Low, Average, High, Very High}, which in our opinion reflects the enterprise jargon and fuzziness is closer to the way of human thinking [24]; moreover, these values appear as comment for each calculated agility factor. For the sake of calculation, implied people associate numerical values to these variables in the following ranges: Very Low≤1, low≤2, Average≤3, High≤4, Very high≤5, and X if the criterion is not applicable in a given EIS, then it is not taken into account in the calculation process.

Table 1. Resource dimension grid

Factor	**Factor and its criteria designation**		**Criterion agility**	**Comment**
R1	What is the role of the personnel in the company?			
	$R1_1$	What is the level of training of the personnel?		

	$R1_2$	Is the personnel considered the most important resource?		
	$R1_3$	Do employees understand their roles?		
	$R1_4$	Are employees involved in management and decision making?		
	Factor agility			
R2	How human resources are managed?			
	$R2_1$	Does the personnel considered as an important resource?		
	$R2_2$	Is the direction encourages and inspires employees by example?		
	$R2_3$	Are the employees motivated?		
	$R2_4$	Is the personnel potentialities quantified and valued?		
	Factor agility			
R3	How human resources are organized?			
	$R3_1$	Does employee's assignment done according to their skills?		
	$R3_2$	Does the assignment of employees is done according to the context?		
	$R3_3$	Do you use cross-functional teams?		
	$R3_4$	Is the range of subordination respected?		
	$R3_5$	Do we use expert systems to improve performance and efficiency of our decisions and of our employees?		
	Factor agility			
R4	What are the characteristics of employees?			
	$R4_1$	Are employees agile and they adapt easily to changes?		
	$R4_2$	Do employees have the initiative and creativity spirit?		
	$R4_3$	Are employees involved in the enterprise management?		
	$R4_4$	Are employees versatile?		
	Factor agility			
R5	Are the hardware and software resources known?			
	$R5_1$	Are the hardware and software potentialities evaluated?		

	$R5_2$	Are the hardware and software resources used?		
	$R5_3$	Are the hardware and software resources known?		
	$R5_4$	Are the hardware and software resources are useful?		
	Factor agility			
R6	How hardware and software resources are managed?			
	$R6_1$	Are the hardware and software resources shared?		
	$R6_2$	Are the hardware and software resources updated?		
	$R6_3$	Are the hardware and software resources accessible?		
	$R6_4$	Are the hardware and software resources secured?		
	Factor agility			
R7	Are the hardware and software resources evolutionary?			
	$R7_1$	Are they interoperable?		
	$R7_2$	Are they integrated at all levels of the EIS?		
	$R7_3$	Can they be adapted to the context?		
	$R7_4$	Are they easily expandable?		
	Factor agility			
R8	What are the characteristics of hardware and software resources?			
	$R8_1$	Are they new?		
	$R8_2$	Is their use is easy?		
	$R8_3$	Are they maintainable easily?		
	$R8_4$	What is the degree of flexibility of the overall structure of hardware and software resources?		
	Factor agility			
R9	What is the lifespan of hardware and software resources?			
	$R9_1$	Are the hardware and software resources type of Sustainable Development (SD)?		
	$R9_2$	Are They robust?		

	$R9_3$	Are they flexible?		
	$R9_4$	Can they be guaranteed?		
	$R9_5$	Do we promote actively the technology allowing change?		
	Factor agility			
R10	Do employees have a privileged place?			
	$R10_1$	Is the intellectual property of employees has more value than physical products?		
	$R10_2$	Do employees have an environment allowing their evolution and development?		
	$R10_3$	Are the employees potentialities are exploited on the overall strategy basis?		
	$R10_4$	Do employees have access to all information necessary to perform their tasks?		
	$R10_5$	Is innovation considered as a main weapon of competitiveness?		
	Factor agility			
R11	How technology is exploited?			
	$R11_1$	Are new ways to use science and technology explored to produce new services and improve existing products?		
	$R11_2$	Is technology used as a means and not as an engine of change?		
	$R11_3$	Do we understand the positive and negative impacts of emerging technologies on our enterprise?		
	$R11_4$	Do we use more and more, technology of e-commerce to achieve our strategic objectives?		
	$R11_5$	Are products dependent on technology or are they based on the technology?		
	Factor agility			
	Agility of resources dimension			

Once the agility of each dimension is calculated, the users set the weights λ_j of the dimensions in order to calculate the overall agility of the EIS.

In order to help users and managers to fix the criteria value, for each applicable question it is associated a five levels scale ranging from 1 to 5. Let‟s consider the example of $R11_5$. The evaluation is based on the following levels‟ notes: (1) dependent on proprietary technologies, (2) dependent on standard technologies, (3) based on proprietary technologies, (4) based on standard and proprietary technologies, and (5) based on standard technologies.

3.6. **Agility preservation and regulation methodology**

An old information system which is not maintained within the logic of preservation of its agility is not a good candidate for important transformation. Hence; in our opinion, it is not sufficient to just make some parts of each dimension of the EIS agile for a given situation or context, but it is important to maintain them agile with time and for any situation; that is, the hardest task begins after the victory. Hence, there is a need of a continuous improvement of the EIS in order to update its agility: may be some previous parts will loose, increase, or decrease their level of agility, and/or some rigid parts will become agile, and so on. For, we suggest a continuous approach of evaluation, regulation and preservation of agility of the EIS within the allowed limits. These limits may be imposed by the technology limits (laws of physics), regulations, global restrictions, or human limits. This supposes that a priori the enterprise managers install an eve mission and define a flexible and revealing or efficient scoreboard in the scope of the EIS governance framework. This later is based on the enterprise strategy. Winsley and Stijin [46] mention the importance of preservation of agility through audits and people education.

Agility is non deterministic; hence, in order to evaluate the agility parameters, we propose to use the linguistic variables concept of fuzzy logic which has the advantage of being adjusted by the user. In order to evaluate the agility of an EIS, we begin with the analysis of the information system and the determination of the target information system grid. Then, we customize the questionnaire and we evaluate the different metrics that allow determining the agility criteria and also the real agility grid. Once the real agility is obtained according to the enterprise potentialities‟ exploitation and the target agility is obtained according to the existing enterprise potentialities evaluation, using the evaluation approach given in the previous section, they are compared with respect of the allowed error ε, and we conclude with an EAIS (Enough Agility of the Information System) message to the evaluator user, or we make the necessary recommendations and adjustments in the case where there is NEAIS (Not Enough Agility of the Information System) in such a way to converge the real agility value to the target agility value such that: (Real GA ≥ Target GA - ε).

We note that the allowed error ε is a positive real number and depends on the type of EIS. Fig. 5 illustrates the main principle of the proposed methodology, which guarantees a continuous improvement of the EIS agility.

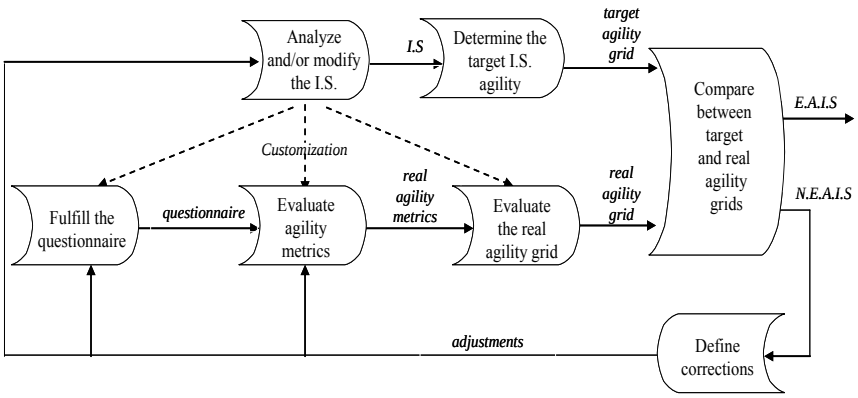


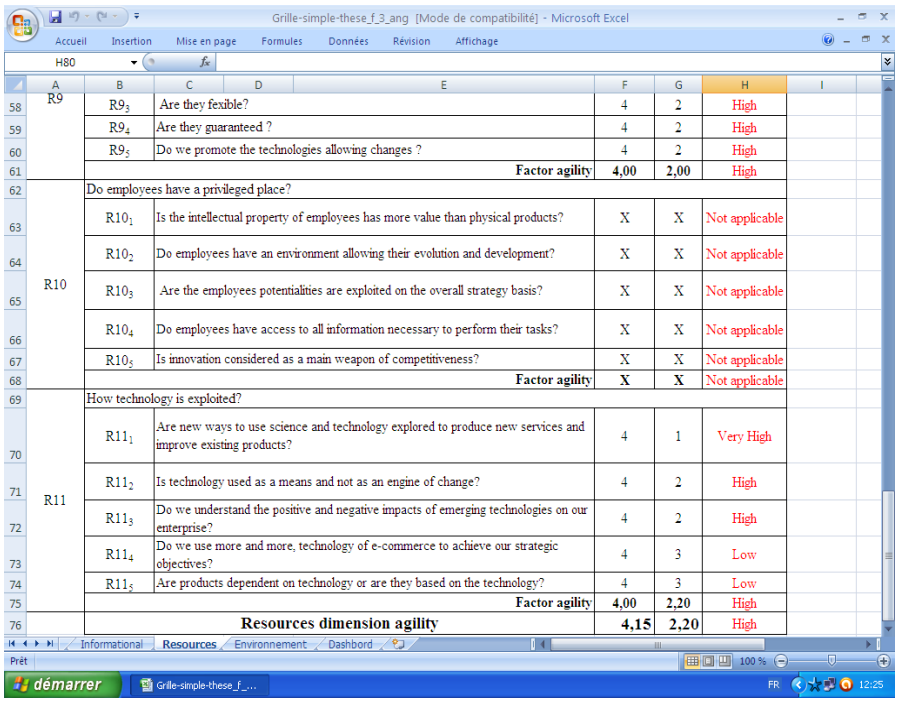
Fig.5. POIRE methodology for fagility evaluation, preservation and regulation.

4. Practical Implementation

4.1. Prototype implementation description

The prototype implementation is carried out using the MS Excel environment, and the tool interface looks like shown in Fig.6 bellow; it is characterized by its practicability and easy of use. The obtained results are given below. Equations (1) , (2) and (3), given in section 3-5 above, are used to calculate the approximate values of the agility of each dimension and the overall agility of the EIS in terms of the existing potentialities, then of their real exploitation. We note that the managers assume that all the dimensions have equal weight λ.

This prototype is used in order to evaluate the target agility on the basis of the EIS analysis in terms of existing potentialities evaluation, and the real agility by evaluating the real exploitation of the existing potentialities. The users and managers set the criteria target and real agility values within the range [1 – 5] if applicable, else X; then the target and real factor agility and the dimension target agility are estimated, automatically, using equations 1 and 2 respectively Once the target and the real agility values of each POIRE dimension are calculated, the target and real overall agility values of the EIS are generated automatically using equation 3; and the results are given in a dashboard form. This later allows, though its analysis to conclude and, eventually, generate the appropriate recommendations in order to improve the agility degree of the EIS.



58	R9	$R9_3$	Are they fexible?	4	2	High
59		$R9_4$	Are they guaranteed ?	4	2	High
60		$R9_5$	Do we promote the technologies allowing changes ?	4	2	High
61			**Factor agility**	**4,00**	**2,00**	High
62	R10	Do employees have a privileged place?				
63		$R10_1$	Is the intellectual property of employees has more value than physical products?	X	X	Not applicable
64		$R10_2$	Do employees have an environment allowing their evolution and development?	X	X	Not applicable
65		$R10_3$	Are the employees potentialities are exploited on the overall strategy basis?	X	X	Not applicable
66		$R10_4$	Do employees have access to all information necessary to perform their tasks?	X	X	Not applicable
67		$R10_5$	Is innovation considered as a main weapon of competitiveness?	X	X	Not applicable
68			**Factor agility**	**X**	**X**	Not applicable
69	R11	How technology is exploited?				
70		$R11_1$	Are new ways to use science and technology explored to produce new services and improve existing products?	4	1	Very High
71		$R11_2$	Is technology used as a means and not as an engine of change?	4	2	High
72		$R11_3$	Do we understand the positive and negative impacts of emerging technologies on our enterprise?	4	2	High
73		$R11_4$	Do we use more and more, technology of e-commerce to achieve our strategic objectives?	4	3	Low
74		$R11_5$	Are products dependent on technology or are they based on the technology?	4	3	Low
75			**Factor agility**	**4,00**	**2,20**	High
76			**Resources dimension agility**	**4,15**	**2,20**	High

Fig.6. Software prototype resource dimension interface

4.2. Tour operator EIS agility assessment

4.2.1. General description

The main mission of a tour operator enterprise is to organize trips throughout the world for its clients. There are three types of destinations: sea, mountains and cities, and for each destination, there are three types of accommodations: hotels, bungalows, or residences. All trips offer many activities, such as swimming, diving, tennis, golf, sailing, water skiing, climbing, cycling, and so on. The enterprise is organized around six departments: financial, marketing, commercial, exploitation, and organization, which depend from the general direction. Travel agencies which are spread through several countries depend from the commercial department. Moreover, the enterprise has collaborating agencies. The role of an agency is to help customers choose the appropriate voyage upon several parameters, such as the country, season, destination, accommodation, activities, prices, and so on. The operation of the enterprise depends on the efficiency of its networks and of its employees, its degree of competitiveness, and the general regulation.

4.2.2. Results and discussion

The application of the POIRE methodology and use of the software prototype in order to assess the agility of the Tour operator IS has yield the following results (table2) and (Fig.7) bellow.

Table 2: Results

Dimension	Target agility	Real agility
PA	4.20	2.45
OA	4.05	1.95
IA	4.05	2.15
RA	4.15	2.20
EA	4.05	2.30
GA	4.10	2.20

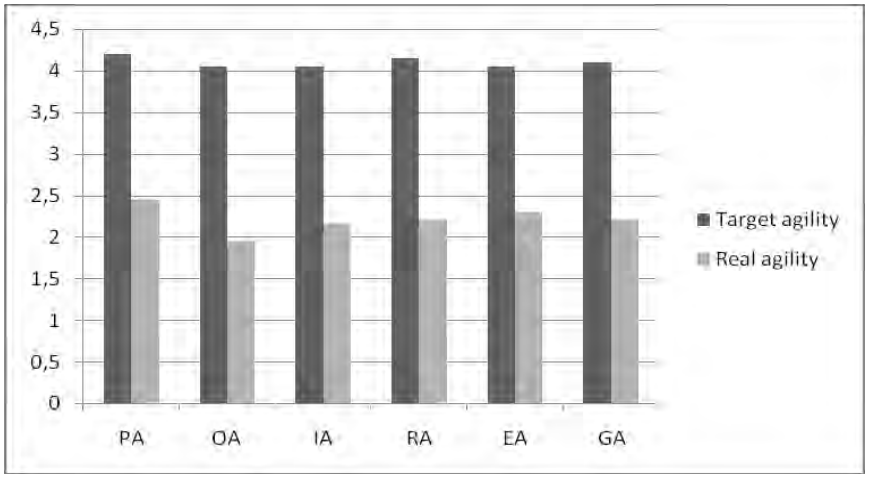


Fig.7. Graphical representation of the results

After being calculated, the real agility of the EIS, it is compared to the target agility in order to define, if any, the necessary actions which will bring the real value at least near the target one according to the allowed error ε that is set by the managers to 1.0 in this case study.

In our case, the estimated target global agility value is about 4.10, and the real value of the global agility is around 2.20; this means that managers and employees of the tour operator enterprise have to define the necessary adjustments, at the level of each dimension of the EIS in order to improve its real global agility such that (Real GA ≥ Target GA – ε); that"s (Real GA ≥ 3.1). This will allow the enterprise to be competitive, sustainable, and increase its benefit. In this scope, the main recommendations, for the managers and users in order to increase the EIS effectiveness and maturity by increasing its agility are as follows: increase customers" integration, increase collaborative work,

enhance publication and sharing of the vision and strategy of the enterprise, increase people agility, increase degree of precision and exhaustiveness of information, enhance training level of personnel, and exploit more the flexibility of resources.

5. Conclusion

This paper deals with agility in the context of POIRE project. It describes the need for enterprise information system agility and proposes an agility fuzzy logic evaluation, regulation and preservation framework based on two main principles: urbanization and continuous improvement that any enterprise should consider in order to manage changes and uncertainties in a competitive environment. Our assessment model is based on the evaluation of the agility of five dimensions that constitute any EIS and that are the result of the urbanization process. This allows organizations strategizing for agility production, consumption, and preservation. In addition this results in creating recoverable and secure architectures in different contexts and the enterprise will be continuously mirrored by its IS, then its sustainability is ensured as long as its agility is preserved. Moreover, our approach can be used to audit EIS agility and helps in defining recommendations for managers and users in order to increase the capabilities of the organization. Furthermore, there is, in our opinion, no contradiction in combining the POIRE approach with the best practices frameworks such as COBIT, ITIL, CMMI, PRINCE...and that each of the agility frameworks described in the related work covers certain parts of the EIS, whereas POIRE covers all levels of the EIS. The POIRE approach promotes individuals and interactions over processes and tools, and fosters collaboration with customers on contract negotiations. Moreover, it allows enterprises to be premonitory and assess their agility degree easily with the software prototype and generate the appropriate recommendations in order to initiate or handle any change rapidly and efficiently.

The application of the proposed model, to estimate the agility of a tour operator enterprise, shows the correlation and the coherence of the different models of the POIRE conceptualization and its practicability. For the managers of the enterprise, this experiment showed them the hidden faces of the different components of the EIS, and highlighted the way they would manage and govern better, in a collaborative manner, the enterprise to increase the benefit and evolve continuously smoothly with internal and external changes.

Finally, this presented model neglects the mutual interactions between the different dimensions factors and criteria of the EIS; so, the obtained values of the agility are not necessary the best ones. In order to improve the precision of calculation, actually, we are studying the interactions between the POIRE dimensions, factors and criteria which are represented by saturated graphs and defining mutual matrices which will define the heterogeneous and homogeneous links [27] between the dimensions and factors respectively,

according to the type and the context of evolution of the EIS; then mutual effects will be included in the mathematical model as product factors that may result in a negative, neutral or positive influence. Finally, we are planning their application to industrial enterprises information systems and e-government information systems. Moreover, further work is under study in parallel, such as studying the different ways of producing agility at different levels of the EIS, and one of the obtained results deals with the agility production by the integration of SOA with ITSM [21], and use the developed prototype to specify and develop the industrial software tool that will be used to generate automatically the recommendations and statistical analysis.

References

1. Adrian E., Coronado M., and Lyons A. C.(2007) “Investigating the Role of Information Systems in contributing to the Agility of Modern supply Chains”, In Desouza K. C. editor, Agile Information Systems: Conceptualization, Construction, and Management, Elsevier, Burlington, USA, ISBN 10: 0-7506-8235-3, pp. 150-162.
2. Bussler C.(2003) "Semantic Web Services: Reflections on web service mediation and composition", Proceedings of the Forth International Conference on Web Information Systems Engineering (WISE 2003) December 10-12, IEEE computer science editor, Roma, Italy, pp 253.
3. Chamfrault T. And Durand C.(2006) "ITIL and services management: methods, implementation and best practices", Dunod editions, Paris, France.
4. Chrissis Mary Beth, Mike Konrad, and Sandy Shrum,(2006) „CMMI(R): Guidelines for process Integration and Product Improvement", Second Edition, SEI Series in Software Engineering.
5. Cigref (2004) „Information system governance", Cigref. (Available at: http://www.cigref.fr)
6. Cigref (2003) " Increase information system agility", Cigref.(Available at www.cigref.fr.)
7. Cigref (2002) „Information system governance: issues and approach", Cigref .(Available at www.cigref.fr.)
8. Claude Y Bernard (2000), „Management by the total quality", AFNOR, Paris, France.
9. Cohen, D., Lindvall, M., and Costa, P.(2004) "An introduction to agile methods", In Advances in Computers, New York: Elsevier Science, pp. 1-66.
10. Conboy K. and Fitzgerald B. (2004) „Towards a Conceptual Framework of Agile Methods: A Study of Agility in Different Disciplines", In proceedings of the 2004 ACM Workshop on Interdisciplinary Software Engineering Research (WISER), Newport Beach, CA, USA, November. ACM Press NY. Pp 37-44.
11. Denny, Williams G. J. and Christen P. (2009) „Visualizing temporal cluster changes using relative density self-organizing maps" Knowledge and Information Systems Journal, DOI 10.1007/s10115-009-0264-5. www.springerlink.com
12. Desouza, K. C (2007) “Preface”, In Desouza K. C. editor, Agile Information Systems: Conceptualization, Construction, and Management, Elsevier, Burlington, USA, ISBN 10: 0-7506-8235-3.
13. Galliers R. D. (2007) „strategizing for agility: Confronting Information Systems Inflexibility in Dynamic Environments". In Desouza K. C. Editor (2007), Agile

Information Systems: Conceptualization, Construction, and Management, Elsevier, Burlington, USA, ISBN 10: 0-7506-8235-3, pp. 1-14.
14. Goldman S. L. and Preiss K; (1991), "21st Century Manufacturing Enterprise Strategy", Bethlehem, PA: Iacocca Institute, Lehigh University.
15. Goranson H. T.(1999) „"The Agile Virtual Enterprise: Cases, Metrics, Tools"". Quorum Books editions, 1999. ISBN 1-567-20264-0.
16. Hasselbring W. (2000) „"Information system integration"", Communications of the ACM, Vol. 43, N° 6, pp. 33-38..
17. Houghton R. J. et al., (2007) "Vigilant Information Systems: The Western Digital Experience", In Desouza K. C. editor, "Agile Information Systems: Conceptualization, Construction, and Management", Elsevier, Burlington, USA, ISBN 10: 0-7506-8235-3, pp. 222-238.
18. ISO (2005), „"ISO 9000, Quality management Systems, Fundamentals and Vocabulary"".
19. ISO/IEC (2005), „"ISO 27001, Information Technology Security Techniques, Information Security Management Systems, Requirements"".
20. IT Governance Institute (2007), „"COBIT 4.1 Framework Control Objectives Management Guidelines Maturity Model"",.
21. Izza S. and Imache R. (2010) „"An approach to achieve IT agility by combining SOA with ITSM"", Int. J. Information Technology and Management, Vol. 9, N°. 4, pp. 423-445. ISSN: 1461-4111. Underscience publishers. www.underscience.com
22. Izza S.(2000) ""Information systems modelling: proposition of a formalism for analysis and evaluation"", Magister thesis, University of Tizi-Ouzou, Algeria, April.
23. Izza S., Imache R., Vincent L., and Lounis Y.(2008) „"An approach for the evaluation of the agility in the context of enterprise interoperability"". In interoperability for enterprise software and applications conference (I-ESA'08). In Mertins K., Ruggaber R., Popplewell K., and Xiafei X. editors, Enterprise interoperability III: new challenges and industrial approaches. Springer Verlag, London, ISBN 978-1-84800-220-3. Pp. 3-14.
24. Kianmehr K., Alshalalfa M. And Alhajj R. (2009) „"Fuzzy clustering-based discretization for gene expression classification"" Knowledge and Information Systems Journal, DOI 10.1007/s10115-009-0214-2. www.springerlink.com
25. Leignel, J. L.(2006) „"Information systems governance"", CIO Strategy, Nice, France, june 12.
26. Leroux B. (2004) „"Urbanization and modernization of information system"", Lavoisier, Paris, france.
27. Long B., Zhang Z. And Yu P. S. (2009) „" A general framework for relation graph clustering"" Knowledge and Information Systems Journal, DOI 10.1007/s10115-009-0255-6. www.springerlink.com
28. Longépé C.(2002) „"The project of information system urbanization"", Dunod, editions, Paris, France.
29. Lui T-W., and Piccoli G.(2007) "Degrees of agility: Imlications from Information systems Desigh and Firm Strategy", In Desouza K. C. editor, Agile Information Systems: Conceptualization, Construction, and Management, Elsevier, Burlington, USA, ISBN 10: 0-7506-8235-3 pp. 122-133.
30. Martenson A. (2007) „"Producing and Consuming Agility"". In Desouza K. C. Editor, Agile Information Systems: Conceptualization, Construction, and Management, Elsevier, Burlington, USA, ISBN 10: 0-7506-8235-3, pp. 41-51.
31. Mitra A. (2006) "Agile systems with reusable patterns of business knowledge: a component-based approach", Lavoisier, Paris, France

32. Mooney J. G. and Ganley D. (2007) "Enabling Strategic Agility Through Agile Information Systems", In Desouza K. C. editor, Agile Information Systems: Conceptualization, Construction, and Management, Elsevier, Burlington, USA, ISBN 10: 0-7506-8235-3, pp. 97-109.
33. Octo Technology (2005) „"Service-Oriented Architecture (SOA): a policy of interoperability"", White Book, Octo Technology.
34. OGC (Office of Government Commerce) (2007) „"ITIL Lifecycle Publication Suite, Version 3: Continual Service Improvement, Service Operation, Service Strategy, Service Transition, and Service Design"..
35. Oosterhout M. V., Waarts E. V, Heck E. V., and Hillegersberg J. V. (2007) „"Business Agility : need, readiness and alignment with its strategies"". In Desouza K. C. Editor, Agile Information Systems: Conceptualization, Construction, and Management, Elsevier, Burlington, USA, ISBN 10: 0-7506-8235-3, pp. 52-69.
36. Oza N., Abrahamsson P. and Conboy K. (2009) „"Positioning Agility"" In Pekka Abrahamsson, Michele Marchesi and Frank Maurer editors, Agile processes in software engineering and extreme programming. Vol. 31, pp. 206-208. DOI 10.1007/978-3-642-01853-4_33. www.springerlink.com
37. Pascot D., Bouslama F. and Mellouli S. (2010) „"Architecturing large integrated complex information systems: an application to healthcare"". Knowledge and Information Systems Journal, DOI 10.1007/s10115-010-0292-1. www.springerlink.com
38. Rouse W. B. (2007) “Agile Information Systems for Agile Decision Making”, In Desouza K. C. editor, Agile Information Systems: Conceptualization, Construction, and Management, Elsevier, Burlington, USA, ISBN 10: 0-7506-8235-3, pp. 16-30.
39. Sambamurthy V., Bharadwaj A., and Grover V. (2003) „"Shaping agility through digital options: reconceptualizing the role of information technology in contemporary firms"". MIS Quarterly, 27 (2): 237-262
40. Sassoon J. (1998), „"Information systems urbanization"", Hermes, Paris, France.
41. Stamos E. and Galanou E.(2006) „"How to evaluate the agility of your organization: Practical guidelines for SMEs"" VERITAS.
42. Swafford P. M.(2003) "Theoretical development and empirical investigation of supply chain agility", Dissertation, Georgia Institute of Technology, Atlanta, USA, April.
43. Takacs B. and Demiris Y. (2009) „"Spectral clustering in multi-agent systems"" Knowledge and Information Systems Journal, DOI 10.1007/s10115-009-0272-5. www.springerlink.com
44. Toublant P. J., Crepet S., Rafii N., and Graton Y. (2002) „"Microcomputer industrial data processing"", Techniques de l"ingénieur, S8190, pp. 1-13.
45. Tsourveloudis N , Valavanis K, Garacanin D, and Matijasevic M. (2002) "On the Measurement of Agility in Manufacturing Systems", Journal of Intelligent and Robotic Systems, Kluwer Academic Publishers Hingham, MA, USA, 33 (3), pp. 329 – 342.
46. Wensley A and Stijin E. V. (2007) „"Enterprise information systems and the preservation of agility"". In Desouza K. C. Editor, Agile Information Systems: Conceptualization, Construction, and Management, Elsevier, Burlington, USA, ISBN 10: 0-7506-8235-3, pp. 178-187.

Rabah IMACHE is lecturer at University of Boumerdès from several years. He teaches Software Engineering and Advanced Information Systems. His main research interests are globally in the field of information systems engineering, integration, agility and governance.

Dr. Saïd IZZA is currently and simultaneously an Associate Professor at EMSE and a Consulting Director at CIA-PACA. His main interests are in the field of enterprise information systems agility and optimization. Currently, he teaches Advanced Information Systems and he leads and develops the service offering on Agility at CIA-PACA in France.

Mohamed AHMED-NACER is the director of computer engineering laboratory at USTHB and is in charge of the software engineering team. His current research interests include process modelling, information systems, software architecture based components and service web development.

Received: November 10, 2010; Accepted: July 1, 2011.

DOI 10.2298/CSIS110315046V

Selecting a Methodology for Business Information Systems Development: Decision Model and Tool Support

Damjan Vavpotič[1] and Olegas Vasilecas[2]

[1]University of Ljubljana, Faculty of Computer and Information Science,
Tržaška 25, 1000 Ljubljana, Slovenia
damjan.vavpotic@fri.uni-lj.si
[2]Vilnius Gediminas Technical University, Information Systems Research Laboratory
Sauletekio al. 11, LT-10223 Vilnius-40, Lithuania
olegas.vasilecas@vgtu.lt

Abstract. The paper presents a decision model and a tool that helps to find an information systems development methodology (ISDM) for a computer-based business information system (IS) that is suitable to a certain IS development project or an organisation dealing with IS development. The intention of the model is not only to suggest a certain ISDM, but also to propose the properties an ISDM should have to suite the project or the organisation. It is designed in a way that facilitates experimentation with different project, organisation and ISDM properties. Based on the model we created a tool that has been applied on several cases in which we validated the correctness of its recommendations and established that it can have a significant positive contribution in the process of ISDM selection and in the process of improvement of existing ISDM.

Keywords: business information systems development, development methodology, decision model.

1. Introduction

An information system can be defined as a set of interrelated components that collect, manipulate, store and disseminate data and information and provide a feedback mechanism to meet an objective [34]. These components include users who perform information related tasks, and different information technologies (IT) that help business users perform these tasks or in some cases perform the tasks autonomously without user intervention. In the context of the paper we focus on IS that are computer-based and support business operations. In the continuation of the paper the abbreviation IS is used for such information systems.

Nowadays, computer systems are the most important IT component and are used to store data and to convert data to useful information. Development of computer support is therefore an important part of IS development. It is a relatively complex process that comprises tasks like gathering requirements for new IS, analysing the requirements, designing the new IS, implementing the new IS, etc. To optimize the development process of IS various IS development methodologies (ISDM) have been developed in the past decades that prescribe various approaches that can be applied during the development to improve the IS development process and the end product i.e. IS to be developed.

An ISDM can be defined as a collection of procedures, techniques, tools, and documentation aids which will help the system developers in their efforts to implement a new information system [3]. In the paper we consider ISDMs that focus on development of computer-based IS. These ISDMs can vary from heavyweight that precisely define every single step of the development, to lightweight ISDMs that only vaguely define the most important parts of the development process. The lightweight ISDMs gained their importance with the emergence of agile software development that stresses the using of the most suitable ISDM for a certain type of IS development project and organisation dealing with IS development [8]. The problem is, however, how to select an ISDM type that suits the requirements of an IS development organisation and its type of development projects. Organisations dealing with IS development often lack knowledge and experience to be able to objectively evaluate different types of ISDMs. Often the selection of an ISDM is based only on an advice of a consultant company trying to sell its own one. Consequentially, such approach often results in selection of an only partially suitable ISDM and can be considered as one of the important reasons for low ISDM adoption levels in IS development organisations. Fitzgerald [10], for instance, found out that 60 per cent of companies do not use ISDM at all and that only six per cent reported following ISDMs rigorously.

In this paper we present a decision model and tool support that IS development organisations can use to consider wider range of ISDM, and thus select a suitable ISDM that meets their requirements and expectations. The purpose of the model is not to appoint the use of a certain ISDM, but merely to suggest what kind of ISDM is suitable for a certain type of IS development project and/or IS development organisation.

The paper is organised as follows. Section 2 presents the background and the research method. It is followed by the description of the decision model in section 3 and description of tool support in section 4. Section 5 in which we present verification of the model in practice is followed by the conclusion.

2. Related work and research method

The problem of selecting a suitable ISDM has been addressed in different ways in the past. However, the proposed approaches and models typically consider only a relatively small number of different ISDM aspects, limit themselves only on selection of ISDM form an ISDM family, or offer general rules and guidelines but do not advise about specific ISDMs.

Cockburn [8], for instance, provides a decision model that helps select the suitable ISDM from a family of ISDMs named Crystal. To select an ISDM that suits the project, the model considers three main project properties: number of people involved in the project, criticality of the project, and priorities of the project. However, the model is limited to selection of an ISDM from the Crystal family of ISDM, although similar properties can be used also for selection of other ISDM. Another example is Rational Unified Process (RUP) [14] that provides guidelines for tailoring the ISDM (RUP in specific) to the needs of a development project. It considers various project parameters and provides rules for selecting the most suitable development lifecycle. However, it is limited only to customization of RUP and does not consider other ISDMs. Yet another example is offered by Mikulenas and Butleris [26], who present the idea of constructing a specialised evaluation model of suitability assessment that considers agile ISDMs only. As an alternative to these relatively specialised models and guidelines, more general guidelines for ISDM selection can be found. For instance, McConnell [25] provides general guidelines on how to select the most suitable development lifecycle and gives practical tips on best practices for various development environments. Similarly, different authors of agile ISDM (e.g. [11, 12, 30]) provide general recommendations for adaptation and use of these ISDM. Although these general guidelines and tips are very useful they do not include recommendations for selection of a specific ISDM. An example of a comprehensive method for evaluating software engineering methods and tools is DESMET [21]. It provides nine methods of evaluation and a set of criteria to help evaluators select an appropriate method. However it is intended to be used by an evaluator to plan and execute an evaluation but does not provide advice about specific ISDMs and their suitability to a certain organisation or project. Other approaches exist that help improve the suitability of an ISDM to a certain IS development project and IS development organisation like method engineering [4, 6]. The result of using such approaches is an ISDM that is tailored to the needs of such project and organisation from existing ISDM components. They consider a wide array of project and organisation properties that form the basis for selection of the most suitable ISDM components. These approaches, however, are not intended to verify whether certain ISDM is more or less suitable for a certain IS development project or organisation dealing with IS development; they rather focus on construction of a tailored ISDM. Furthermore, IT governance frameworks and best practices like Control Objectives for Information and Related Technology (COBIT) [16] and Information technology infrastructure library (ITIL) [28] exist that help improve IT related processes and provide

valuable knowledge regarding managing and improving these processes. Similarly, Information Services Procurement Library (ISPL) [17] that is based on former Euromethod [9] provides best practices in the field of acquisition processes related to Information Technology. Although all these frameworks provide valuable information and can be used to access and improve the general quality of different IT related processes they are not intended to provide advice about selection of a specific ISDM that is suitable for a specific project or organisation.

We first presented idea of the decision model that would help select a suitable ISDM for and IS development project at ISD conference in 2003 [38] where we proposed four basic requirements for such decision model. Firstly, the model should be extendable so that users could add their own properties and decision rules and adapt the model to the needs of their organisation. Users should also be able to include additional ISDM and project types. Secondly, the results of the model should be transparent i.e. users should be informed about the reasons why a certain ISDM is preferred over the other in given context and what are the ISDMs desired properties. Thirdly, the model should facilitate experimentation with various properties and their weights, thus enabling the user to receive the most valuable results for certain situation. Finally, the model should facilitate work with incomplete information, which enables the user to receive at least partial results even though he does not have enough information to enter all properties. Since this initial idea, the model has been considerably extended, improved and used on various practical cases.

The research was divided into two phases. It the first phase we used literature review to assemble various decision rules and models regarding ISDM selection. Then we integrated these rules into a comprehensive decision model and developed a tool that implemented the model. We tested the decision model by applying the tool on three real life projects. Based on the findings from these tests results we improved the decision model and used it in further research and practical applications. The aim of the second phase of the research was to validate the recommendations produced by the model and to confirm our proposed hypothesis that the model can significantly contribute to ISDM selection and improvement process in an IS development organisation. This part of the research was performed as a case study [41].

3. The decision model

The selection of an ISDM that suits a certain IS development project and/or a certain organisation dealing with IS development requires careful consideration of a variety of the project and the organisation properties and a variety of ISDM properties. Decision situations in which a large number of properties have to be considered can be well managed by multi-attribute decision models [37, 42] that also form the basis for our decision model. An

approach that is often used in similar situations is analytic hierarchy process (AHP) by Saaty [31]. Although we build sets of project/organisation properties and ISDM properties in a similar way as criterion hierarchies are built in AHP using only AHP approach is not sufficient as we have to determine which ISDM properties suite certain project/organisation properties. To determine the suitability of ISDM properties to project/organisation properties we have to use various decision rules. Therefore we propose a decision model that is a hybrid between a weighted score model and a rule-based model. The decision model uses two separate sets of properties: a set of properties that describe ISDMs (discussed in subsection 3.2) and a set of properties that describe projects and/or organisations (discussed in subsection 3.3). Both sets are organised as weighted score models where the weights of ISDM properties are pre-set and are not intended to be changed while weights of the properties that describe projects and/or organisations can be changed by users of the model who in this manner can emphasize the properties that they see as the most important for his project and/or organisation. To determine how suitable a certain ISDM property is for certain IS development project/organisation property the decision model considers various decision rules. We organised these part of the decision model as a rule-based model where the rules are organised in a matrix discussed in subsection 3.4. These rules present a link between ISDM and project/organisation properties.

The decision model facilitates two different types of users shown on Figure 1. The first type is an ISDM expert who has advanced knowledge in the field of ISDM and provides the values for the properties that define a certain ISDM – ISDM descriptions. His responsibility is to select ISDM descriptions that are considered by the decision model during the decision making process and if required he can also to define additional ISDM descriptions. The ISDM expert can also define new decision rules for ISDM selection. Although the general decision rules for ISDM selection are predefined and are part of the decision model the ISDM expert can define additional decision rules that are specific for a certain organisation or project. The second type of user is a business user who requires recommendation about suitability of different ISDMs for his IS development project or IS development organisation. The responsibility of the business user is to provide the values for properties that describe the IS development project or the IS development organisation – project and organisation descriptions. He can also set weights that define which of the project or the organisation properties are more important in his case and consequentially have greater influence on the final decision produced by the decision model.

The decision model can be used in two basic ways. Firstly, it can be used to recommend the most suitable ISDM for the given IS development project or IS development organisation. This recommendation is based on: ISDM descriptions, IS development project or organisation descriptions, and the decision rules that are part of the decision model. Secondly, the decision model can produce a set of ISDM properties that are recommended for certain IS development project or IS development organisation. This set of recommended properties can then be used to adapt and improve an ISDM

that already exists in the organisation. Details on how the decision model produces these recommendations are discussed in subsection 3.4.

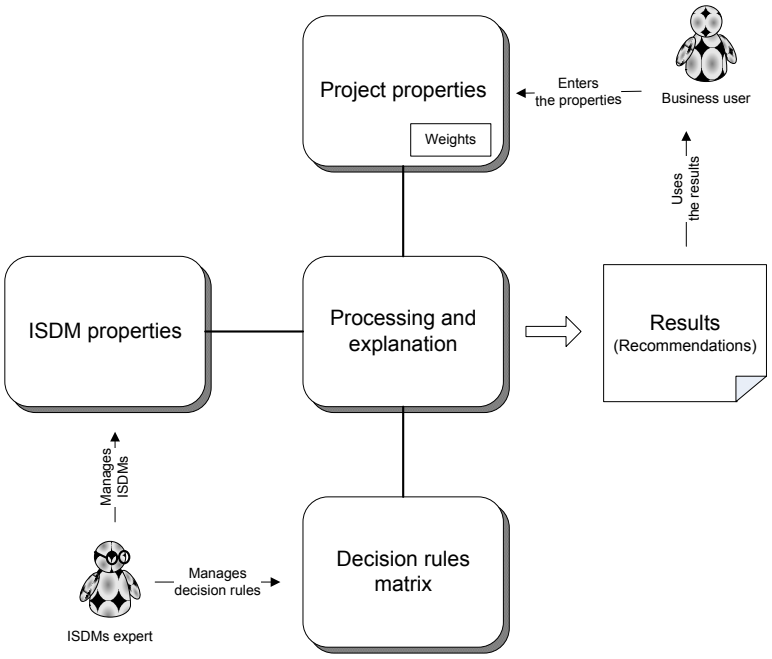


Fig. 1. The main components of the proposed decision model for ISDM selection

We limited the scope of the decision model to ISDM for conventional development of computer-supported IS and did not consider ISDMs for special purposes like ISDMs for ERP systems (e.g. [1]) or people oriented ISDMs (e.g. [27]). The types of ISDM that were considered include: agile ISDMs (e.g. Extreme programming [5]), object oriented ISDMs (e.g. IBM Rational Unified Process [14]), data and process oriented ISDMs (e.g. Information Engineering [24]) and other custom ISDM implemented and used by organisations in which we applied the decision model (e.g. see section 5).

3.1. Project properties

Based on the review of existing studies and models discussed in section 2 and our experience form practical tests of the model discussed in section 5 we propose the set of project properties shown in Table 1. We propose to divide the properties into five main property groups that are further divided into properties for which the predefined values are assigned. For instance, property grup *Size and complexity of the system* has property *Planned future* that can take any of the predefined values: *Maintenance only*, *Minor upgrades*, and *Major upgrades or new versions based on this version*. For

certain properties we also defined quantitative measures that help the user of the decision model to select the suitable value. For instance *System size* can be expressed in man-month, where the definition of size is based on existing definitions (e.g. [40]) and is as follows: very small = less than 30 man-month, small = 30 to 100 man-month, medium = 100 to 250 man-month , large = 250 to 500 man-month, very large = more than 500 man-month. Other properties are defined in a similar manner.

Certainly it is possible to find additional properties that define a project. However, we tried to select properties that could be easily identified for most projects and are important for the decision process. The number and the contents of the properties proved sufficient during the use of the model in most cases.

Table 1. Property groups and properties of project

Property groups	Properties
Size and complexity of the system	System size (defined in man-month or alternatively in equivalent KLOC, number of use-cases, functional points, etc.) System complexity (defined on the basis of number of components and/or architectural layers, number of connections to other systems, etc.) Criticality of the system (defined on the basis of severity of consequences of the system malfunction, a more critical system has to be more reliable) System history (defined on the basis of existence of preceding systems and on how the preceding systems would be used for creation of the new system) Planned future (defined on the basis of predicted future of the system, system can be only temporary and only maintenance is predicted or it can form a foundation for further upgrades and development of new systems even on different platforms, will the number of users of the system raise in the future, etc.)
Project type	Project priorities (defined as two main different priorities: productivity, where the most important aspect is that workable system is developed in short time; and legal liability where it is also important that artefacts are traceable and work on project is documented and tracked) Project predictability (defined by stability of requirements and maturity of business– projects with more stable requirements and more mature business can be better predicted) Time limitations (defines whether the project has very strict

Property groups	Properties
	time limitations that cannot be postponed) Cost limitations (defines whether the project has very strict cost limitations that cannot be changed)
System type	Types of target applications (defines common types of applications developed as the result of a project; these include traditional rich-client desktop applications, web-based applications and mobile applications) Architectures (defines types of architectures used in the project; these include service-oriented architectures & patterns, common object oriented architectures & patterns and structured architectures & patterns) DBMS type (defines types of DBMS used in the project; these include relational DBMS, hierarchical DBMS and object-oriented DBMS) Connectivity (defines the types of communication and connections to external systems used by the project) Legacy support (defines whether there is need for support of legacy technologies which require special approaches and what kind of support is needed)
Development team and environment	Experience in development (defines the level of developers' experience in the field of IS development) Experience in ISDM use (defines the level of developers' experience in the field of ISDM) Problem domain experience(defines the level of developers' experience in the field of system's problem domain) Willingness to learn (defines the level of developers' preparedness to learn to apply new approaches and technologies) Cooperation (defines the level of developers' preparedness to cooperate and share knowledge and experience with their colleagues) Discipline (defines the level of developers' discipline in following the rules and guidelines prescribed by an organisation) Team culture (defines the general team culture that can be autocratic/centralised or democratic/participative) Team location (defines physical location of the development them that can be centralised in one building or dispersed over different continents)

Property groups	Properties
	Team size (defines the size of the development team on a project or typical size of development team in an organisation) Number of development organisations involved in the project (defines the number of different development organisations that cooperate on the same project)
Customer	Requirements regarding ISDM and documentation (defines how rigorous are the customer's requirements regarding ISDM and the documentation produced) Cooperativeness (defines the level of customer's willingness to cooperate with developers especially in specifying the system) Problem domain knowledge (defines the level of customer's knowledge in the target problem domain)

Different projects often stress different aspects of the development. For instance, property called *Legacy support* might play an important role in selection of the right ISDM for a project dealing with upgrade of a legacy system, but is significantly less important on projects that build new IS from scratch, though legacy support might be still desired. Therefore a weight (see Figure 1) can be assigned to each property that enables the user of the decision model to put emphasis on properties that are more important for specific project.

3.2. ISDM properties

We propose a set of ISDM properties shown in Table 2. The properties are grounded on literature discussed in section 2 and our experience gained from application of the decision model in practice (see section 5). We propose to organise the properties in seven main property groups that are further divided into properties for which the predefined values are assigned.

The ISDM descriptions are typically maintained by an ISDM expert and not by the business user of the decision model. The ISDM expert describes an ISDM by assigning the predefined values to the ISDM properties. It is possible that an ISDM has special properties and values that are not predefined, but are important for the decision process. In such case, the ISDM expert can add these new properties and values and also define additional decision rules that are in connection with these new properties and values. However, practical application of the decision model showed that the number and content of our proposed ISDM properties is sufficient for selection of ISDM in most cases.

Table 2. ISDM properties

Property groups	Properties
Process – general properties	Process lifecycles (defines process which lifecycles are supported by an ISDM like waterfall, iterative, incremental etc.)
	Time distribution (defines how the development time is distributed during process; front-loaded ISDM focus on analysis and design while back loaded ISDM focus on implementation and testing)
	Development perspective (defines the primary development perspective of an ISDM that can be top-to-bottom or bottom-up)
	Prototype support (defines whether an ISDM explicitly supports use of prototypes)
	Artefact traceability (defines whether and how well artefact traceability is assured by an ISDM)
Process – primary disciplines scope and detail	Requirements acquisition (defines whether and in how much detail requirements acquisition is described by an ISDM)
	Analysis (defines whether and in how much detail analysis is described by an ISDM)
	Design (defines whether and in how much detail design is described by an ISDM)
	Implementation and integration (defines whether and in how much detail implementation and integration is described by an ISDM)
	Testing (defines whether and in how much detail testing is described by an ISDM)
	Deployment (defines whether and in how much detail deployment is described by an ISDM)
	Maintenance (defines whether and in how much detail maintenance is described by an ISDM)
Process – supportive disciplines scope and detail	Project management discipline (defines whether and in how much detail project management is described by an ISDM)
	Management of development environment (defines whether and in how much detail management of development environment is described by an ISDM)
	Configuration management (defines whether and in how much detail configuration management is described by an ISDM)
	Change management (defines whether and in how much detail change management is described by an ISDM)

Property groups	Properties
Techniques and methods	Modelling techniques and notation (defines which modelling techniques are recommended or prescribed by an ISDM to produce different ISDM artefacts) People and project management techniques (defines which techniques related to people and project management are recommended or prescribed by an ISDM to manage the development team)
Tools and development environment	Development languages (defines the development languages that recommended or suitable for use with an ISDM) Supportive tools (defines whether tools are available that automate certain parts of an ISDM's disciplines) Development frameworks and environments (defines the types of frameworks and development environments that are supported by an ISDM)
Type of development	Types of target applications (defines common types of applications that are supported and are normally developed by an ISDM) Architectures (defines types of architectures that are supported and are normally used by an ISDM) DBMS type (defines types of DBMS that are supported and are normally used by an ISDM) Connectivity (defines the types of communication and connections to external systems that are supported and are normally used by an ISDM) Legacy support (defines whether an ISDM includes support for legacy technologies and what kinds of technologies are supported)
Support and learning	Consistency of concepts throughout ISDM (defines the level of consistency of concepts used in different disciplines and other parts of an ISDM) Consistency of concepts use for a single role (defines the level of consistency of concepts used in different disciplines and other parts of an ISDM that are executed by a single role) Consistency of concepts in supportive tools (defines the level of consistency of concepts used in ISDM with concepts used in tools) Available support (defines what kind of support is available for ISDM)

Property groups	Properties
	Available training for different roles (defines what kind of training is available for ISDM for different roles)

The descriptions of ISDMs are needed in cases when the business user requires concrete names of ISDMs that are suitable for his IS development project/organisation. In cases when the business user only needs information about ISDM properties that are suitable for his project, but does not require a concrete ISDM name, the ISDM descriptions are not used (further discussed in subsection 3.4).

3.3. Decision rules

The decision rules form the backbone of the decision model. Our intention was to develop a set of rules that facilitate selection of ISDM using a computer supported decision model. To define the content of the rules we focused our efforts on collection and composition of existing rules that can be extracted from guidelines and models presented in literature discussed in section 2. It is important that these rules come from practice and have already been tested in real life environment. We transformed these rules from natural language representation found in the literature to format required by the decision model. We avoided changing the content and meaning of the decision rules during the transformation, however minor adaptations and generalizations of some of the rules were required so that they could be used in a computer supported decision model and that they could serve as common guidelines for ISDM selection.

We propose to organise the decision rules in a form of a two-dimensional matrix depicted in Figure 2. The header column and the header row of the matrix contain all ISDM properties and project properties correspondingly. Each property has all of its possible values enlisted. In this manner a matrix containing decision rules for each combination of an ISDM property and a project property is formed. A decision rule for a combination of two properties is written in a sub-matrix containing evaluations for each combination of values for the two properties. There are five evaluations that can be assigned to a certain combination of ISDM property value and project property value: very suitable (2), suitable (1), neutral (0), unsuitable (-1), very unsuitable (-2).

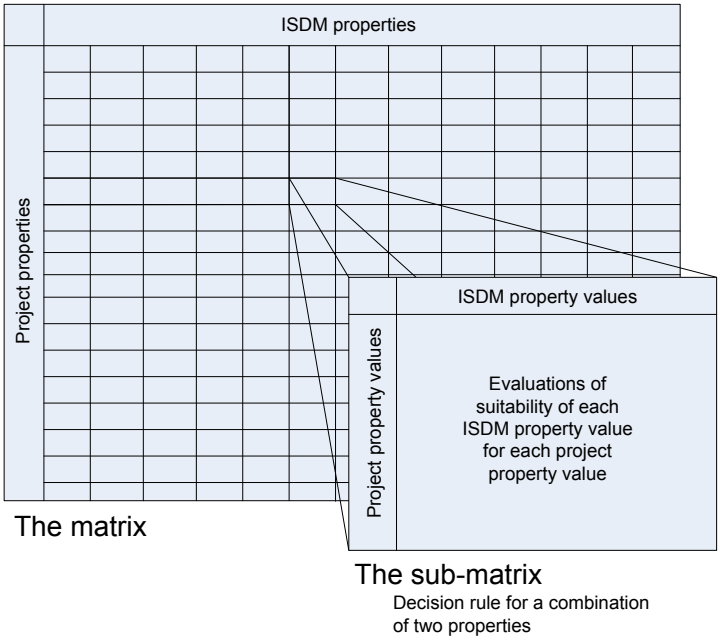


Fig. 2. The matrix containing decision rules

In the following example we demonstrate the structure and function of the decision rule sub-matrix. Table 3 shows an example of such sub-matrix. For the purpose of demonstration we focus only on one sub-matrix i.e. combination of one ISDM and one project property. *Implementation and integration* is one of ISDM's properties (scope of the ISDM). This property can take three different values: *ISDM contains no description of implementation and integration*, *ISDM covers implementation and integration in moderate detail*, and *ISDM covers implementation and integration in detail*. These values form the first of the two dimensions of a decision rule sub-matrix. *Criticality of a system to be developed* is one of project's properties that can take four different values: *system failure can result in a loss of comfort*, *system failure can result in a loss of discretionary money*, s*ystem failure can result in a loss of essential money*, and *system failure can result in a loss of life*. These values form the second dimension of the decision rule sub-matrix. Three different values on the first dimension and four different values on the second dimension form a sub-matrix with twelve possible evaluations, one evaluation for each combination of the two dimensions' values. Based on literature [8] we defined the following rule: "In case the ISDM does not describe *Implementation and integration* at all it is very unsuitable for development of a system, failure of which can result in a loss of life." So we evaluate this combination of values as very unsuitable (-2) in the sub-matrix. Similarly, rules for all other combinations of ISDM property value (*Implementation and integration*) and project property value (*Criticality of a*

system) are defined. Certainly, in some cases the evaluations presented in Table 3 could differ slightly. However, in our experience moderate alterations of some evaluations typically do not affect the final result significantly as there are many other properties that are also considered during evaluation.

Table 3. An example of the decision rule sub-matrix

		Implementation and integration		
		no description	moderate detail	in detail
Criticality of a system to be developed	loss of comfort	0	2	0
	loss of discretionary money	-1	1	1
	loss of essential money	-2	1	1
	loss of life	-2	-1	2

The matrix is designed in a way that it can be expanded. However, adding a new rule is not a trivial task as each new rule requires testing. In our experience, the predefined rules suffice for most of typical cases. Expansion of the rules might be needed in special cases only i.e. to suite special types of organisations and projects. It is important to consider that any new rule should be added by an expert that has profound knowledge in the field of ISDMs and that each new rule should be carefully tested before inclusion in the decision rules matrix.

Table 4. An example of a decision rule explanation in natural language for business user

Project property	ISDM property
Project priorities	Artefact traceability
In traditional ISDM traceability is an important part of the process, however in agile ISDM traceability is not guaranteed [based on [18]]. In case that project priority is traceability, processes that assure traceability are favoured over the processes that do not.	

Furthermore, each decision rule sub-matrix has a corresponding natural language description that is based on the original description of the decision rule found in literature. This description helps the business user to better understand the reasons for recommendations produced by the decision model. Table 4 shows an example of such natural language description of the

decision rule that describes the relation between project property *Project priority* and ISDM property *Artefact traceability*.

3.4. Processing and explanation

In this subsection we discuss our proposed approach for processing the project/organisation descriptions, ISDM descriptions and the decision rules matrix discussed in the preceding subsections in order to produce the recommendations for the selection of an ISDM that suits the needs of an IS development project or an IS development organisation. As recommendation without explanation is rarely sufficient, we propose an approach that also explains why a certain ISDM is preferred over the other for a certain IS development project/organisation.

In cases when it is required to select the most suitable ISDM not only for a project, but for the whole organisation dealing with similar projects the recommendation is produced by using the same approach as for a single project except that average values that are typical for the organisation are used.

By using the proposed approach two main types of results can be produced: firstly, a recommendation of one or more concrete ISDM that suit the project/organisation and secondly, a list of generally recommended ISDM property values for the project/organisation. The first type of result is useful especially in cases when there is a tension to use a well-known predefined ISDMs and a comparison of these concrete ISDMs is required. The second type of result gives a general advice on property values that an ISDM should have for an IS development project/organisation. This type of result can be used either during selection of a concrete ISDM or for development of a new ISDM or improvement of an existing ISDM.

The first type of result is presented as a two dimensional table where heading row contains the names of considered ISDMs, heading column contains names of considered projects and the content of the table are the results of the evaluation for each combination of ISDM and project. We propose the following formula to compute this type of result:

$$\text{Score}(\text{ISDM}_x, \text{Project}_y) = \sum_{a \in A} \sum_{b \in B} (\text{IPT}(a, \text{ISDM}_x) \times (\text{PPT}(b, \text{Project}_y) \times \text{PWT}(b)) \times \text{DRM}(a,b) . \quad (1)$$

where
$A = \{\forall a: a = \text{Characteristic}_{\text{ISDM}}\}$,
$B = \{\forall b: b = \text{Characteristic}_{\text{Project}}\}$,
$ISDM_x$ = an instance of ISDM,
$Project_y$ = an instance of project/organisation,
$IPT(a,b)$ is a cell in column a and row b of ISDM properties table,
$PPT(a,b)$ is a cell in column a and row b of Project properties table,
$PWT(b)$ is a record in row b of Project properties weight table,

DRM(*a*,*b*) is a cell in column *a* and row *b* of Decision rules matrix.

The second type of result is also presented in a two dimensional table where heading row contains the names of all ISDM properties and heading column contains the names of all projects. The content of the table shows the most suitable values of ISDM properties for each project, regardless of whether a concrete ISDM having all these properties exists or not. This type of result is therefore especially useful when the user is planning to develop a new custom ISDM that can actually possess most of such recommended properties. We propose the following formula to compute the recommended value for $ISDMProperty_x$ and $Project_y$:

$$RValue(ISDMProperty_x, Project_y) = DRM(a', r) . \quad (2)$$

where

$ISDMProperty_x$ = the ISDM property for which we want to find the suitable value,
$Project_y$ = an instance of a project/organisation,
r = the number of the row in DRM table that contains the names of property values,
a' = the number of the column in DRM table that contains the name of the most suitable property and is computed by using the following formula:

$$a' = \arg\max_{a \in A\_ISDMProperty\_x}(PVScore(a, b)) . \quad (3)$$

where

$$PVScore(a, b) = \sum_{b \in B}((PPT(b, Project_y) \times PWT(b)) \times DRM(a, b) . \quad (4)$$

and

$A_{ISDMProperty\_x}$ is a subset of columns for $ISDMProperty_x$.

To explain and better understand the recommendations computed by the discussed formulas we propose to compute positive or negative contribution of each individual combination of ISDM property and project property for selected ISDM and project. Using such explanations it is possible to quickly detect the ISDM properties that are evaluated as the most unsuitable for an IS development project/organisation. We propose the following formula to calculate the score of individual combinations of properties:

$$PScore(ISDM_x, Project_y, i, p) = IPT(i, ISDM_x) \times (PPT(p, Project_y) \times PWT(p)) \times DRM(i, p) . \quad (5)$$

where

$ISDM_x$ = an instance of ISDM,
$Project_y$ = an instance of project/organisation,
i = the selected property of $ISDM_x$,
p = the selected property of $Project_y$.

Complementary way of explaining the recommendations that is important especially for business users that are less experienced in the field of ISDM is using natural language descriptions of the decision rules discussed in section 3.3.

4. Implementation of the decision model

Processing of large number of decision rules to produce a recommendation is a complex and time consuming task that cannot be done without appropriate tool support. Therefore, for testing of the decision model we had to use tool support already in the beginning of the research. We developed a prototype tool using MS Excel that enabled us to experiment with different properties and to fine tune the decision rules. This prototype was also used in real projects to facilitate selection of the suitable ISDM. The early version of prototype is presented in [15, 38].

However, although MS Excel is a versatile environment in which we were able to experiment with the decision model quite easily, practical tests showed that this environment has some important drawbacks. Firstly, it was difficult to create a user friendly interface in this environment. This hindered direct use of the prototype outside of the research group. Consequentially, only the results of an evaluation were presented to the companies' experts, but they were unable to test the prototype directly. We did not see this as an important obstacle at the first stage of testing, but later when we tried to enable the companies' experts to experiment with the tool directly this presented a significant difficulty. Secondly, when the number of rules and properties increased the maintenance of the Excel based prototype was becoming more and more awkward. Thirdly, we sought to make the decision support tool accessible through Web so that users could easily access the model without having to install the tool in their computers. However, the model in MS Excel format was not convenient for use in Web environment.

Consequentially, it was necessary to redevelop the tool which implements the proposed decision model. We developed new tool using MS Visual Studio [15]. The tool is designed as a web application that is divided onto the four modules. Figure 3 shows a use-case diagram depicting the main functionality of the tool. The tool is used by two different actors – a business user and a ISDMs expert. The tool allows the business user to maintain projects and their properties, to run analyses and receive the recommendations, and to

explore these recommendations. The ISDMs expert can maintain ISDMs and their properties, add new ISDMs, and modify set of decision rules.

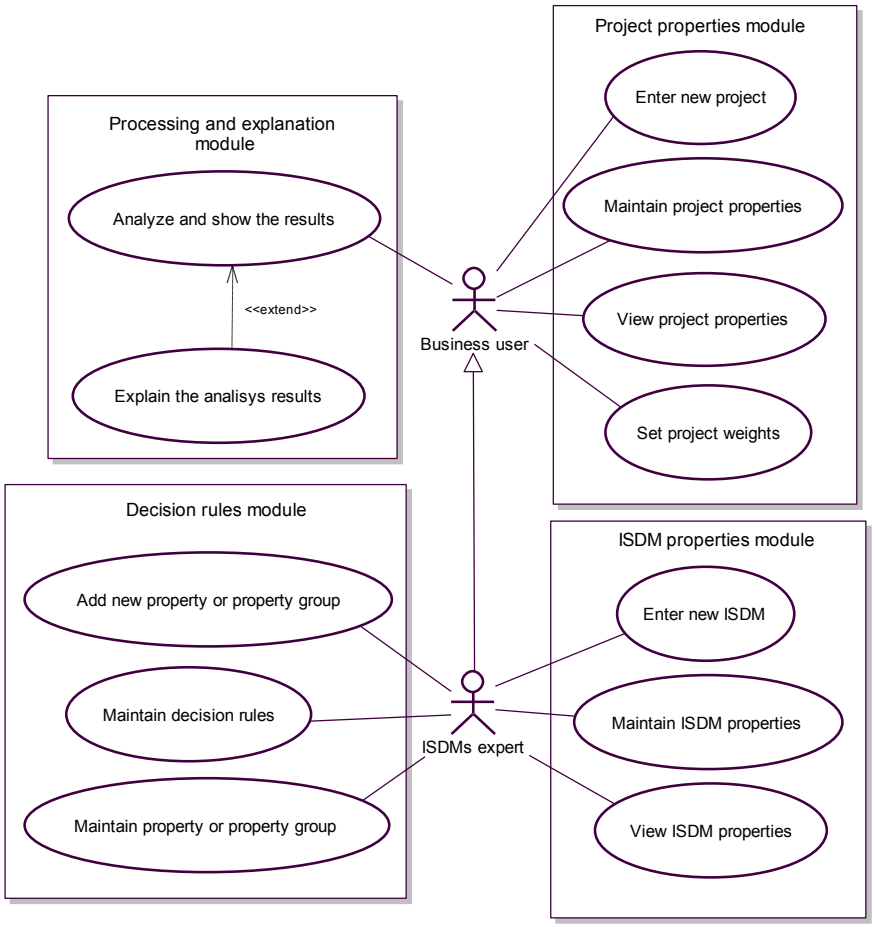


Fig. 3. Use-case diagram depicting the main functions of the tool

The tool uses a relational database to store the data that is the main integration point of the four modules. The E-R diagram in Figure 4 shows the structure of the database. Both ISDM and project properties are stored in the entity type called *Property*. More *Property* values can be assigned to each *Property* instance. The *Decision rule* entity type contains evaluations for each combination of ISDM and project property value thus forming the decision rules matrix. *Decision rule description* contains textual descriptions of rules stored in *Decision rule* entity type and describes one decision rule sub-matrix. The database supports more *Decision rule sets* so that the user can experiment with variations of decision rules. This is especially useful in cases when certain decision rules apply only for certain type of development (e.g.

development of web applications). *Project* and *ISDM* are defined by their *property values* and together form a decision *Case*. The *Case* entity is used to select a subset of ISDM and/or projects which are considered during an evaluation. Each *Property* of *Property group* can be assigned *Property weight* and *Property Group Weight* correspondingly. By using weights the user is able to emphasize the properties that are more important in his case. To ease the experimentation the database supports more *Weight sets*.

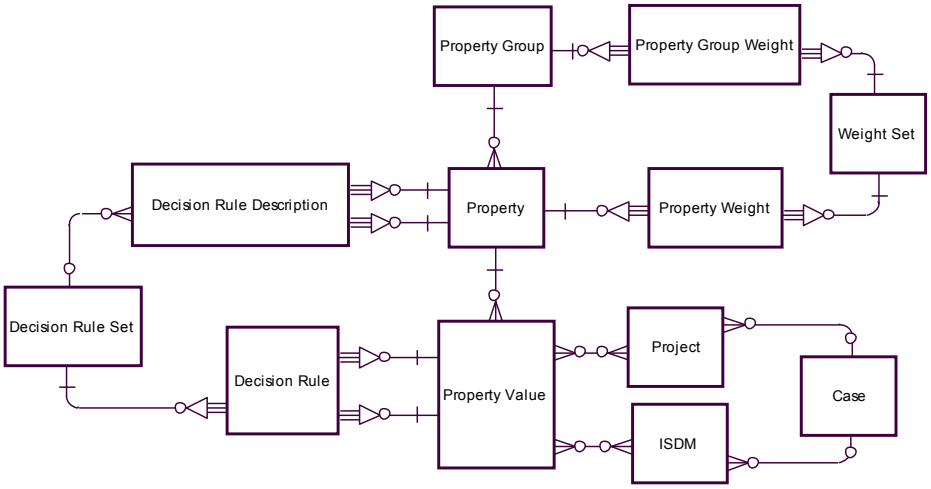


Fig. 4. ER model of the tool's database

Although the project properties module and the ISDM properties module are used to describe two different types of properties – ISDM or project, and are used by two different types of users – business user or ISDM expert, they have quite similar functionality. In both cases the user describes the project or the ISDM by setting the values of the predefined properties. In cases when the user lacks certain information he can choose undefined value. This means that the property will not be considered during evaluation. An addition to project properties module is an interface that allows user to modify weights for different properties of his project. This way the user can define which aspects of development are more important for his case. Figure 5 depicts the main form of project properties module.

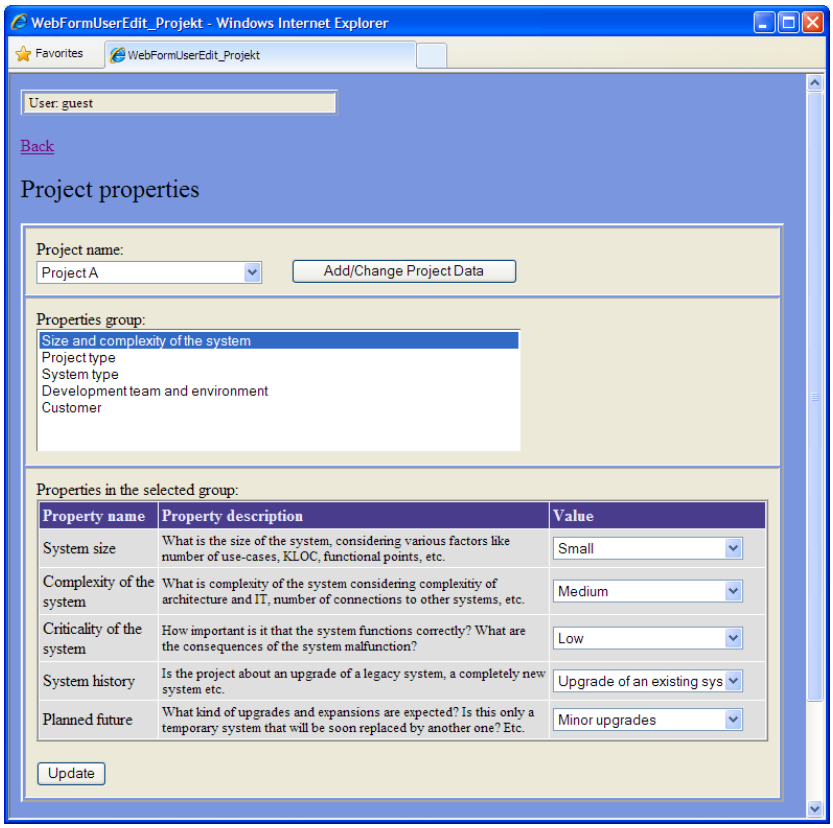


Fig. 5. Screenshot of the main form in Project properties module

Decision rules module is accessible only for an ISDM expert. The primary task of the module is to modify decision rules matrix. The module can store more rule sets which enables the expert to create rule sets for different situations. Besides modifying the rules matrix, the module also allows an expert to add properties and property values. Adding new property or property value is a relatively complex task because the expert has to create a large number of decision rules for each new property. Figure 6 shows the main form of decision rules module that enables an ISDM expert to redefine existing decision rules or to define new decision rules for the chosen decision rule sub-matrix (discussed in section 3.3). Figure 6 shows the decision rule for a combination of project/organisation property *System size* and ISDM property *Requirements acquisition*. In this case one property can take five different values and another can take three different values thus forming a matrix with 15 possible evaluations. As number of different values that can be taken by different properties can vary the form is dynamically generated to show different decision rules. In case that an additional value is added to an existing ISDM property additional rules have to be defined that map this additional value to values of project/organisation properties. In such case the

form shows existing evaluations that form the decision rule and also enables a user to enter the missing evaluations to complete the rule. Below the decision the form shows textual description of the rule that explains the rule and also references the rule to existing research. In case shown in Figure 6 the rule is based on [2] and [8].

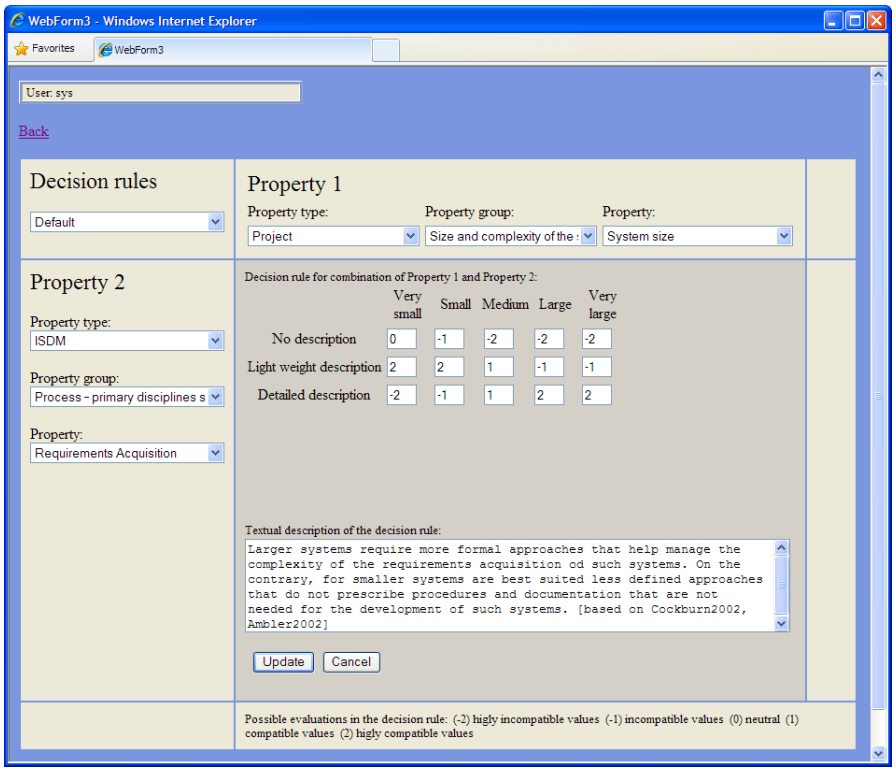


Fig. 6. Screenshot of the main form in Decision rules module

Processing and explanation module produces the recommendation report. The report shows the results of the comparison of different ISDM and their appropriateness for different projects and the most suitable ISDM properties for the projects. The module can also produce explanation report for a selected combination of ISDM and project property as described in section 3.4.

5. Verification of the decision model in practice – case studies

So far the decision model has been applied in five cases i.e. three organisations dealing in IS development and two IS development projects.

The recommendations produced by the decision model were used in the two ways: as input in the process of selection of a suitable ISDM for a new project and as input in the process of improvement of existing organisation's ISDM. These practical applications of the decision model enabled us to verify our proposed two main hypotheses about the decision model.

The first hypothesis was that ISDM experts agree with and approve the recommendations for ISDM selection produced by using the decision model. We asked four ISDM experts to examine the recommendations produced by using the decision model on the five cases. ISDM experts agreed with and approved the recommendations for all five cases, which was not surprising as the decision model's rules are based on generally acknowledged guidelines for ISDM selection available in existing literature discussed in section 2.

The second hypothesis was that the recommendations produced by using the model contain nontrivial information that can be efficiently used to select a suitable ISDM for an organisation or project. We tested this hypothesis in each of the five cases by discussing the results with business users of the decision model i.e. organisation's employees or project members responsible for management of their ISDM who used the decision model. We were especially interested in their opinion whether the model significantly contributed to their understanding of the problem and their decision. On the whole we discussed the results with fourteen business users in the five cases. Six of these users changed their initial decision after using the decision model. Three of the users did not have any initial decision and the model helped them to reach the decision. The initial decision of five users was the same as the model's recommendation. All fourteen users reported that the model helped them to understand the problem of ISDM selection better. They also indicated that they felt more confident in their decision after using the model.

In the following subsections we present two cases in detail. They show how the model contributed to the decision making process during ISDM selection or improvement. In case A the model was used to select a suitable ISDM for a new IS development project and in case B the model was used to improve existing IS development organisations' ISDMs.

5.1. Case A – selecting a suitable ISDM for a new project

Case A was a project where two development teams cooperated to develop and deploy a part of IS for a government institution. The teams came from two IS development companies. Throughout the project the first team that was responsible for the development of the central part of the IS was expected to have from 9 to 12 members and the second team that was responsible for the development of a subsystem was expected to have 5 members. Even though both companies used certain internal standards and frameworks, and had some experience with iterative project management they did not follow any formal ISDM. On both companies' preceding projects the development process was typically organised by a project manager in an

ad hoc manner. However, in this case the government institution required the use of a formally defined IS development documentation and procedures. Although, the institution did not prescribe the use of a particular ISDM, it required that the companies define which documentation would be produced, which activities and in what order would be followed, how the progress of the project would be monitored, etc. Therefore the companies decided that for the needs of the project they would use one of publicly or commercially available ISDM that they would partially adapt to their needs.

The selection of ISDM was difficult as the companies did not have much experience in the field of ISDM. To ease the selection process and to limit the number of ISDM candidates they used the decision model proposed in this paper. The decision model was used by the project manager and some of the team members of the both teams. Initially they needed our help to correctly understand and set the necessary parameters. After a short introduction they were able to use the decision model independently and make experiments with different settings. This experimentation was also one of the main advantages of using the decision model as they could get better understanding of appropriateness of various ISDM.

Eight different ISDM were considered: Ext reme programming (XP), Scrum [30], Rational Unified Process for small projects (RUP for small projects), Rational Unified Process (RUP), Oracle Custom Development Method (CDM), Unified Methodology of IS development (UMISD - used in government institutions), Rapid Application Development (RAD) [25] and a custom ISDM that was developed for the needs of one of the two companies some time ago, but was never used in practice. Table 5 shows the input values used in the evaluation.

Table 5. The input values used in the evaluation

Size and complexity of the system System size = small System complexity = medium Criticality of the system = medium System history = new system Planned future = maintenance and minor upgrades	Development team and environment Experience in development = high Experience in ISDM use = low Problem domain experience = medium Willingness to learn = medium Cooperation = medium Discipline = medium Team culture = open & participative Team location = centralized Team size = small to medium Number of devel. org. involved in the project = 2-3
Project type Project priorities = traceability Project predictability = predictable Time limitations = strict Cost limitations = sufficient	
System type	

Types of client applications = web Architecture = three-tier OO DBMS type = relational Connectivity = present IT Legacy support = not needed	Customer Requirements regarding ISDM and documentation = high Cooperativeness = medium Problem domain knowledge = medium

The recommendation produced by the decision model was RUP for small projects (see Figure 7). This result was somewhat unexpected by the project management that favoured XP at the time. Therefore, the explanation of the recommendation that presented weaknesses and advantages of each ISDM was even more important. The explanation exposed two major weaknesses of using XP in the project. Firstly in case of using XP, there was lack of formal documentation that would be shared among the two teams and presented to the customer, and secondly, there was lack of traceability which was a project priority. Even though XP had an advantage in time to learn, RUP for small projects was considered a better choice as it offered formal documentation and facilitated traceability.

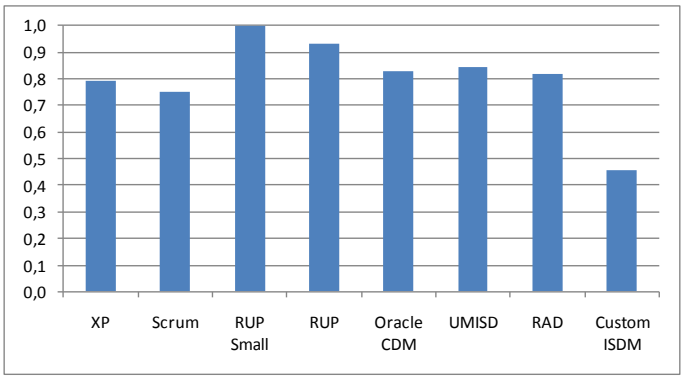


Fig. 7. Normalized evaluation scores for all eight ISDM

After examining the explanation of the results the project management agreed that the RUP for small projects was the better choice for their case. They acknowledged that due to their low experience in the field of ISDM they neglected some of the important aspects of ISDMs and noted that the model helped them to get a better understanding of ISDM field in a relatively short time. The case and the recommendations of the proposed decision model were additionally examined by external ISDM experts who agreed with recommendations produced by the decision model.

A light configuration of RUP for small project was actually used on the project. It enabled the companies to produce the required documentation, assure traceability and successfully complete the project. The decision to use RUP for small projects was actually seen as a compromise between the

customer's requirements for formal documentation and traceability, and companies' existing development experience and initial tendency to use XP.

5.2. Case B – improving existing organisation's ISDM

Case B was an organisation dealing in IS development. Its main field of operation was development of pre-packaged business solutions. The work was organised in four teams each working in one product or project. The typical size of their team that worked on one product was 7 to 10 members, though occasionally they worked on custom projects that involved fewer members. The leader of each team was responsible for organisation of the team's development process. Consequentially, different informal development approaches were used in each team. The management planned to partially formalize the existing informal development process and upgrade it with selected parts of publicly or commercially available ISDMs. The main goal was to standardize their development processes which should enable: employees to easily switch between teams and simultaneously work on more products or projects, easier employment of new employees, better cooperation with other organisations, etc.

The decision model was used in two main ways. Firstly, it was used to help produce a list of recommended ISDM properties for each development team, and secondly, it was used to establish a set of suitable ISDM that could serve as a reference for improvement and standardization of the development process. Six different ISDM were considered: Scrum, Feature driven development (FDD) [29], Extreme programming (XP), Dynamic Systems Development Method (DSDM) [35], RUP and RUP for small projects.

A list of suitable ISDM properties for each of the four development teams was created in cooperation with each team leader. The leaders were encouraged to experiment with the decision model to get better understanding of the ISDM field and the needs of their team. Our initial evaluation showed that all four teams share similar characteristics. Not surprisingly, the recommendations produced by the model for the three of the four teams were quite similar (see Figure 8). In all three cases less rigorous approaches that follow agile principles were recommended and Scrum and XP were found to be the most suitable ISDMs. However, there were important differences in the recommendations for the remaining team for which more rigorous properties were recommended and RUP for small projects was proposed as the most suitable ISDM. After performing detailed analysis of the results and discussion with the team leaders, we discovered that the leader of the fourth team had different expectations of ISDM than leaders of the other three teams. The first three leaders put emphasis on simplicity of ISDM as they believed that learning and adoption of ISDM were the largest obstacles. The opinion of the fourth leader was that learning ISDM is not so difficult, so he did not see this as an important obstacle. His expectations focused mainly on establishing traceability and formality for his projects. Consequentially, the model's recommendation was a more rigorous ISDM.

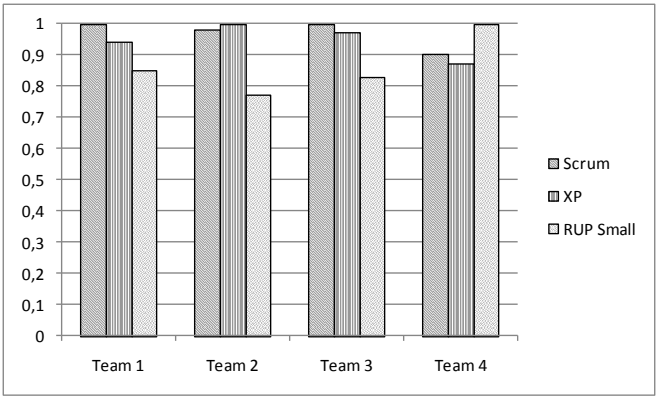


Fig. 8. Normalized evaluation scores for Scrum, XP and RUP for Small projects for each of the four development teams

To reach the final decision the advantages and disadvantages of the three recommended ISDM were presented to the leaders. They agreed to use Scrum as a reference ISDM. They would formalize and adapt their development process to follow basic principles of Scrum, however on projects requiring more formality they would add certain RUP's artefacts to Scrum's Product backlog [30] to assure better traceability.

All four project leaders found the decision model to be a useful tool that enabled them to gain understanding of appropriateness of various ISDM faster. They stressed the importance of possibility of experimentation with the model and confirmed that information offered by the decision model was relevant. The recommendations were also reviewed by external experts who validated that they are suitable for the needs of the organisation.

6. Conclusion

The analysis of related literature (e.g. [12]) and our experience shows that organisations and IT departments dealing with development of computer-based business IS often lack in-depth knowledge of ISDM field. This hampers their selection of ISDM and often limits their choice to ISDM vendors who favour their own ISDM. To improve this situation and help organisations to actively participate in the process of ISDM selection, we propose a decision model and a tool based on the decision model for help in ISDM selection.

The decision model and the tool have been applied in several practical cases confirming that they can be efficiently used to improve the ISDM selection process and help make better decision. The decision model does not substitute ISDM experts, but is merely a tool that on the one hand helps people responsible for ISDM selection to make a more informed decision and on other hand helps an ISDM expert to obtain better understanding of the

needs of an IS development organisation. An important precondition for successful application of the decision model is that users of the decision model are cooperative and prepared to learn through experimentation with the decision model.

Our further work will focus on extension of the decision model's characteristics and rules. We intend to focus especially on ISDM that include explicit support for service oriented architecture (e.g. [13, 23]), service oriented frameworks (e.g. [36]) and approaches that support model driven architecture (e.g. [20]). Furthermore, we intend to examine possibilities to apply the decision model in the field of specialized methodologies like methodologies for building ontology (e.g. [40]), methodologies dealing in enterprise architecture (e.g. [39]), development approaches based on business rules perspective (e.g. [7, 19, 33]) and other alternative approaches and studies on existing ISDM (e.g. [22, 32]).

References

1. Ahituv, N., Neumann, S.,Zviran, M.: A system development methodology for ERP systems. The Journal of Computer Information Systems, Vol. 43, No. 3, 56-67. (2002)
2. Ambler, S.W., Jeffries, R.: Agile Modeling: Effective Practices for Extreme Programming and the Unified Process John Wiley & Sons. (2002)
3. Avison, D.E., Fitzgerald, G.: Information systems development : methodologies, techniques and tools. 3rd ed. McGraw-Hill, London, xv, 592 p. (2003)
4. Bajec, M., Vavpotic, D.,Krisper, M.: Practice-driven approach for creating project-specific software development methods. Information and Software Technology, Vol. 49, No. 4, 345-365. (2007)
5. Beck, K., Andres, C.: Extreme Programming Explained: Embrace Change. 2 ed. Addison-Wesley Professional, 224. (2004)
6. Brinkkemper, S., Lyytinen, K.,Welke, R.J.: Method Engineering – Principles of method construction and tool support. IFIP TC8. Chapman & Hall, Atlanta, USA. (1996)
7. Ceponiene, L., Nemuraite, L.,Vedrickas, G.: Separation of event and constraint rules in UML&OCL models of service oriented information systems. Information Technology and Control, Vol. 38, No. 1, 29-37. (2009)
8. Cockburn, A.: Agile software development. Agile software development series. Addison-Wesley, Boston, 304. (2002)
9. Euromethod: Welcome to Euromethod. (2011) [Online]. Available: http://projekte.fast.de/Euromethod/ (current 2011 28th of June)
10. Fitzgerald, B.: Am empirical investigation into the adoption of systems development methodologies. Information & Management, Vol. 34, No. 6, 317-328. (1998)
11. Highsmith, J.: Adaptive software development: A Collaborative Approach to Managing Complex Systems. Dorset House Publishing, 392. (2000)
12. Highsmith, J.: Agile Software Development Ecosystems. Addison Wesley, 448. (2002)
13. IBM: IBM RUP for Service-Oriented Modeling and Architecture v 2.4 (IBM Rational Method Composer Method plug-in). IBM Corp. (2006)

14. IBM: Rational Unified Process v 7.0.1 (IBM Rational Method Composer plugin). IBM Corp. (2006)
15. Infolab: ISDM Selection Tool Web Page. (2009) [Online]. Available: https://infolab.fri.uni-lj.si/ISDMSelection (current 2010 12/2)
16. ISACA: COBIT Framework for IT Governance and Control. (2011) [Online]. Available: http://www.isaca.org/Knowledge-Center/COBIT/Pages/Overview.aspx (current 2011 28th of June)
17. ISPL: Information Services Procurment Library. (2011) [Online]. Available: http://projekte.fast.de/ISPL/ (current 2011 28th of June)
18. Jacobsson, M.: Implementing Traceability In Agile Software Development, in Department of Computer Science. Faculty of Engineering, LTH: Lund, Sweden. (2009)
19. Kalibatiene, D., Vasilecas, O.: Ontology Axioms for the Implementation of Business Rules. Technological and Economic Development of Economy, Vol. 16, No. 3, 471–486. (2010)
20. Karsai, G., Neema, S.,Sharp, D.: Model-driven architecture for embedded software: A synopsis and an example. Science of Computer Programming, Vol. 73, No. 1, 26-38. (2008)
21. Kitchenham, B., Linkman, S.,Law, D.: DESMET: a methodology for evaluating software engineering methods and tools. Computing and control engineering journal, Vol. 8, No. 3, 120-126. (1997)
22. Lavbič, D., Lajovic, I.,Krisper, M.: Facilitating information system development with Panoramic view on data. Computer Science and Information Systems, Vol. 7, No. 4, 737-767. (2010)
23. López-Sanz, M., et al.: Modelling of Service-Oriented Architectures with UML. Electronic Notes in Theoretical Computer Science, Vol. 194, No. 4, 23-37. (2008)
24. Martin, J.: Information Engineering, Book I: Introduction. Prentice Hall. (1989)
25. McConnell, S.: Rapid development : taming wild software schedules. Microsoft Press, Redmond, Wash., xix, 647 p. (1996)
26. Mikulenas, G., Butleris, R.: An approach for constructing evaluation model of suitability assessment of agile methods using analytic hierarchy process. Electronics and Electrical Engineering, Vol. 10, No. 106, 99-104. (2010)
27. Mumford, E.: Effective Requirements Analysis and System Design: The ETHICS Method. Macmillian, Basingstoke, UK. (1995)
28. OGC: Welcome to the Official ITIL Website. (2011) [Online]. Available: http://www.itil-officialsite.com/ (current 2011 28th of June)
29. Palmer, S.R., Felsing, J.M.: A practical guide to feature-driven development. Prentice Hall. (2002)
30. Rising, L., Janoff, N.S.: The Scrum software development process for small teams. IEEE Software, Vol. 17, No. 4, 26-32. (2000)
31. Saaty, T.L.: Decision Making for Leaders: The Analytic Hierarchy Process for Decisions in a Complex World. RWS Publications, Pittsburgh, Pennsylvania. (2008)
32. Sánchez, P., et al.: Model-driven development for early aspects. Information and Software Technology, Vol. 52, No. 3, 249-273. (2010)
33. Smaizys, A., Vasilecas, O.: Business Rules Based Agile ERP Systems Development. Informatica, Vol. 20, No. 3, 439-460. (2009)
34. Stair M., R., Reynolds W., G.: Principles of Information Systems, A Menagerial Approach. Eight Edition ed. Thomson course technology. (2008)
35. Stapleton, J.: DSDM Dynamic Systems Development Method: The Method in Practice. Addison Wesley. (1997)

36. Šaša, A., Jurič, M.B.,Krisper, M.: Service-oriented framework for human task support and automation. IEEE transactions on industrial informatics, Vol. 4, No. 4, 292-302. (2008)
37. Triantaphyllou, E.: Multi-Criteria Decision Making Methods: A comparative Study. 1 ed. Applied Optimization. Vol. 44. Kluwer Academic Publishers, 320. (2000)
38. Vavpotic, D., Bajec, M.,Krisper, M.: Software development methodology evaluation model. In 12th International Conference on Information Systems Development. Kluwer Academic / Plenum Publishers, New York, Melbourne. (2004)
39. Wegmann, A.: On the Systemic Enterprise Architecture Methodology. In International Conference on Enterprise Information Systems (ICEIS). 483-490. (2003)
40. Wongthongtham, P., et al.: Ontology-based multi-site software development methodology and tools. Journal of Systems Architecture, Vol. 52, No., 640-653. (2006)
41. Yin, R.K.: Case study research : design and methods. 3rd ed. Applied social research methods series ; v. 5. Sage Publications, Thousand Oaks, Calif., 200. (2003)
42. Yoon, K.P., Hwang, C.-L.: Multiple Attribute Decision Making: An Introduction (Quantitative Applications in the Social Sciences) Sage Publications, Inc, Thousand Oaks, London, New Delhi, 83. (1995)

Damjan Vavpotič. Dr. Damjan Vavpotič is a senior-lecturer of Information Systems and Information Systems Development in the Faculty of Computer and Information Science at the University of Ljubljana. He received his doctorate in Information Systems Engineering from University of Ljubljana, Slovenia. His research interests include information systems development methodologies, methodology adoption and evaluation, and agile methodologies. His research has appeared in journals such as Information and Software Technology, Informatica, Applied Informatics, and Nurse Education Today, and in proceedings of many international conferences. He participated in many projects dealing with evaluation and improvement of ISDMs, IS strategic planning, and IS development.

Olegas Vasilecas. Prof. habil. (hp) dr. Olegas Vasilecas is full time professor at the Information Systems Department, and head of Information Systems Research Laboratory in Vilnius Gediminas Technical University. He is author of more than 200 research papers and 5 books in the field of information systems development. His research interests: knowledge, including business rules and ontology, based information systems development. He delivered lectures in 7 European universities including London, Barcelona, Valencia, Athens and Ljubljana. O. Vasilecas carried out an apprenticeship in Germany, Holland, China, and last time in Latvia and Slovenia universities. He supervised 7 successfully defended doctoral theses and now is supervising 5 more doctoral students. He was leader of many international and local projects. Last time he leaded „Business Rules Solutions for Information Systems Development (VeTIS)“ project carried out under Lithuanian High Technology Development Program.

Received: March 15, 2011; Accepted: September 1, 2011.

DOI 10.2298/CSIS110503072V

Improving the Evaluation of Software Development Methodology Adoption and its Impact on Enterprise Performance

Damjan Vavpotič[1] and Tomaž Hovelja[1]

[1]University of Ljubljana, Faculty of Computer and Information Science,
Tržaška 25, 1000 Ljubljana, Slovenia
{damjan.vavpotic, tomaz.hovelja}@fri.uni-lj.si

Abstract. Although the literature studying software development methodologies (SDMs) lists several significant positive effects of the deployment of SDMs, investments into SDMs by the enterprises remain relatively limited. Strategic investments decisions, such as SDMs investments, are mostly taken with the goal of improving enterprise performance. In this paper a model for evaluation of the adoption of SDMs that focuses on the abovementioned SDMs impact on enterprise performance is proposed. The model was empirically tested in four case studies in software development small and medium enterprises (SMEs) in Slovenia. The case studies confirmed that the use of the proposed model enabled SMEs to improve SDMs related investment and adoption decisions and enabled SMEs to invest their limited resources in the most productive and competitive way. The case study experience with the proposed model suggests that its use would also bring similar benefits to larger software development enterprises.

Keywords: Software Development Methodologies, Enterprise Performance, SDM Adoption, Evaluation Approach.

1. Introduction

A SDM can be defined as a collection of procedures, techniques, tools, and documentation aids which will help the system developers in their efforts to implement a new information system [4]. In the past decades various formal SDMs emerged that were based on different underlying philosophies and were developed in both academic and commercial environments. The main motive for their creation was to provide efficient software development procedures that would produce better software for computer supported information systems at an acceptable cost. However, many enterprises dealing with software development do not use formal SDMs and rely mainly on ad hoc development procedures. Fitzgerald [21], for instance reports that 60 per cent of enterprises do not use any SDM and that only 14 per cent use a formalised commercial SDM. Different reasons for this situation have been

identified [33, 44, 59] and considerable efforts have been invested into improvement of SDM adoption in software development enterprises [20, 51, 52]. Most of these efforts considered SDM adoption as an IT decision that technicians should decide upon and often neglected the role of business management. Moreover, the assumption that SDM adoption is an IT decision is frequently encouraged by the rest of the enterprise [4]. However, application of SDM typically involves a considerable investment in time, effort and money. Therefore, it is argued that business managers in general should participate more actively in such decisions [4]. To increase business manager's involvement in SDM investments it is important to better understand the impact of SDM on enterprise performance. However, knowing only the potential impact of a SDM procedure on enterprise performance does not enable managers to enact proper SDM related improvement actions. They also need to understand the level of adoption of a specific SDM procedure, since only sufficiently adopted SDM procedures deliver actual enterprise performance benefits. Therefore only the combined understanding of adoption of SDM procedures and their impact on enterprise performance can form the appropriate basis for managers' investment decisions concerning SDM in the enterprises.

In order to significantly improve the managerial organising vision of what for, how and how much should SDM be used the following research questions need to be posed:

1. How can the evaluation of SDM procedure adoption and impact on enterprise performance be modelled? (RQ1)
2. Can such model identify SDM procedures with important enterprise performance benefits? (RQ2)
3. Can such model provide nontrivial information about SDM procedures that was previously unknown to managers responsible for SDM management? (RQ3)
4. Can such model improve SDM related investment and adoption decisions? (RQ4)

To answer these questions the paper starts with the review of the relevant literature in section 2. Based on this review a model is proposed that evaluates the adoption of SDM procedures and their influences on enterprise performance in section 3. Next, in section 4, the paper addresses all the methodological issues that had to be resolved before the conceptual model could be empirically tested. Finally, in section 5, the results of the multiple case study are presented and the paper concludes with a discussion about the technological and managerial implications of the results.

2. Literature review

The research on adoption of SDM and other software process innovations (SPI) in organisations dealing with software development [24, 26, 28] is often based on Rogers' diffusion of innovations theory (DOI) [45]. This universal theory attempts to explain why certain innovations spread among their potential users while others remain unused. Researchers in the field of SPI and SDM consider a SDM or its parts as an innovation and try to predict and explain target adopter attitudes and their innovation-related behaviour [23]. Beside DOI other innovation diffusion models and theories like the Theory of Planned Behaviour (TPB) [2], Technology Acceptance Model (TAM) [15, 54], Perceived Characteristics of Innovating (PCI) [37], and the Theory of Reasoned Action (TRA) [19] can be used to predict/explain adoption of innovations in the field of SDMs. These models focus on SDM adoption mostly from sociological, psychological or cultural aspects of target adopters. However, they typically do not consider business aspects of adoption that are of key importance for management decisions. On the other hand, general claims regarding the positive influence of SDM on enterprise performance have been made not only by SDM vendors (e.g. [30]) but also by different researchers (e.g. [4]). Thus this gap remains to be bridged.

The research on SDM adoption typically observes a SDM as a whole or observes only a single procedure of a SDM (e.g. procedure of unit testing), but does not consider a SDM as a composition of interrelated procedures. For the purpose of this study a SDM procedure is defined as a comprehensive unit that comprises different SDM activities, tasks, tools, documentation templates, design techniques etc. that are used to perform a specific part of software development like functional testing procedure, software documentation procedure, requirements acquisition procedure, etc. Due to the dynamic environment and constantly changing requirements it is hard to adopt only one SDM with strictly defined SDM procedures [34]. Therefore it is important to observe a SDM as a composition of interrelated procedures. This improves the understanding of the adoption of specific SDM procedures and their impact on enterprise performance. In this way the differences in adoption levels and performance impacts between different SDM procedures are not overlooked. The importance of considering a SDM as a composition of different parts is recognised also by the research in the field of situational method engineering [9, 32, 43] that attempts to construct a SDM suitable for a certain situation from procedures and other parts of different existing SDMs. As it is probable that a SDM is constructed from procedures of different SDMs, the performance and adoption of a SDM can be better studied on the level of its individual parts [51] or in our case procedures.

To determine the level of individual SDM procedures' adoption two dimensions were measured:

1. Frequency of SDM procedure use in a case of a given opportunity (FrqUse).
2. Frequency of opportunities for SDM procedure use (FrqOpp).

Even though existing research in the field of SDM adoption commonly uses only a single frequency measure of SDM use (e.g. [27]) new research shows that such approach is inadequate when measuring the frequency of use of individual SDM procedures [51]. The inadequacy stands in the difference of opportunities for use (FrqOpp) and actual use (FrqUse) of different SDM procedures. FrqUse measures how often software developers apply a certain SDM procedure in a case that an opportunity for its use arises during the development. On the other hand FrqOpp measures how frequently an opportunity to use a certain SDM procedure arises during the software development disregarding whether the SDM procedure is actually used or not. Used alone, frequency of SDM use is of limited value as a measure because opportunities for use can vary widely. It needs to be combined with a measure of the number of opportunities for its use.

After appropriate measures of SDM adoption were developed for the proposed model adequate enterprise performance measures were selected to complete the proposed model. The most appropriate enterprise performance measures were selected on the basis of the empirical models from the literature that studies the impact of IT on enterprise performance, theories of the enterprise (firm) and the literature on (IT) project success criteria. In standard neo-classic microeconomics the enterprise is the economy's basic unit of production that combines inputs into outputs at the lowest cost possible in order to maximize its profits [11, 12, 42]. Thus in standard micro-economic models variables of productivity (output/input) and/or profitability (profit/input) are the most widely used as measures of enterprise performance.

Therefore it cannot come as a surprise that the most established measure in the empirical literature that studies the economic impact of IT is added value per employee and its growth [16, 17]. The reason why this measure of productivity established itself over the different measures of profitability can probably be found in the fact that it allows the researchers to avoid the long lasting discussion if enterprises really behave as profit maximizers [14, 47]. When it comes to Slovenian enterprises there is ample evidence that the majority of enterprises do not behave as profit maximizers [5, 8, 41] thus the profitability measures of enterprise performance were not used. To find the appropriate enterprise performance measures a pre-case focus group of 6 SME managers was formed. The focus group found it difficult to objectively evaluate the additional added value (output) generated by a specific software part developed by using certain SDM procedures, despite the fact that the literature posits additional added value as the best enterprise performance criteria. Instead of added value managers saw costs as a more tangible measure that they routinely used to asses SDM performance. Therefore, as the managers were unable to confidently evaluate differences in outputs, the proposed model had to use the differences in costs (input) as the main productivity measure of performance (Cost).

Even though the above developed measure of enterprise performance is in accordance with the majority of empirical models studying the impact of IT on

enterprise performance and neo-classic theory the findings of other important organizational, behavioural, managerial and strategic theories of the enterprise should not be ignored. These theories do not see enterprises as one-dimensional profit-maximizers but complex entities with many different and sometimes conflicting goals [6, 14, 22, 25, 38, 42, 46, 58]. Thus grounding the proposed model in different theories of the enterprise requires the measurement of other dimensions of enterprise performance in addition to measurement of enterprise productivity. One of these additional measures of enterprise performance that was included in the proposed model is how a SDM affects enterprises ability to reach their goals (Goal), since reaching the goals is stressed as a key measure of enterprise performance by several theories of enterprises.

Unfortunately many times the goals enterprises set themselves are far from being in line with the interests of the key environmental stakeholders (customers, business partners) [40, 53]. For this reason in addition to measuring the impact of SDM on enterprise productivity and on enterprise ability to achieve their key goals an additional dimension of SDM influence on enterprise performance was introduced. This third dimension measures how SDM benefits the environmental stakeholders through its impact on the improvement of enterprise's products and services (Prod).

Our three goals differ from the classic iron triangle (cost, time, quality) traditionally used as (IT) project success criteria in that they incorporate all the recent findings in the relevant literature [1, 3, 31, 36, 49, 56, 57] about the iron triangle limits. Thus the success criteria were broadened to include measures of organisational strategic success and environmental stakeholder success as follows:

1. The time criterion was broadened to include the organisational (strategic) goals (Goal).
2. Instead of just scope or quality the environmental product value for the customers and business partners of the produced software [3, 7, 31] was measured (Prod).
3. From the original classic iron triangle only costs remained unmodified in the proposed model, since the managers of the pre-case focus group preferred costs as a measure of productivity to additional added value (Cost).
4. The three above developed measures of SDM impact (Cost, Goal, Prod) together with the two SDM adoption measures (FrqOpp, FrqUse) form the core of the proposed model for evaluation of SDM procedure adoption and their impact on enterprise performance. The proposed model can be seen in Figure 1.

3. The proposed model for evaluation

Figure 1 shows the three steps that were employed to study the impact of SDM procedures on enterprise performance. The first step was to catalogue SDM procedures used in the studied enterprises. In this step a focus group comprised of technical managers, SDM users and external experts made a list of all SDM procedures available to the enterprise. Technical managers were employees that had a comprehensive overview of both technical and business aspects of studied SDM procedures, while SDM users were directly involved in the use of these SDM procedures. The cataloguing process was guided by external experts that assured that a comprehensive and complete list of SDM procedures was generated. In the second step each catalogued SDM procedure was evaluated.

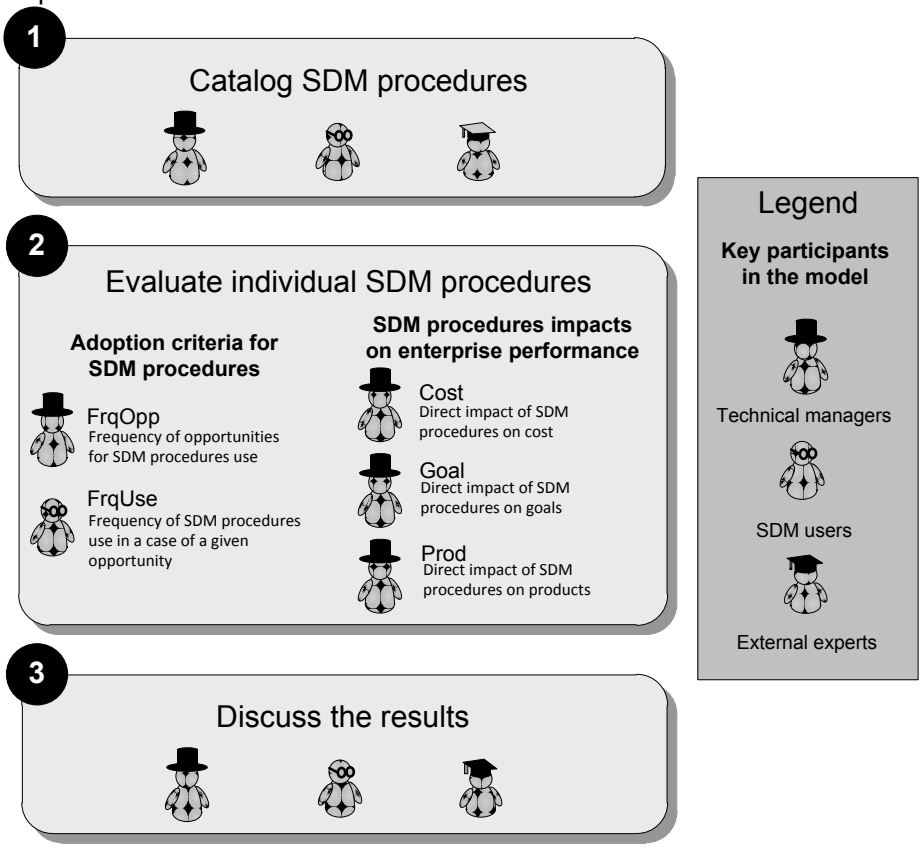


Fig. 1. The proposed model for evaluation of SDM procedure adoption and their impact on enterprise performance

The evaluation was performed by individual SDM users and technical managers. SDM users evaluated only frequency of use in case of given

opportunity (FrqUse) of the SDM procedures that were related to their everyday work. While frequency of opportunities for SDM procedures use (FrqOpp) was evaluated by technical managers as only they possessed sufficient knowledge about these procedures to evaluate their FrqOpp objectively. For the same reason they also evaluated the direct impact of these SDM procedures on costs (Cost), goals (Goal) and products (Prod). The last step was an in-depth discussion of results from the second step with evaluation participants. With the development of the proposed model the first research question (RQ1) posed in this paper is addressed.

There are no theoretical reasons that would limit the scalability of the proposed model concerning the number of SDM procedures evaluated and the number of evaluation participants. For this reason the usefulness of the model should not be affected by the size of the enterprises. However, due to the fact that SMEs software development enterprises are the most widespread type of software development enterprise in Slovenia their study was a priority. Therefore the paper presents case studies performed in SMEs. Nevertheless, further testing of the proposed model in larger enterprises should be performed to confirm this assumption of scalability.

4. Methodology

The research methodology is based on embedded multiple case design with replicability analysis as defined by Yin [60]. Adoption of SDM procedures and their impact on enterprise performance in different software development contexts that are common in SMEs in Slovenia were studied. SDMs were studied on level of their procedures as stated in the literature review. The study had to take into account that adoption of a SDM is a long-term process with impacts which can be observed only over longer time periods. Thus it focused on perceptions of primary characteristics of SDM procedures (adoption levels, economic impact) of key employees that have a good understanding of their SDM. Researchers in the field of IT innovation adoption often use perceived characteristics instead of directly measuring primary characteristics. According to Downs and Mohr [18] the findings of many studies, which have examined the primary characteristics of innovations, have been inconsistent, as primary attributes are intrinsic to an innovation independent of their perception by potential adopters. The behaviour of individuals, however, is predicated by how they perceive these primary attributes. Since different adopters might perceive primary characteristics in different ways, their eventual behaviours might differ. Similar approach was used also in other SDM studies [29, 51].

The proposed model was used in four different cases: a software development enterprise dealing with development of web-based applications (case A), two different software development enterprises that produce their own pre-packaged business solutions software (cases B and C), and an enterprise producing financial software (case D). The basis for the evaluation

of SDM procedure adoption (FrqUse and FrqOpp) and their impact on enterprise performance (Goal, Cost and Prod) were questionnaires that were used in each single case. They included the 2 dimensions of SDM adoption and the 3 dimensions of enterprise performance and were filled out partially by technical managers and partially by SDM users of the studied enterprises. The questions about FrqUse (how often is a SDM procedure actually used in a case of a given opportunity) and FrqOpp (how often opportunities for use of SDM procedure arise) were evaluated with 7-point ordinal scales (never=1, very seldom=2, seldom=3, sometimes=4, often=5, very often=6, always=7). The questions about SDM procedure impact on enterprise performance were also close-ended questions, however they used a seven-point Likert scale between 7 (strongly agree that a specific SDM procedures affects a specific dimension of enterprise performance) and 1 (strongly disagree that a specific SDM procedures affects a specific dimension of enterprise performance), where 0 meant neither agreement or disagreement with the statement.

The results are presented on a scatter chart comprising FrqUse as a vertical dimension and FrqOpp as a horizontal dimension. To help direct management efforts in improvement of individual SDM procedures four quadrants were formed in the scatter chart by using the medians of FrqUse and FrqOpp. Medians were used instead of means of scale because of the positive perception bias encountered during case studies which is discussed in detail in section 6. The four quadrants follow the logic suggested by Vavpotic and Bajec [51]: the first quadrant contains inefficient and unadopted SDM procedures, the second quadrant contains inefficient but adopted procedures, the third quadrant contains unadopted but efficient procedures and the fourth quarter contains adopted and efficient procedures. Efficient SDM procedures are those that have a lot of opportunities for use during the software development process (FrqOpp higher than median) while adopted SDM procedures are those that are actually used in the software development process (FrqUse higher than median).

For procedures in the first quadrant managers should investigate whether it makes sense to continue investing in them. An example of such procedure could be a procedure that is less appropriate for the needs of an organisation and has never been popular between SDM users. A possible course of action would be to discard such SDM procedures (e.g. an old procedure for system design based on structured methods in an enterprise that now uses object oriented development environment). For procedures in the second quadrant managers need to focus on creating more opportunities for their use as these procedures are already well used when an opportunity for their use arises. For instance, these can be SDM procedures that had a lot of opportunities for use in the past and are still accepted by SDM users, but are technically less appropriate for development of new systems. A recommended course of action to improve such procedure might be to replace accepted procedures with technically more appropriate procedures that do not require of SDM users to drastically change the way they work (e.g. procedures that use Enterprise generation language [39] might be introduced instead of procedures that employ Java to replace procedures based on an old

programming language that developers are familiar with). The third quadrant shows procedures that have a lot of opportunities for use but are not used. An example could be a new procedure that is very suitable for development of new systems but are not yet accepted by SDM users. In such case managers should try to improve acceptance of such SDM procedures by additionally training and educating SDM users (e.g. procedures for test automation might be introduced, but not accepted by testers that are accustomed to manual test procedures therefore the testers should be additionally trained in the field of test automation). The fourth quadrant shows procedures that have a lot of opportunities for use and are well used by SDM users. Such procedures should be monitored so that when they fall from the fourth quadrant management can deploy an appropriate course of action (e.g. procedure for data modelling that is technically efficient and also accepted by SDM users should be monitored to detect technological advances that could make this procedure obsolete).

Although the suggested separation of SDM procedures into the four quadrants enables managers to select a general course of action [51] it does not help them to understand the value of individual SDM procedures for the enterprise which would in our opinion significantly improve managerial SDM related investment and adoption decisions (RQ4). For this reason each SDM procedure is presented by a point in one of the four quadrants of the scatter chart that is additionally described by its average evaluated impact on cost (marked as C), goal (marked as G) and product (marked as P).

Finally, the results of the evaluation and possible courses of action are discussed with the evaluation participants. In this way the evaluation participants are involved not only in the collection of facts, but also in the co-construction and interpretation of the case narrative [10].

The four cases are described in the following subsections. The cross-case discussion is included in the last subsection.

5. Case studies

5.1. Case A

Case A is a small enterprise focusing on development of web-based applications. Their target customers are enterprises that require web presence from simple web pages to more advanced customer oriented web applications like web-shops, customer web support etc. They use their own SDM that was mainly developed inside the enterprise and is based on structured SDMs [4]. It comprises 21 procedures describing activities that have to be performed and artefacts produced, for instance: testing the user interface, introducing new code into configuration management system, documenting program code, changing the data model etc. These procedures

also define various programming, testing and documenting standards. The SDM is regularly updated and modified to facilitate the application of new IT.

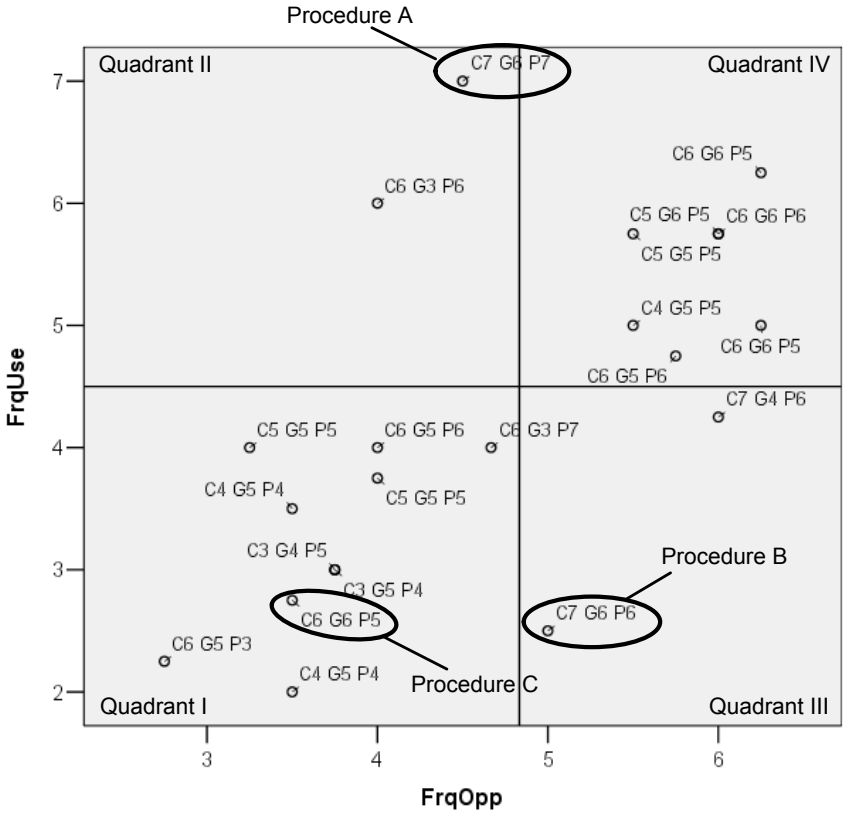


Fig. 2. Scatter chart showing evaluation of SDM procedures in Case A

Four technical managers having good knowledge about enterprise's SDM procedures and their impact on enterprise's work and four SDM users participated in the evaluation. With their help the procedures of their SDM were catalogued. Next, they individually evaluated the SDM procedures related to their work. The results of their combined evaluations are presented in a scatter chart in Figure 2 as described in the methodology section (for the purpose of clearer presentation of evaluations on the scatter chart only the integer parts of cost, goal and product evaluations are shown).

After closely examining their answers considerable differences between the 21 procedures were detected. Based on the discussion of the evaluation results the management focused on improving SDM procedures with the highest impact on cost, goals and products that were not in the fourth quadrant of the scatter chart. To illustrate the differences and the possibilities for different courses of SDM related managerial actions three individual

procedures marked as A, B and C in Figure 2 that management selected for improvement are shown.

Procedure A is describing the use of a database schema generation tool only for generation of a new database schema. In such cases Procedure A is used every time. However, when only modifications to existing database schemas are required these modifications are preformed manually and therefore Procedure A is not used. The procedure was evaluated as having high impact on all three performance criteria since it speeded up the database schema development and improved its quality. Based on the discussion of the individual evaluations of Procedure A with the evaluation participants the management decided to adapt Procedure A to also facilitate its use for modifications of existing database schemas.

Procedure B is describing the use of project management and bug-tracking tool that was newly introduced in the development team at the time. Procedure B enabled the project participants to monitor their current work tasks schedule and to report progress. It also enabled project managers to assign work tasks more efficiently and to detect work tasks that have fallen behind the schedule. The evaluation of procedure showed that it had relatively high opportunities for use, but was not used on all projects. The management expected that the procedure would be used on larger projects while on smaller projects the work tasks would be still managed manually. However, evaluation of actual use showed that despite management expectations it was not used even on most large projects as there was significant resistance among developers. The procedure was evaluated as having high impact on all three performance criteria since it improved productivity of developers and quality of the end product by increasing the efficiency of work task allocation. It also reduced the time that developers spent at work place without actually working through better developer's reports of performed work. Based on the discussion of the individual evaluations of Procedure B with the evaluation participants the management decided to invest in additional training of developers and to make the use of the procedure mandatory for every large project.

Procedure C is describing the preparation and use of formal project development time plan. The formal project development time plan was prepared only for larger projects while on smaller projects the development time plan was presented in an informal manner. Furthermore, in both cases the plan was not kept up to date over the course of the project which resulted in its low use. The procedure was evaluated as having high impact on all three performance criteria since it enabled the management to improve the coordination and organization of the projects. Based on the results of the evaluations the management decided to require project managers to apply Procedure C in regular intervals over the project lifecycle.

As with Procedures A, B and C management used the proposed model to manage other procedures shown in Figure 2. They especially focused on procedures that were identified as having high performance benefits. This supports our second research question (RQ2). They also confirmed that the proposed model provided them with nontivial information about SDM

procedures management (RQ3) which enabled them to significantly improve their SDM related investment and adoption decisions (RQ4).

5.2. Case B

Case B is an enterprise that builds and supports its own ERP solution for small businesses. It uses object-oriented development environment and a self-developed SDM partially grounded on simplified Rational Unified Process [30] and partially on agile methodologies [13]. It comprises 21 procedures (coincidentally the same number as in case A). These procedures are best described as well-defined comprehensive activities that also include short descriptions of artefacts produced by the activities and roles that perform the activities. In addition the SDM prescribes the use of supportive tools for requirements acquisition, task assignment and test automation. Interestingly, the majority of these tools were developed inside the enterprise and are therefore highly adapted to its needs and SDM. The SDM is only used to develop a single product, so there is not much need for SDM tailoring for different projects; however there is need to fulfil individual clients' requirements. It is updated on a regular basis to support new trends in IT. Three technical managers and six SDM users each experienced in the use of different procedures of the enterprise's SDM took part in the evaluation.

Similarly as in case A the procedures were catalogued first, next the evaluation participants answered the given questionnaires and finally discussed the evaluated results. After the discussion of the evaluation results the management focused on improving specific SDM procedures with the highest impact on cost, goals and products. Three individual procedures marked as A, B and C that show the course of managerial actions undertaken are presented in Figure 3. Management expected the greatest enterprise performance benefits from the improvements of these three procedures.

Procedure A is describing the use and development of common component libraries that can be reused in creation of different system functions. There are a lot of opportunities to apply this procedure except when development of custom code is required for the needs of subsystems and prototypes that use specific information technologies (e.g. technologies for integration with specific legacy systems). Although the Procedure A is often used the position of the procedure on the scatter chart shows that there is still a lot of room for improvements. Discussion of the evaluation results showed that developers do not apply all component libraries consistently since they are not familiar with some of them. Furthermore management pointed out that consistent application of all component libraries would improve product stability and maintainability and reduce the cost and time of the development, thereby allowing the company to take on additional projects. Based on these findings the management decided to strengthen the control over the produced code by introducing formal code reviews that would insure that proper component libraries are used.

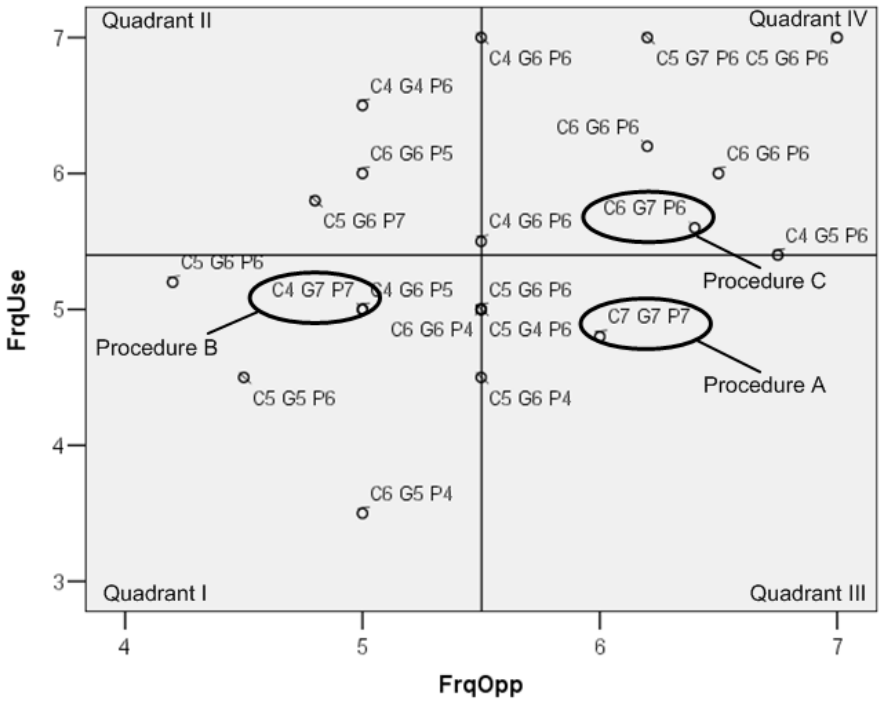


Fig. 3. Scatter chart showing evaluation of SDM procedures in Case B

Procedure B is describing how to document customer requirements for different software system functions in detail. This detailed description of software system functions is used to better understand the requirements and is typically performed only for more complex functions while less complex functions are only briefly described. The evaluation of Procedure B showed that the procedure was too cumbersome to be used for description of simpler functions therefore opportunities for its use were limited to complex procedures. Furthermore the developers did not use the procedure consistently as they perceived it as additional work that was not used by programmers in implementation phase of software development. The procedure was evaluated as having high impact on goals and product since it improved technical quality of the software and its use resulted in software that better matched customer expectations. Based on the discussion of the individual evaluations of Procedure B with the evaluation participants the management decided to simplify the procedure. For instance, they substituted parts of textual description of software functions with graphical representation of user interface. In this manner management hoped to lower the costs and increase the opportunity for use of Procedure B also for less complex software functions. Additionally, management hoped that developers would find such adapted Procedure B more useful.

Procedure C is describing the documentation of source code that betters the understanding, easies maintenance and lowers the cost of upgrades of software. The procedure has many opportunities for use, is used often and importantly impacts enterprise performance. Even though the management was satisfied with the adoption levels of this procedure they additionally wanted to expand its use on parts of the code developed for database management systems. In this way they wanted to make sure that the procedure remains firmly embedded in the fourth quadrant for the foreseeable future due to its importance for enterprise performance.

The management used the proposed model to manage other procedures shown in Figure 3 in a similar way as Procedures A, B and C. They prioritised the procedures that had a high impact on performance (RQ2) and confirmed that the proposed model provided them with nontrivial information about SDM procedures (RQ3) which enabled them to significantly improve their SDM related investment and adoption decisions (RQ4).

5.3. Case C

Case C is a mid-sized enterprise that develops its own pre-packaged business solutions for SMEs. Its speciality is that the management tries to standardize and organize its development process mainly through the use of various commercial development tools. It uses object-oriented design and development tools, an automated testing environment, tools for management of requirements and changes, tools for project management etc. As the enterprise's development process mainly relies on the use of the development tools, the description of the SDM is relatively coarse comparing to the other three cases. It comprises eight SDM procedures that offer general description of different working fields like programming, testing, requirement acquisition etc. The SDM is grounded on Information engineering [35], though it was significantly simplified and partially reorganised to support object-oriented development environment. Three technical managers each of whom was responsible for several of the SDM procedures and five SDM users participated in the evaluation.

As in case A and B the procedures were catalogued, evaluated and results discussed by the evaluation participants. After the discussion of the evaluation results the management focused on improving specific SDM procedures. They especially focused on improving the SDM procedures with the highest impact on costs, since the competition forced them to reduce their development costs. Three individual procedures marked as A, B and C that show the course of managerial actions undertaken are presented in Figure 4. Management expected the greatest enterprise performance benefits from the improvements of these three procedures.

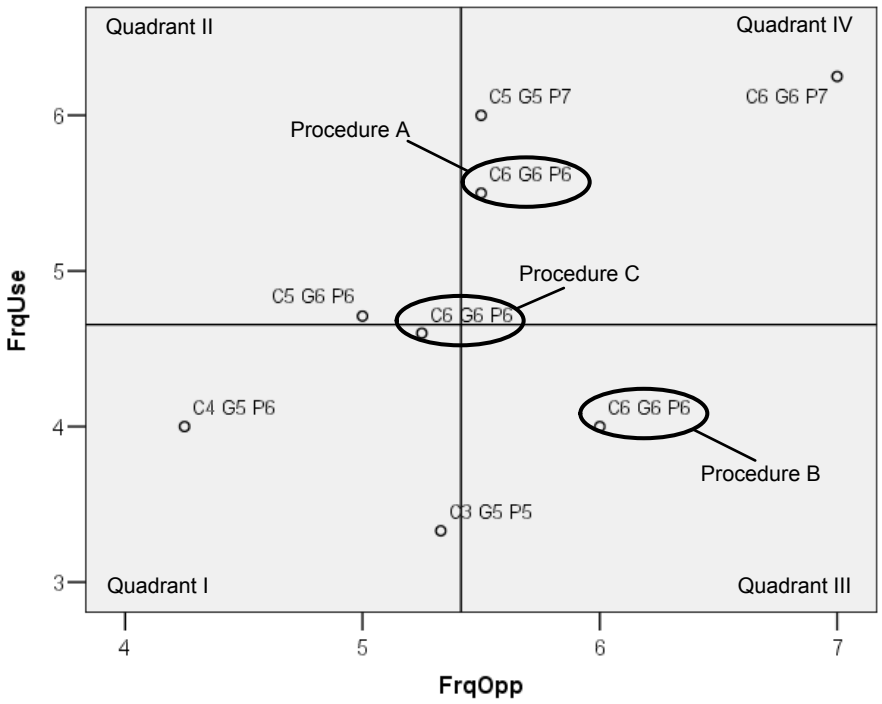


Fig. 4. Scatter chart showing evaluation of SDM procedures in Case C

Procedure A is describing the approach and notation for modelling business processes. The opportunities for use of the procedure are not uncommon however there are still possibilities for improvements. The enterprise policy was to formally model only key and complex business processes while simpler business processes were purposely left out and only addressed through communication between development teams and customers in later phases of development, mainly in requirements acquisition phase and partially in implementation phase. The main motive for such policy was that the management did not want the developers to spend too much effort on business modelling but rather on design and development of programming code. The discussion showed that such policy was not always adequate since it sometimes resulted in development of different programming code for similar activities used in various business processes and functions. To reduce the cost of coding of individual activities they decided to additionally standardise the modelling of certain simpler business processes and activities which facilitates reuse of programming code. In addition the management decided to prescribe a standardised set of architectural patterns [48] for future modelling.

The previously described enterprise policy to formally model only key and complex business processes had also a strong impact on Procedure B.

Procedure B describes the approach and notation for requirements acquisition, however developers did not use it regularly and instead often preferred to use their own simpler ad-hoc approaches that were spontaneously established through long term cooperation with customers. The reason for the limited use of Procedure B was that the development teams interpreted the aforementioned policy as justification for not spending much effort on requirements acquisition for simpler business processes while key and complex business processes were in their opinion already sufficiently described through the use of Procedure A. The discussion showed that such approach caused several problems mainly related to inconsistency of the used notation [50] which resulted in misinterpretation of requirements in the implementation phase as well as difficulties in cost monitoring and project management due to the fact that such ad-hoc requirements acquisition made it hard to clearly specify projects tasks and estimate development teams workloads. To address this situation management decided that requirements should be described formally and that only tasks related directly to formally described requirements can be reported and will count as work hours done by the development team.

Procedure C describes the approach and notation for creation of logical database design that can be used for generation of a database schema in a database management system. Although the procedure assumes the use of a database modelling and generation tool it does not prescribe a specific tool. The enterprise promoted the use of several different database design tools in the past few years, however the use of the tools remained limited and simpler database schemas continued to be created and modified manually. The discussion showed that this was mostly due to general lack of deeper knowledge about any of the available tools as a significant number of potential users i.e. developers did not know how to use the tools to perform more complex tasks. To address this situation the managers decided to prescribe the use of one specific database development tool and organize an in-depth training for the developers.

Other procedures shown in Figure 4 were evaluated in a similar way as Procedures A, B and C. The priority was given to the procedures that had the highest impact on performance (RQ2). Management also confirmed that the proposed model provided them with nontrivial information about SDM procedures (RQ3). This enabled them to significantly improve their SDM related investment and adoption decisions (RQ4).

5.4. Case D

Case D is an enterprise developing financial business software. The enterprise uses a structured SDM [4] that has been considerably modified and improved over the years to support new development approaches and IT. The enterprise has a relatively long history in software development and uses a variety of IT. Their SDM is divided into three branches that are specialized for the needs of different financial software products and used by

different developer groups. Two of the three branches of SDM are intended for projects that are completely run inside the enterprise; the remaining branch is intended for projects in which the implementation is outsourced. The SDM consists of 24 comprehensive procedures that cover different parts of work for each group of developers. Five technical managers and 15 SDM users took part in the evaluation. The selected technical managers had advanced knowledge of different procedures of their SDM and understood the procedure's impact on enterprise performance.

The SDM procedures were catalogued, then evaluated and discussed. Three individual procedures marked as A, B and C seen in Figure 5 were considered as the best candidates for improvement by the management.

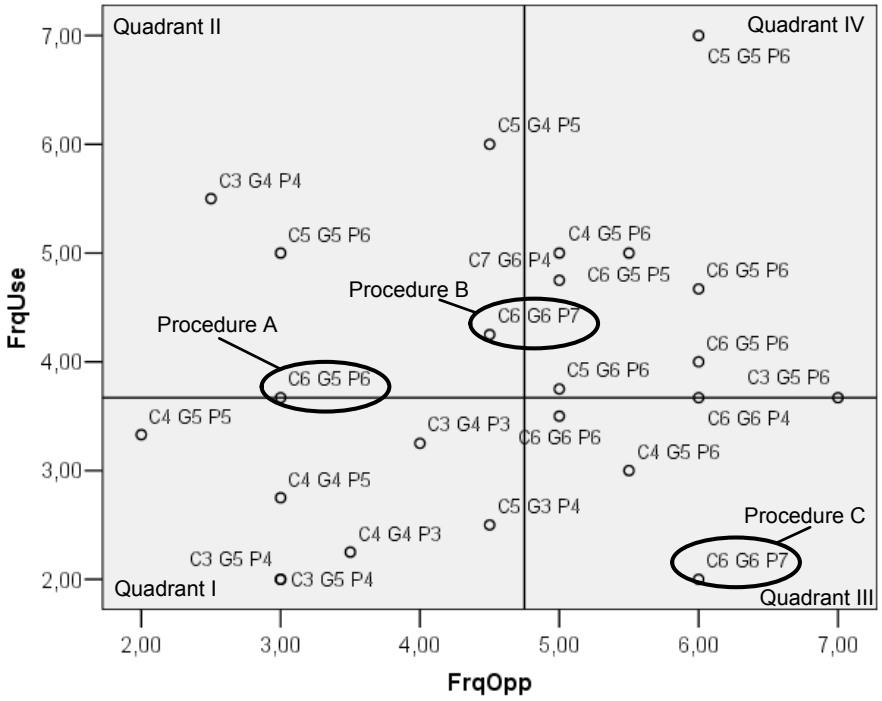


Fig. 5. Scatter chart showing evaluation of SDM procedures in Case D

Procedure A describes the use of a tool for database modelling and schema generation. Although the tool was not limited to modelling and generation of new databases, the Procedure A did not consider this fact and prescribed the use of the tool only for generation of new databases. The opportunities for use of Procedure A were therefore evaluated as quite limited. Furthermore, actual use of the procedure was also low due to the fact that older developers comprising the majority of workforce resisted change. They wanted to keep their "old and proven routines" of manual database

schema development. The management considered this situation very problematic since a fast enforcement of the change could create a hostile work environment. Thus they decided to address the situation as a part of a package of long-term initiatives. These initiatives were designed to lower the resistance to change [55] by lowering the average age of the developers, training the developers in the use of the prescribed procedures and reorganizing the development teams through systematic promotion of the developers open to change. Furthermore they also decided to increase the opportunities for use of Procedure A by adapting it so that it can also be used for modifications of existing database schemas.

Procedure B describes the approach to automated software testing with tool support. The opportunity to use the procedure was limited due to significant number of custom development projects, where automation of testing was considered to be too expensive. The management considered the use of test automation as a cost effective way to improve quality of pre-packaged software. However, management similarly as for Procedure A considered this to be a problematic situation and decided to address it again as part of the long-term initiatives described in the preceding paragraph.

Procedure C describes the approach of software implementation and integration. The opportunity for use was very high since the procedure can be used in every software development project except when software implementation is outsourced. Contrastingly, actual use was very low. The discussion uncovered that the cause of the low use were again the established routines and ad-hoc approaches. These were already detected as the main reason for the resistance to change in the analysis of the preceding procedures A and B. For this reason Procedure C was also addressed through the mentioned long term initiatives.

Although the management was aware of significant resistance to change by the majority of older developers before using the proposed evaluation model they were not aware of the severity of the resistance to change. The realization of the problem's severity motivated them to start the above described long-term initiatives (RQ3). Equally important, the proposed evaluation model enabled them to identify the procedures that can contribute most to enterprise performance if their use and/or opportunities for use are improved (RQ2, RQ3). This enabled them to significantly improve their SDM related investment and adoption decisions (RQ4).

6. Cross-case study results and discussion

In all four cases similar patterns were observed. The proposed model was successfully used by evaluation participants to identify SDM procedures with important enterprise performance benefits (RQ2). Even in case 3 that had the coarsest SDM procedure descriptions the identification of SDM procedures with important enterprise performance benefits was not problematic although it required more discussion. The identified SDM procedures were the ones

that then received most of the management improvement efforts which confirmed that the proposed model provides nontrivial information to the management (RQ3). Additionally, managers in all four enterprises confirmed that SDM related investment and adoption decisions were improved by use of the proposed model (RQ4). The management also confirmed the usefulness of the presentation of the proposed model in a scatter chart linking adoption levels and enterprise performance impacts of individual SDM procedures. The four cases thus clearly show the replication of results consistent to the expectations formulated in our research questions.

The application of the proposed model in practice showed that the evaluation participants on average expressed a positive bias when evaluating their SDM procedures. An exception was the FrqUse in Case D where resistance to change caused a small negative bias. The discussions showed that the evaluation participants perceived themselves as capable software developers that can tackle any procedure prescribed by the technical managers as long as they perceive it as useful. To overcome this problem of possible positive or negative bias the SDM procedures adoption needs to be observed relatively to each other and not on an absolute scale. Therefore the medians were proven to be a better option to group the SDM procedures into the four quadrants than the centres of the adoption scales.

Because of limited resources it was not possible to conduct more case studies. However, according to Yin [60] more than two cases already make a strong argument. The study was limited to software development SMEs which predominantly undertook direct revenue earning projects. Further research should broaden the spectrum of SMEs and include SMEs that also undertake other project types and also test the proposed model in larger enterprises and government institutions.

7. Conclusion and further work

The paper proposes a model for SDM evaluation that concurrently takes into account adoption levels and enterprise performance impacts of SDM procedures. The proposed model allows a software development enterprise to comprehensively evaluate its SDM procedures and to develop appropriate actions for their improvement. The proposed model has been applied in four software development SMEs. These cases showed that the proposed model can significantly improve management understanding of SDM related issues and significantly improve the management of SDM by allowing the managers to focus on the SDM procedures with highest impact on enterprise performance.

The proposed model builds on existing SDM adoption models and augments them by introducing enterprise performance impact of SDM procedures as an important additional dimension. This additional dimension enables managers to focus their actions on improving key SDM procedures and allows them to employ portfolio management practices to management

of SDM procedures. Furthermore, it offers them suggestions on how to improve SDM procedures by dividing SDM procedures into four separate groups (quadrants on a scatter chart) that require different improvement actions.

Further research should focus on the question whether the proposed model can be successfully used in larger enterprises dealing with software development. Moreover, it is possible to expand the set of SDM adoption measures by considering how frequently SDM procedures could be used in an optimally organised software development process. Such measurement could introduce the quality of an organisation as an important additional factor that moderates the interaction between SDM adoption and enterprise performance.

References

1. Agarwal, N., Urvashi, R.: Defining 'success' for software projects: An exploratory revelation. International Journal of Project Management, Vol. 24, No. 4, 358-370. (2006)
2. Ajzen, I.: The Theory of Planned Behavior. Organizational Behavior and Human Decision Processes, Vol. 50, No., 179-211. (1991)
3. Atkinson, R.: Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria. International Journal of Project Management, Vol. 17, No. 6, 337-342. (1999)
4. Avison, D.E., Fitzgerald, G.: Information systems development : methodologies, techniques and tools. 4th ed. McGraw-Hill, 656. (2006)
5. Babič, K.: Managerski odkupi v slovenskih razmerah. Ekonomska fakulteta: Ljubljana. 44. (2003)
6. Baumol, W.J.: Business Behavior, Value and Growth. MacMillan, New York, 164. (1959)
7. Belassi, W., Tukel, O.I.: A new framework for determining critical success/failure factors in projects. International Journal of Project Management, Vol. 14, No. 3, 141-152. (1996)
8. Brezar, M.: Vpliv političnega kadrovanja na poslovanje podjetij v državni lasti: analiza na primeru slovenskih družb Mercator d.d. in Petrol d.d. Ekonomska fakulteta: Ljubljana. 38. (2007)
9. Brinkkemper, S., Lyytinen, K.,Welke, R.J.: Method Engineering – Principles of method construction and tool support. IFIP TC8. Chapman & Hall, Atlanta, USA. (1996)
10. Bygstad, B., Munkvold, B.E.: Exploring the role of informants in interpretive case study research in IS. Journal of Information Technology Vol., Available online from 2010/08/24. (2010)
11. Chandler, A.D.: The Emergence of Managerial Capitalism The Business History Review, Vol. 58, No. 4, 473-503. (1984)
12. Coase, R.H.: The nature of the firm. Economica, Vol. 4, No. 16, 386-405. (1937)
13. Cockburn, A.: Agile software development. Agile software development series. Addison-Wesley, Boston, 304. (2002)
14. Cyert, R.M., March, J.G.: A Behavioral Theory of the Firm Blackwell, Oxford. (1963)

15. Davis, F.A.: Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. MIS Quarterly Vol. 8, No. 3, 318-339. (1989)
16. Dedrick, J., Gurbaxani, V.,Kraemer, K.L.: Information Technology and Economic Performance: A Critical Review of the Empirical Evidence. ACM Computing Surveys (CSUR), Vol. 35, No. 1, 1-28. (2003)
17. Dewett, T., Jones, G.R.: The role of information technology in the organization: a review, model, and assessment. Journal of Management, Vol. 27, No. 3, 313-346. (2001)
18. Downs, G.W., Jr., Mohr, L.B.: Conceptual Issues in the Study of Innovation. Administrative Science Quarterly, Vol. 21, No., 700-714. (1976)
19. Fishbein, M., Ajzen, I.: Belief, attitude, intention, and behavior : an introduction to theory and research. Addison-Wesley Pub. Co., Reading, Mass., xi, 578 p. (1975)
20. Fitzgerald, B.: Am empirical investigation into the adoption of systems development methodologies. Information & Management, Vol. 34, No. 6, 317-328. (1998)
21. Fitzgerald, B.: Systems Development Methodologies: The Problem of Tenses. Information Technology & People, Vol. 13, No. 3, 174-185. (2000)
22. Freeman, R.E.: Strategic management: A stakeholder approach. Pittman Publishing, Marshfield, MA. (1984)
23. Gallivan, M.J.: Organizational adoption and assimilation of complex technological innovations: development and application of a new framework. The DATABASE for Advances in Information Systems, Vol. 32, No. 3, 51-85. (2001)
24. Gallivan, M.J.: The influence of software developers' creative style on their attitudes to and assimilation of a software process innovation. Information & Management, Vol. 40, No. 5, 443-465. (2003)
25. Grant, R.M.: Toward a Knowledge-Based Theory of the Firm. Strategic Management Journal, Vol. 17, No. Special Issue, 109-122. (1996)
26. Green, G.C., Hevner, A.R.,Collins, R.W.: The impacts of quality and productivity perceptions on the use of software process improvement innovations. Information and Software Technology, Vol. 47, No. 8, 543-553. (2005)
27. Huisman, M., Iivari, J.: The individual deployment of systems development methodologies. In: Lecture Notes in Computer Science. Vol. 2348. Springer. 134-150. (2002)
28. Huisman, M., Iivari, J.: The organisational deployment of systems development methodologies. In Advances in Methodologies, Components, and Management. Kluwer, 87-99. (2003)
29. Huisman, M., Iivari, J.: Deployment of systems development methodologies: Perceptual congruence between IS managers and systems developers. Information & Management, Vol. 43, No. 1, 29-49. (2006)
30. IBM: Rational Unified Process v 7.0.1 (IBM Rational Method Composer plugin). IBM Corp. (2006)
31. Jugdev, K., Müller, R.: A retrospective look at our evolving understanding of project success. Project Management Journal, Vol. 36, No. 4, 19-31. (2005)
32. Karlsson, F., Agerfalk, P.: Exploring agile values in method configuration. European Journal of Information Systems, Vol. 18, No. 4, 300-316. (2009)
33. Khalifa, M., Verner, J.M.: Drivers for software development method usage. IEEE Transactions on Engineering Management, Vol. 47, No. 3, 360-369. (2000)
34. Lavbič, D., Lajovic, I.,Krisper, M.: Facilitating information system development with Panoramic view on data. Computer Science and Information Systems, Vol. 7, No. 4, 737-767. (2010)

35. Martin, J.: Information Engineering, Book I: Introduction. Prentice Hall. (1989)
36. Milis, K., Vanhoof, K.: Analysing success criteria for ICT projects. Int. J. Nuclear Knowledge Management, Vol. 2, No. 4. (2007)
37. Moore, G., Benbasat, I.: Development of an Instrument to Measure the Perceptions of Adopting an Information Technology Innovation. Information Systems Research Vol. 2, No. 3, 192-222. (1991)
38. Perrow, C.: The Analysis of Goals in Complex Organisations. American Sociological Review, Vol. 26, No. 6, 854-866. (1961)
39. Pluta, J.: Developing Web 2.0 Applications with EGL for IBM i. MC Press, Lewisville, 200. (2009)
40. Porter, M.E., Kramer, M.R.: Strategy and Society: The Link Between Competitive Advantage and Corporate Social Responsibility. Harvard Business Review, Vol. 2006, No. December, 78-92. (2006)
41. Prašnikar, J., Svejnar, J.: Investment, Wages and Ownership During the Transition to a Market Economy: Evidence from Slovenian Firms, in William Davidson Institute Working Papers Series. (1998)
42. Putterman, L., Kroszner, R.S.: The Economic Nature of the Firm. Cambridge University Press, Cambridge, 400 (1996)
43. Ralyte, J., Deneckere, R.,Rolland, C.: Towards a Generic Model for Situational Method Engineering. In 15th International Conference on Advanced Information Systems Engineering. (2003)
44. Riemenschneider, C.K., Hardgrave, B.C.,Davis, F.D.: Explaining software developer acceptance of methodologies: A comparison of five theoretical models. IEEE Transactions on Software Engineering, Vol. 28, No. 12, 1135-1145. (2002)
45. Rogers, E.M.: Diffusion of innovations. 5th ed. Free Press, New York, xxi, 551 p. (2003)
46. Simon, H.A.: Administrative Behavior: A Study of Decision-Making Processes in Administrative Organizations. The Free Press, New York. (1947)
47. Simon, H.A.: Rational choice and the structure of the environment. Psychological Review, Vol. 63, No. 2, 129-138. (1956)
48. Šaša, A., Krisper, M.: Enterprise architecture patterns for business process support analysis. Journal of Systems and Software, Vol. 84, No. 9, 1480-1506. (2011)
49. Trkman, M., Trkman, P.: A wiki as intranet: a critical analysis using the Delone and McLean model. Online Information Review, Vol. 33, No. 6, 1087-1102. (2009)
50. Vasilecas, O., Dubauskaitė, R.,Rupnik, R.: Consistency checking of UML business mode. Technological and Economic Development of Economy, Vol. 17, No. 1, 133-150. (2011)
51. Vavpotic, D., Bajec, M.: An approach for concurrent evaluation of technical and social aspects of software development methodologies. Information and Software Technology, Vol. 51, No. 2, 528-545. (2009)
52. Vavpotic, D., Vasilecas, O.: An Approach for Assessment of Software Development Methodologies Suitability. Elektron. Elektrotech., Vol. 8, No. 114, 107-110. (2011)
53. Veblen, T.: The Theory of the Business Enterprise. Transaction Books, New Brunswick, New Jersey. (1904)
54. Venkatesh, V., Davis, F.D.: A theoretical extension of the Technology Acceptance Model: Four longitudinal field studies. Management Science, Vol. 46, No. 2, 186-204. (2000)

55. Vrhovec, S., Rupnik, R.: A model for resistance management in IT projects and programs. Electrotechnical Review, Vol. 78, No. 1-2, 24-37. (2011)
56. Wateridge, J.: IT projects: a basis fo success. International Journal of Project Management, Vol. 13, No. 3, 169-172. (1995)
57. White, D., Fortune, J.: Current practice in project management - an empirical study. International Journal of Information Management, Vol. 20, No. 1, 1-11. (2002)
58. Williamson, O.E.: Markets and Hierarchies: Analysis and Antitrust Implications. The Free Press, New York. (1975)
59. Wynekoop, J.L., Russo, N.L.: Systems-Development Methodologies - Unanswered Questions. Journal of Information Technology, Vol. 10, No. 2, 65-73. (1995)
60. Yin, R.K.: Case study research : design and methods. 4th ed. Sage Publications, Thousand Oaks, Calif., 240. (2008)

Damjan Vavpotič. Dr. Damjan Vavpotič is a senior-lecturer of Information Systems and Information Systems Development in the Faculty of Computer and Information Science at the University of Ljubljana. He received his doctorate in Information Systems Engineering from University of Ljubljana, Slovenia. His research interests include information systems development methodologies, methodology adoption and evaluation, and agile methodologies. His research has appeared in journals such as Information and Software Technology, Informatica, Applied Informatics, and Nurse Education Today, and in proceedings of many international conferences. He participated in many projects dealing with evaluation and improvement of ISDMs, IS strategic planning, and IS development.

Tomaž Hovelja. Dr. Tomaž Hovelja received bachelor's degree, master's degree and his PhD in Business Administration from the Economic Faculty at the University of Ljubljana. He is employed as an assistant at the Faculty of Computer and Information Science at the University of Ljubljana. His research areas are social, economic and organizational factors of IT deployment in enterprises and IT projects success criteria. His research has appeared in journals such as Economic Annals, Economic and Business Review for Central and South-Eastern Europe, Management - Journal of Contemporary Management Issues and others.

Received: May 3, 2011; Accepted: November 28, 2011.

DOI 10.2298/CSIS101103040I

A Framework for Developing and Implementing the Enterprise Technical Architecture

Tiko Iyamu

Department of Informatics, Tshwane University of Technology
Pretoria, South Africa
iyamut@tut.ac.za

Abstract. Organizations build, buy and reuse different types of technology with the intention to addressing their organizational needs, challenges and for competitive advantage. Unfortunately, the means is not the end. Instead, in some ways, it leads to complications and complexities, and more importantly, consumes more resources. Some organizations have adopted the technical architecture approach to address the challenges posed by technology deployment. The technical architecture is intended to address aspects, from strategic planning to implementation of technology infrastructures. This is to consistently effect significant technological change within the environment. The technical architecture approach facilitates and enables prioritization of analysis, development and implementation, which are based on value added business requirements and vision. It therefore allows the organization to proceed at its own pace while progressing at the same time. The paper presents model which reflects the consistent approach that adaptive enterprises could employ to build, maintain, and apply technical architecture in the computing environment. The model emphasizes a holistic approach to technical architecture deployment in the organization.

Keywords: Enterprise Architecture, Technical Architecture, Operating model, Development, Implementation.

1. Introduction

In the last two decades, effort to improve on information technology (IT) strategies and operations has increased. This includes bridging the gap between IT and the business, pragmatics assessment and accessibility of technologies in addressing needs, semiotics of technology deployment, and how the technology strategy impact the vision of the organization. Specifically, the EA could be deployed to manage strategies, processes, and resources systematically, focusing on information technology and systems [1]. Some studies including Kilpeläinen [2] argue that the role of technical solutions is to support business objectives and operations. However, there has been improvement in some areas such as systems development, project

management and technology deployment [3]. Despite these improvements and efforts, organizations still find it difficult to realize and comprehend returns on their IT investments. According to Brynjolfsson [4], IT has made the MIS manager's job of justifying investments particularly difficult. He further argued that the disappointment in IT has been chronicled in articles disclosing broad negative correlations with economywide productivity and information worker productivity. Unfortunately, this makes it seem as though none of these efforts have been able to completely resolve the challenges of meeting the organization"s needs timely, and ensuring that value is achieved from the investments on IT infrastructures. These challenges could be attributed to complexities of deployment, duplications of technologies and how they intend to enable business needs [5]. As a result, many organizations have sought solutions through the deployment of technical architecture. What is even more important in achieving the technical architecture objectives is the semiotics, the understanding between human and technical actors, in the deployment of technologies. The semiotics includes the process, and how the requirements are derived in the development and implementation of the technical architecture. According to Ross et al [6], *"operating model is the necessary level of business process integration and standardization for delivery goods and services".* Otherwise, the challenges could possibly be increased and prohibitive. Many organizations have developed technical architecture, but were never implemented [7].

Technical architecture is a prevailing paradigm, which questions the goals, scope, processes and roles that are considered on technological deployment and introduction of new technology infrastructure potentialities. In the last decade, there has been an increasing interest in technical and other architectures in organisations [8]. Patel [9] argues that there are many challenges which need to be addressed by IT functions; this includes planning, rapid business and environmental change. In an attempt to answer the questions (how IT infrastructure can be used to achieve business needs, as well as gain return on investment?) which the technical architecture poses, the business and IT units of the organization engages in an intensive *learning* experience. The former develops a thorough understanding of a model which reflects on current practices and acquire understanding of the potential of technical architecture to transform how and what work is done. Thus, standard and principles are employed. Iyamu [7] defines principles as guiding statements of positions that communicate fundamental elements, truths, rules, or qualities that must be exhibited by the organization during implementation.

The technical architecture is intended to provide the means to maintain an adaptive infrastructure through sets of principles, standards, configurations, process and governance. It ensures that existing and new information technologies and systems are maintained, selected, respectively to achieve the strategic goals of the organization [10]. Patel [9] further posits that IT governance is affected by an organization"s unique culture and working practice, and should reflect its own goals and ambitions. Hafner, et al. [11] describe IT architecture as the domain of architectures which represents an aggregate, enterprise wide model of hardware and communications

components as well as dependencies between the technology artifacts. The architecture also drives the introduction of new technology by focusing on function and scalability, compatibility and interoperability.

The paper introduces and puts into context the modus operandi of the technical architecture as could be deployed within an organization as its architecture framework. The experience, on which this work is based, comes from the author"s work as an architect, and from working closely with other architects who have designed systems for large corporate and government departments. This will give clarity and understanding of some of the words, elements and components that are involved in the development and implementation of technical architecture. A four phase approach is adopted. Like many other frameworks, it is not a definitive guide to technical architecture modeling. It could be customized within environmental and technological context over time. The framework begins by presenting overview of the technical architecture deployment including a diagrammatical representation (Figure 1).

The remainder of the paper is structured into six sections in accordance to the phases of technical architecture deployed. The first section provides overview on the deployment of technical architecture. The second section highlights the primary objectives of the architecture. The third section covers the business strategy. The fourth and fifty sections discuss how technical architecture is developed and implemented, respectively. The paper is concluded with highlights of the contribution of the study.

2. OVERVIEW: Technical Architecture Deployment

This paper presents a methodological approach (model) for the deployment of technical architecture. As depicted in Figure 1 below. The deployment is a four-phase approach. The model starts by identifying the business-driven requirements for the technical architecture. This is the importance of the relationship between the business and IT. According to Luftman et al [12] both IT and business need to listen to one another, communicate effectively, and learn to leverage IT resources to build competitive advantage. Based on the business vision and requirements, technical architecture focuses on deriving high-level requirements to give direction and priority to its domains [13], [14]. While there is a suggested dependency among the requirements, not all requirements are necessarily – identified in earlier paths through the model. For instance, while focusing on technical architecture, an architecture team could concentrate on deriving technical architecture requirements directly from environmental or technology trend relating to business strategy. Others could find the need to develop a more robust and intuitive set of technical architecture requirements after first deriving business requirements. It is important to remember and understand that requirements directs the architecture in what they are to provide in support of the business, not how they will provide it [15].

Figure 1 illustrates the four logical phases (Objectives, Business Strategy, Development and Implementation) involved in the deployment of technical architecture. Change occurs through the development and implementation of the technical architecture. Iyamu [16] argued that the architecture is often deployed with the intention to *manage technology, change from system to system, implement new technology, maintain compatibility with existing technologies, and change from one business process to another.* In the context of this paper, technical architecture is defined as a logically consistent set of principles, standards and models that are derived from business strategy (business vision, requirements and contextualization) in order to guide the engineering of the organization's information systems and technology infrastructure across the various domain architectures [17].

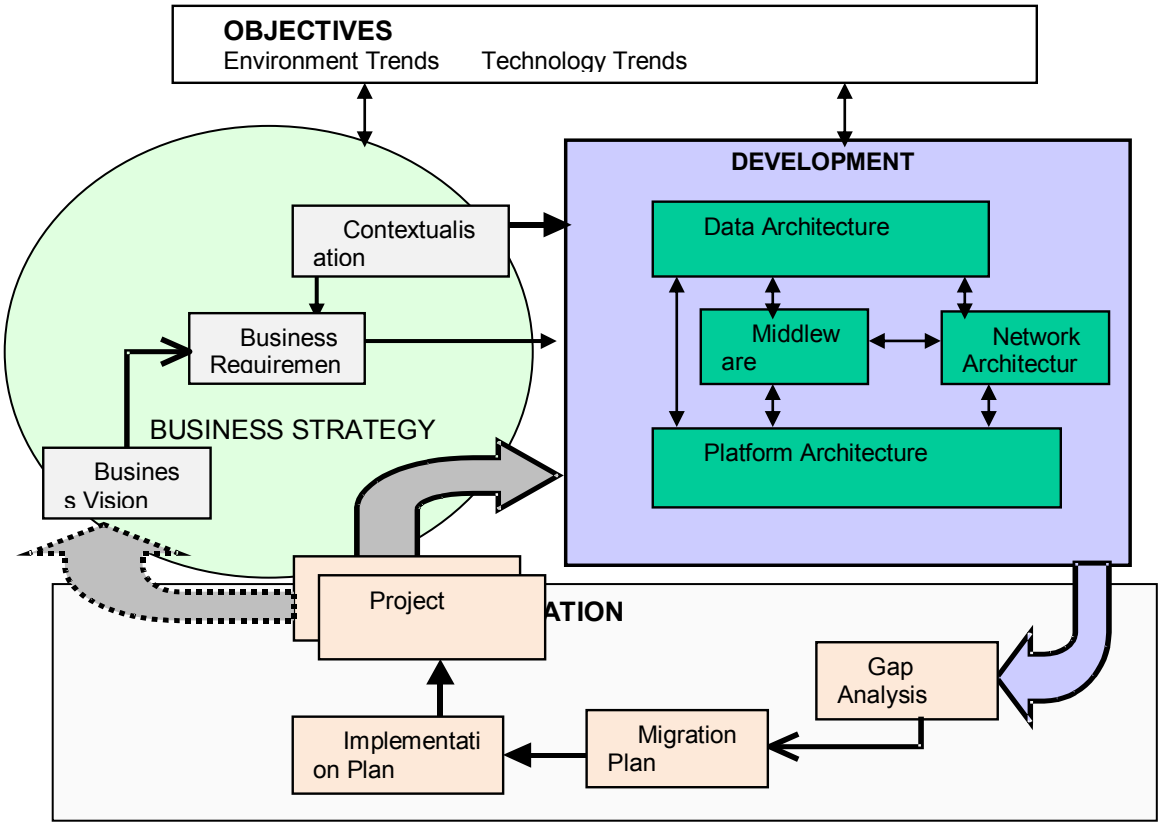


Fig 1. Technical Architecture Deployment

3. Phase One: Objectives

The objective of the technical architecture framework is to provide a baseline of the current and future states of an organization"s technology environment. One of the challenges in managing change toward the future is gaining an accurate view of the current state. This includes facilitating assessment of the impact of technology change on the current organizational environment and the conception of strategic alternatives for consideration. This led to the main aim of the paper, which provide guide for technical architecture deployment.

Conceptually, objectives of the technical architecture are to:

i. Effectively remove hardware obsolescence or vendor dependency as a requirement;

ii. Re-engineer technology artefacts in the organisation that deploys it;

iii. Enable information systems;

iv. Periodically review the systems in support of the organisation" needs; and

v. Ensure that the rapidly changing external and environmental trends are enforced to significantly change the business and technical environments within organisations.

The technical architecture requires ongoing evaluation and iterative processes to protect the major investment in information technology and systems by keeping them current with the changing environment. The next section explains the development of technical architecture, which begins with the requirements.

4. Phase Two: Requirements

The business units or divisions often have different requirements, often far apart. As a result poses challenge in attempt to map the requirements for the organisation"s common goal. Also, some of the stakeholders are actors in many units or divisions, thereby, causing conflict of interests. On another hand, technical actors also pose challenge of shared services and infrastructure.

As depicted in Figure 1, the second stage, business strategy is the primary driver for the development and implementation of technical architecture. It consists of the business vision, business requirements and contextualization. This phase is primarily influenced by factors which include organisational processes and activities, and environmental trends [18]. The trends relate to the business of the organization as well as relevant technologies that enables them.

The outcome of phase 1 (Visioning and Business strategy) as described above is articulated and translated to address business and technological

challenges in the organization. At this phase, the business strategy layer, the technical architecture is developed to the engineering of information systems and technology infrastructure in the organization. This is to enable and support processes, business logic and activities.

The components of the business strategy phase are discussed as follows:

4.1. Business Vision

As illustrated in Figure 1, this business vision is the first component of the second phase in the deployment and implementation of the technical architecture. The primarily purpose is to provide a clear vision of "business futures" by capturing the most important enterprise business strategies being pursued at the time [19]. It focuses mainly on strategic thinking about the future.

For the purpose of business visioning, the project team gathers strategic planning documentation from the business and extracts the relevant business drivers as well as high level information and application needs. Environmental trends are considered in the process and a business vision document is prepared and verified with senior management and other stakeholders.

4.2. Business Strategy

The business requirements are derived from business vision and expressed as functional statements, which give direction and priority to, among other activities, technical architecture. Aerts et al. [20] argued that a business strategy targets the goals of business processes, which the architecture is purposely deployed to addresses their realization.

Generically, the business requirements include:

- Enabling the organization to implement its strategies over time with minimal impact to ongoing service delivery and processes.
- Enabling business scalability - be able to respond and operate at the same rates in which business process and activities occur.
- Enabling the organization to compete in the economy according to technology and environmental trends.
- Delivering flexibly packaged and priced services through multiple channels at multiple geographic locations.
- Enabling governance in the computing environment.

4.3. Contextualization

At this stage of the deployment, articulation of the organization"s system into technical architecture context is performed. The context consists of the business vision and requirements of the organization, as captured at the time. The requirements are extracted into set of needs which are then used to

develop the subsequent architectural products, a set of blueprints and views of the organization. Each view is expressed in terms of components, connections and constraints, which are governed by architectural model. A key feature of the approach is the conceptual mechanism that provides traceability between views.

Once the gathering of information, requirements from the business strategy is completed, the architects begin the development of the technical architecture.

5. Phase Three: Development

The development of technical architecture is in four stages, including requirements; classification of domains; documentation; and governance principles. Technical architecture is developed for purpose, which is relative to organizational strategy. The development of the technical architecture includes four main sequential steps:

i. Technical requirements;
ii. Definition of the domains;
iii. Documentation of technologies into current-to-strategic forms; and
iv. Formulation of governance principles.

5.1. Technical Architecture Requirement

Technical architecture requirements are derived from the architectural requirements contained in the business strategy stage as they relate to the technical architecture. The requirements for technical architecture are derived from the overall business strategy as identified in the business vision, requirements and contextualized. They describe the basic functions required of the technical architecture in enabling the business drivers specified by the organization in the business strategy.

The development stage describes the components of technical architecture and their relationships. It also gives a view of the sequence involved in developing technical architecture. The above introductory paragraphs demonstrate how technical architecture could be developed and implemented, based on the empirical evidence that comes from experience. This section focuses on development, beginning with the definition.

Different definitions of the technical architecture such as Tan and Gan [21] do exist. According to Rosenberg and Stephens [22], technical architecture generally describes and defines the system and software of *structure*. In this study, the technical architecture is defined as a logically consistent set of principles, standards and models that:

i. Are derived from business requirements;
ii. Guide the engineering of the organisation"s information systems and technology infrastructure across the various domain architectures;

iii. Are understood and supported by senior management and lines of business of the organisation;
iv. Take into account the "full context" in which the domains of technical architecture will be applied; and
v. Enable rapid change in the organisation"s business processes and the applications that enable them.

In the development, implementation and practice of technical architecture, definition is key, it is importantly emphasized, so to understand the different entities, components, scope and boundaries of the domains of the architecture. The definition, guides the integrations, collaborations and the changes the technical architecture engineer through its development. The next subsection addresses the domain architecture including its definitions.

5.2. Technical Architecture Domains

Each domain of technical architecture consists of both technical and non-technical (such as process and people) factors. Technical architecture enables change by bridging the gap between strategic planning and implementation efforts through a strategy process that is holistic in scope. Technical architecture comprised of different types of domains, Data, Middleware, Network, Platform and Distributed architectures. Each domain has unique and specific deliverables, analysis methods, processes and participants. Table 1 provides definition (domain, description and category) of the domains. The domain architectures are created to provide principles and standards for using technologies as they are related, and to enable specific business objectives.

As expressed in the abstract and introduction sections, the technical architecture provides the means to maintain an adaptive infrastructure through sets of principles, standards, configurations, process and governance. It also drives the introduction of new technology by focusing on function and scalability, compatibility and interoperability to fulfill business and technical needs. Individual templates (Tables 1, 2, 3, and 4) are used to achieve the objectives of the technical architecture within the context of the organization needs. The templates could be manual or automated; they are used for information capturing, validation and processing of the architecture activities and deliverables.

Domain-level technical architectures contain the prescriptive elements of the technical architecture that guide the IT engineering activities within the organization. This includes the analysis, design and implementation and operations management. They provide the organization with a means to categorize related technologies for the purpose of identifying reusable technology and they consist of sets of principles and standards (industry, product, and configuration) that govern the selection and use of related technologies in specifically defined logical domains. The development phase, domains of technical architecture (Data, Middleware, Network, Platform and Distributed) are inter-dependent as they each evolve iteratively. They are

defined as in Table 1. the last column, “Process/Procedural” is guided by the same pattern.

Table 1. Technical architecture domains

Technology Domains	Description	Technology Categories	Process/Procedural
Data Architecture	Defines the mechanics for managing, securing and maintaining the integrity of the organization"s significant logical entities. These entities are recorded and accounted for in the business information environment. The architecture provides standards for accessing data, as well as business objects if appropriate.	Data and Object Repositories, Data Encyclopedia, Data Modeling, Replication and Administration Tools, Object-Oriented Databases.	Based on the business requirements, principles are formulated, within which data is classified and managed; technologies selected and deployed.
Middleware Architecture	Defines the components that create an integration environment between user workstations and legacy and server environments to improve the overall usability of the distributed infrastructure. It creates uniform mechanisms for application integration independent of network and platform technologies.	Remote Procedure Calls (RPC), Messaging-Oriented Middleware (MOM), Object Request Brokers (ORBs), Transaction Processing (TP) Database (DB) Gateways.	The business and technical requirements dictates the principles and standards in the selection and deployment of technology.

Technology Domains	Description	Technology Categories	Process/Procedural
Network Architecture	Provides the communication infrastructure for the distributed computing environment. It consists of logical elements, physical hardware components, carrier services, and protocols.	Network Hardware, Network Operating Systems, Security, Carrier and Internet Services.	The business and technical requirements guides the design, and management of the network.
Platform Architecture .	Defines the components of technology infrastructure, including the client and server hardware platforms; the operating systems executing on those platforms; and the database environments and interfaces supported.	Hardware (Workstations, Servers), Operating Systems, Database Management systems.	Based on the business and technology requirements, principles and standards are formulated, within which technologies are selected, deployed and managed.
Distributed Architecture	Defines how the hardware and software components of the environment will be managed. It focuses on issues of configuration management, fault detection and isolation, testing, performance measurement, problem reporting, and software upgrades and controls.	Network Systems, Configuration, Storage Management and, Security, Performance Management, Capacity Planning.	Based on the business and technology requirements, principles and standards are formulated, within which technologies are selected, deployed and managed.

Actors‟ enrolment in the different domains is based on skill and area of specialization. However, personal interest is a strong influencing factor as well. There are three main steps involved in creating the technical domain architectures, these are:

- Creating domain architecture that represents the logical technical domains and their relationships to each other, referring to existing frameworks and modifying them accordingly, rather than creating one from scratch.
- Defining domain-level principles, categorise technologies into appropriate domains, then document technology, product and configuration standards as appropriate groups complete technology research, product evaluations, and configuration designs.
- Relate standards to appropriate and supported infrastructure patterns within the organisation.

Each of the technical architecture domains provides a unique capability view of the computing environment. The Data and Network domains provide the tools, models, techniques, and participants to manage the impact of change on business processes and partners; Middleware and Distributed domains enable the management of change on business system logics and applications; while the Platform domain enables the management of change on technology infrastructure.

The domain architectures represented in Table 1 have been identified as the most appropriate groupings of technologies that are required to enable and support the business drivers. These domain architectures evolve in an iterative manner as both technical and business requirements dictate. Primarily, the role of each of the domain architectures is intended to organize technologies while their usage rules to assist architects in identifying common uses of technologies, and eliminate redundancy as much as possible.

5.3. Classification

All technologies are grouped into domains (Table 1), which defines their strategic or non-strategic terms. A consensus among the key stakeholders is reached on the following: employees‟ involvement; authenticity of technologies; where the technologies resides; in what forms and their use. Technical requirements are then defined in a technical architecture product catalogue, which contains information about the products. The rules pertaining to the individual technologies are reflected in a Table 2, per domain architecture. Table 2 is an example of a populated template of the technical architecture.

The columns may be defined as follows:

A. **Technology category** refers to each of the broad logical groupings of technologies within the domain *e.g. data warehouse.*
B. **Technology** refers to the actual technology involved *e.g. ETL (extract, transform, load) tools.*
C. **Standards** are the (local and international) within which the products complies *e.g. “ODBC”.*
D. **Products** are the named vendor products (hardware, software, etc.) *e.g. Oracle Warehouse Builder.*

E. **Current / Future**) indicates whether the product is currently in use or planned for use in the future.

F. **Architectural Status** is the formal status given to each product as agreed by the Architects and directly impacted stakeholder, and it is indicated as follows:

S - Strategic *(all new development will utilise this product according to the configuration standards, and existing solutions are to migrate to or be replaced by this product where feasible)*;

M - Maintain *(do not proliferate, maintain for as long as the product is required - there should be a replacement strategy in the 3 to 5 year timeframe)*;

O - Outdated *(must be upgraded or replaced)* or **Obsolete** *(must be removed)*;

N - Not supported or **not approved** *(any products evaluated and rejected)*;

T - Tactical *(interim, short-term solution for up to 3 years, to be replaced by a strategic choice)*;

E - Evaluating *(being researched, in proof of concept phase and or being piloted - if successful, will be given tactical or strategic status - if not successful, will be given "not approved" status with supporting documentation)*.

Configuration standards are used to describe how the product will be used in the organisation. Software configuration standards include the release or version number to be used, installation options and configuration settings, upgrade paths and maintenance procedures. Hardware configurations should include the model name and number, configuration settings, installation procedures and any related peripheral standards. References to diagrams used to position the product, procedures, and or guidelines may be used where necessary. The positions are indicated as follows: **S**ingle user, **G**roup users, **D**epartmental, **F**ew Department and Entire **O**rganisation.

The domains of the technical architecture are standardised to ensure uniformity, assessment, evaluation and reduce complexity.

The key for the content of Table 2 above are as follows:

- **A** – Consists of one or more **B**s;
- **B** – Each **B** could have more than one **C**; and **D**, depend on the organization;
- **E** – Its either a **C** or an **F**;
- If **E** is equal to **C**, then **F** is equal to **S/N/E** and
- If **E** is equal to **F**, then **F** is equal to **S/M/O/N/T**

The last stage of the technical architecture development is the formulation of the principles. This is covered in the following section.

Table 2: Technical architectural grouping.

Technology Category (**A**)	Technology (**B**)	Standards (**C**)	Products (**D**)	**C**urrent/ **F**uture (**E**)	Architectural Status (**F**)	Configuration (**G**)
A1	B1	C1	D1	C	S	O
		C2	D2	F	E	S
	B2	C3	D3	F	S	D
		C4	D4	C	M	G
		C5	D5	C	N	F

5.4. Domain Principles

Principles, as defined by Patel [9] as guiding statements of position that communicate fundamental elements, truths, rules, or qualities that must be exhibited by the organization. Principles are formulated from the business strategy as they pertain to the technical architecture and are further broken down into rationale, process and guidelines. This applies to all the domains of technical architecture.

The primary purpose of principles is to enable the organization to take an incremental and iterative approach of transitioning to formal modeling, while allowing it to influence decision making immediately and consistently. In the technical architecture context, principles are used as evaluation criteria in the absence of detailed models that direct decision making more discretely and comprehensively. For example, one type of architecture model is a technology domain configuration standard that details technology products and the way they are configured together to deliver a reusable building block of technical infrastructure (e.g., an application server). In the absence of a defined configuration standard, an application development team's technical design for an application should be evaluated for its consistency with the principles dealing with applications, information, and technical infrastructure.

Table 3: Principle formulation

Principle	Rationale	Process	Guideline
Articulate and give a name which reflects the intention and essence of the objective of the organization.	Highlights potential benefits in adhering to the principle.	Formulated procedure to assists in achieving the rationale as set-out by the organization.	Specify rules and regulations that must be followed in adhering to the process.

The principles lead to policies and procedures and are used to give guidance to the designers, developers and implementers of the technical architecture (Data, Middleware, Network, Platform and Distributed). The rationale for the principle is documented along with any process or procedure and guideline, which could be used when applying the principle.

The principles impact the selection, design, and implementation of software packages, application components, infrastructure deployment and business objects.

6. Phase Four: Implementation

The implementation phase consists of four stages: Gap analysis; Migration planning; Implementation planning; and Project – each of these stages is briefly described below. The technical architecture is expected to enable rapid technology-related change in the organization"s business processes and the applications that enable them. The technical architecture is dependent on:

- The business vision of the organisation, which captures the most important business strategies being pursued at the time;
- The organisational requirements which consists of high-level architectural requirements, derived from business strategies, to give direction and priority; and
- The architecture which contains sets of principles, derived mainly from best practices and trends that are relative to, and consistent across each domain architectures area within the business strategy of the organisation.

6.1. Gap Analysis

The final phase of the deployment consists of conducting gap analyses across the domain architecture areas to determine corrective action, develop prioritized migration plans and finally draw up implementation plans. Carrying out the projects in the implementation plans change the organization from the current state to future state as defined during the project.

The purpose of the gap analysis is to assess the current state of the technical architecture against the desired state as reflected by the drivers, which is covered during phase two (Business Strategy). This assessment is an iterative and ongoing process and is reflected by a checklist (objectives and business strategy) and an accompanying action plan. These assessments are stored in the technical architecture repository and made accessible to the architects and other stakeholders.

Table 4: Implementation.

Requirement (Future State)	Action	Deliverable	Responsibility	Status	Target Date
Principles					
The entity that defines the rationale and motivation for the task. This entity provides guidance for the scope.	Defines the individual tasks including the deliverables.	Tasks to be carried out.	Who carries out the tasks – architect, business units, etc.	Have options: (1) Not started (2) In Progress (3) Completed	Date agreed to by the primary stakeholders including the architect.

6.2. Migration Planning

Positioning strategy and movement from one architectural phase to another is a very complex issue. It is much more complex than simply bringing in a new vendor or new technology.

The current trends in technology architectural direction are toward higher levels of connectivity and functionality. True, the current movement in the industry is toward more "open" strategies, but if this ever occurs in reality, the building block approach to implementation will be a viable strategy.

Certainly, one of the key decision processes involved with architectural planning is the need to have a future target. Shorter-term goals can then be defined as stepping-stones to the strategic goal. Of course the problem here is that historically, information technologists have not been all that accurate in predicting product directions and timing. For example, who had in the 80s predicted the impact that PCs have on the industry, today? Or who would have predicted the explosion of the Internet?

6.3. Implementation Planning

Once the current inventory is defined, the first step in planning is to assess the strategic and technical fit of the current technologies. Strategic fit is assessed based on the technology"s contribution to business value, as defined by the business strategy. Technical fit is assessed based on the technology"s adherence to the technical architecture design principles and technology standards. This assessment enables the applications to be placed in the categories as shown in Table 4.

6.4. Project

The development and implementation project passes through this cycle multiple times. There are continuous interactions between the project implementers and the architecture.

Each technical architect project is required to model the technology environment, including infrastructure configuration standards and guidelines for using standard products and configurations. The models provide both views of the recommended technology and the bases for assessing the impact of new and replacement of technologies within the context of the whole enterprise-wide technology infrastructure (rather than just within the context of the specific technology being considered).

The technical architecture is not a project, but an iterative process. The process phase ensures the operationalisation of technologies in an architected approach.

7. Conclusion

The modeling provides guidance concerning the information technology assets to knowledge workers, information processors, IT application developers, infrastructure managers, and executives. The model helps organizations to explore the factors leading to the success or failure of technical architecture in their computing environment. The paper presents a new paradigm for building technical architecture that improves the effectiveness of functional operations to include their efficiency and use of technology throughout the organization.

The contributions of the paper are from two main perspectives, methodological and practicality. The methodological contribution of the paper is through the perspective in which we gain better understanding of the procedural deployment and use of technical architecture in the computing environment. The other main contribution is the practicality, which constitutes a learning curve for information technology architects in the development and implementation of technical architecture. Based on the practicality, it is an aggregation of ideas and experience to which many architects would subscribe to. Also, it is expected to benefit the computing industry, IS researchers on the capabilities of the technical architecture involving change and through it, contributes to the body of knowledge in this sub-field.

An area recommended for future research, which this article did not cover, is the social context in the deployment (development and implementation) of the technical architecture in the organization. It would be interesting to both academics and profession to understand the interplay between various actors during the deployment of technical architecture in the organisations.

Another future research area is an understanding of the roles and impacts of both organisational structure and structures, in the deployment of technical architecture. Structures, as defined by structuration theory, according to [23],

the word „structure" must not be confused with its obvious connotation of organisational hierarchy in the English language. Giddens [24] defined structure as rules and resources in Structuration Theory.

References

1. Kang, D, Lee, J. and Kim, k.: Alignment of Business Enterprise Architectures using fact-based ontologies. Expert Systems with Applications, Vol. 37, No. 2010, 3274–3283. (2010).
2. Kilpeläinen, T.: Business Information Driven Approach for EA Development in Practice. In the Proceedings of the 18th Australasian Conference on Information Systems. Australia, 5-7. (2007).
3. Zachman, J.: Enterprise Architecture: The View Beyond 2000. In the Proceedings of the 7th International Users Group Conference on Warehouse Repository Architecture Development, Technology Transfer Institute. (1996).
4. Brynjolfsson, E.: The productivity paradox of information technology, Communications of the ACM, Vol. 36, No. 12, 67 – 77. (1993).
5. Armour, F., Kaisler, S., Bitner, J.: Enterprise Architecture: Challenges and Implementations. In Procedings of the 40th Annual Hawaii International Conference on System Sciences, USA, Hawaii, 217 – 217. (2007).
6. Ross, J.W., Weill, P., Robertson, D.: Enterprise Architecture as a Strategy: Creating Foundation for Business Execution, Harvard Business School Press, Boston, Massachusetts. (2006).
7. Iyamu, T.: Strategic Approach used for the Implementation of Enterprise Architecture: A Case of Two Organisations in South Africa. In the Proceedings of the 11th International Conference on Informatics and Semiotics in Organisations, IFIP Press, 100-108. (2009).
8. Versteeg, G. and Bouwman, H.: „Business Architecture: A New Paradigm from Relate E-Business Strategy to ICT". Information Systems Frontiers, Vol. 8, No. 2, 91-102. (2006).
9. Patel, N.: Global E-Business IT Governance: Radical Re-Directions. In the Proceeding of the 35th International Conference on Systems Sciences, USA, Hawaii. (2002).
10. De Vries, M. and van Rensburg A.C. J.: Enterprise Architecture – New Business value perspectives, *South African Journal of Industrial Engineering May 2008 Vol. 19, No. 1-16. (2008).*
11. Hafner, M. and Winter, R (2008). Processes for Enterprise Application Architecture Management, Proceedings of the 41st Hawaii International Conference on System Sciences – 2008
12. Luftman, J.N. Papp, R., Brier, T. Enablers and Inhibitors of Business-IT Alignment, Communications of AIS, Vol 1, No 11. (1999).
13. Burke, B.: The Role of Enterprise Architecture in Technology Research, Gartner Inc. publication. (2007).
14. Cook, M.A.: Building enterprise information architectures: reengineering information systems, Prentice-Hall, Inc., Upper Saddle River, NJ. (1996).
15. Spewak, S.H.: Enterprise Architecture Planning: Developing a Blueprint for Data," Applications and Technology, John Wiley & Sons Inc., New York. (1992).

16. Iyamu, T.: Institutionalisation of the Enterprise Architecture: The Actor-Network Perspective. *International Journal of Actor-Network Theory and Technological Innovation,* Vol. 3, No. 1, 27 - 38. (2011).
17. McDavid, D.W.: A standard for business architecture description, IBM Systems Journal, Vol 38. (1999).
18. Youngs, R., Redmond-Pyle, D., Spaas, P., Kahan, E.: A standard for architecture description, Enterprise Solutions Structure, Vol. 38, No. 1. (1999).
19. Watson, RW.: An Enterprise Information Architecture: A Case Study for Decentralized Organizations. In the Proceedng of the 33rd International Conference on System Sciences, 59-70. (2002).
20. Aerts, A., Goossenaerts, J.and Hammer, D. and Wortmann, J.: Architectures in context: on the evolution of business, application software, and ICT platform architectures, Information & Management 41 781–794. (2004).
21. Tan, E. P., Gan, W. B.: Enterprise architecture in Singapore government. In Saha, P. (Ed.). 23. Handbook of Enterprise Systems Architecture in Practice. (2007).
22. Rosenberg, D., Stephens, M.: Use Case Driven Object Modeling with UML: Theory and Practice, APress. (2007).
23. Iyamu, T. and Roode, D.: The use of Structuration and Actor Network Theory for analysis: A case study of a financial institution in South Africa. *International Journal of Actor-Network Theory and Technological Innovation,* Vol. 2, No. 1, 1 - 26. (2010).
24. Giddens, A.: The Constitution of Society: Outline of the Theory of Structuration. Cambridge, UK; Polity Press. (1984).

Tiko Iyamu is a Professor of informatics at the Tshwane University of Technology, Pretoria, South Africa. He also serves as a Professor Extraordinaire at the Department of Computer Science, University of the Western Cape, South Africa. Before taken fulltime appointment in academic, he held several positions in both public and private organisations. He was Chief Architect and Head of Architecture & Governance at the Government and Private institutions, respectively. His research interests include Mobile Computing, Enterprise Architecture, Information Technology Strategy; and he focuses on Actor Network Theory (ANT) and Structuration Theory (ST). Iyamu is an author of numerous peer-reviewed journal and conference proceedings articles. He serves on journal board and conference proceedings committees.

Received: November 3, 2010; Accepted: May 27, 2011.

DOI 10.2298/CSIS101120042M

Avoiding Unstructured Workflows in Prerequisites Modeling

Boris Milašinović and Krešimir Fertalj

Faculty of Electrical Engineering and Computing, Unska 3,
10000 Zagreb, Croatia
{boris.milasinovic, kresimir.fertalj}@fer.hr

Abstract. Integrating prerequisite relationships, partially defined as graph components, produces a directed graph that corresponds to a well-defined and well-behaved workflow consisting only of and-splits and and-joins. Such a workflow often cannot be transformed to a structured workflow. This paper presents an approach to producing a corresponding structured workflow that will, with some adjustments in the runtime, correspond to the original unstructured workflow. The workaround is based on element cloning and on a workflow wrapper handling clones in order to avoid multiple element instances. An algorithm for finding clones and an algorithm for reducing the number of clones are proposed. Correctness of the algorithms is analyzed and some drawbacks and possible improvements are examined.

Keywords: Workflow management, structured workflows, unstructured workflows, modeling prerequisite relationships, and-splits

1. Introduction

In order to follow the key guidelines for designing a business layer of an application [17] one of the tasks is to extract a workflow component that defines and coordinates multistep business processes. Although traditionally related to enterprise systems, workflow management has other applications. Applications of prerequisite relations between workflow elements can be found in merging dependencies between UML components [8], in modeling course prerequisites in a learning management systems [4][14][22], in modeling relationships between workflow sub-components of a learning management system [27] or in modeling workflow-based data for e-learning [25]. Development of proprietary workflow management software can enhance flexibility and help in avoiding limitations of existing systems in dealing with synchronization points in unstructured workflows [18], but it requires a lot of work and raises compatibility and interoperability issues. On the other hand the use of existing workflow management software induces problems during the design of the business process model, because the workflow management systems impose different syntactical restrictions on models. One of the restrictions is that the workflow model should (and often

must) be structured. Perspectives regarding this restriction vary. In [9] authors propose that workflow models should be structured in order to avoid the so called “spaghetti business process modeling”, but there might be good reasons to use unstructured business processes [5].

In this paper an approach to avoiding unstructured workflows in prerequisite modeling using directed graphs is described. The idea was introduced in [19], but here we elaborate on how to produce a structured workflow that will, with some additional adjustments in the runtime environment, correspond to the original unstructured workflow. An overview of a related work is given in the second section. The third section describes the process of merging partially defined prerequisite relationships into a directed graph that corresponds to a workflow with only and-split and and-joins and further expands on the idea of a modification based on the graph"s vertex cloning. Finding vertices that have to be cloned is done using an original vertex labeling algorithm formally described in the fourth section. The fifth section deals with reducing the number of clones prior to transforming an integrated graph into a concrete workflow model. For the sake of presentation simplicity we use the Windows Workflow Foundation model [23] (in further text WF-model). The algorithm for reducing the clones is described in the sixth section. The seventh section proves the correctness of the proposed algorithms and the eighth section discusses their complexity. The ninth section deals with possible improvements of the proposed algorithms.

2. Related Research

In [11] authors define a well-behaved workflow as one that can never lead to a deadlock or result in multiple active instances of the same activity. Another body of research [7] defines well-formed workflows, relates them to well-behaved workflows and defines prerequisites that a well-formed workflow must have in order to be structured.

Tools for transforming a process model into a corresponding structured model [21] and for translating an unstructured workflow into a structured BPEL model [13], [2] automate the transformation, but not all models can have their structural pair. In [11] and [12] it is shown that there are well-behaved workflow models that cannot be modeled as structured workflow models. Moreover, [15] shows that a workflow containing only and-splits [24] is always well-behaved but does not have a structured mapping if it does not meet certain requirements. Intuitively expressing those requirements, in order to have a structured mapping, such workflow must have and-splits paired with corresponding and-joins in a parallel routing form [1].

An approach for prerequisite modeling in which dependencies are modeled in a tree form, which has the target element as the root node of the (sub)tree and its prerequisites as its children [8], enables direct transformation to the structured workflow, but can produce large trees with same elements in different subtrees. For such elements the same dependency subtree has to

be added several times and the tree can grow rapidly. As there might be many elements having a common prerequisite, which is quite common in course modeling, it is better to use directed graphs. The more common prerequisites exist, the more obvious the benefit of this approach is. The drawback of this idea is that such graphs usually produce unstructured workflows.

The idea how to produce a structured workflow that will, with some additional adjustments in the runtime environment, correspond to the original unstructured workflow introduced in [19] is based on workflow elements cloning (duplication), which in normal circumstances leads to multiple instances of an element but one can resolve these in the runtime. As clones will formally be different elements, the formal verification techniques will treat such a workflow as a structured workflow. Although this approach also introduces duplicates, their number should be significantly less than using trees due to common split and merge points.

3. Partially Defined Prerequisite Relationships

Each prerequisite relationship can be graphically presented as a directed arc between two vertices. Direction of the arc defines the dependency of a target vertex upon a source vertex. The vertices and arcs form a single (not necessary connected) integrated directed graph in which each vertex can have one or more incoming and outgoing arcs. Each component of a graph with one source and one sink vertex can be paired with a well-formed workflow model containing only and-splits and and-joins. **Fig. 1** shows this straightforward process of transformation from a directed graph into a well-formed workflow model. Each vertex having two or more outgoing arcs is presented with one workflow element connected to an and-split with outgoing arcs moved from the vertex to an and-split. Each vertex having two or more incoming arc is presented with one and-join and one workflow element, where incoming arcs are moved to the and-join.

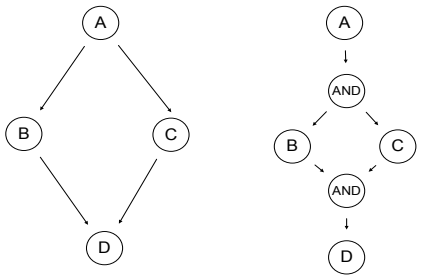


Fig. 1. From a directed graph to a well-formed workflow model

With some modifications described later in this section, it is guaranteed that the integrated graph will be connected and will have one source and one sink

vertex and therefore can be represented with one well-formed workflow. In the remainder of the paper, workflow models will be depicted as directed graphs without explicitly shown and-split and and-join elements in order to simplify the figures.

The duplication of vertices [11] in the process of transformation of simple workflows without parallelism into structured workflows cannot be directly applied because it would lead to multiple instances of certain elements. If it can be guaranteed that multiple instances are handled in the way that only one of them is effectively executed during the runtime, then the corresponding structured workflow can be created. We propose cloning common vertices (workflow elements) and putting them into parallel branches. Concrete implementation of the workflow model must ensure that the corresponding clones behave as a single element and that no data duplication occurs in the runtime. This can be ensured by building a workflow wrapper that would expose only unique instances of elements to other layers of the application.

In order to briefly illustrate the idea, two graphs are shown in the **Fig. 2**. The graph on the left hand side corresponds to an unstructured workflow. If the vertex C is cloned into two instances C1 and C2, and if C1 is prerequisite for E and C2 is prerequisite for F the outcome is a structured workflow model presented in the graph on the right hand side. The graph on the right hand side is equivalent to a structured workflow that can be easily transformed into a WF-model (or any other concrete workflow model). In the runtime it has to be guaranteed that C1 and C2 are treated as a single instance of C. After the integration, vertices such as vertex C in this example have to be labeled in order to be cloned later in the process of the WF-model creation.

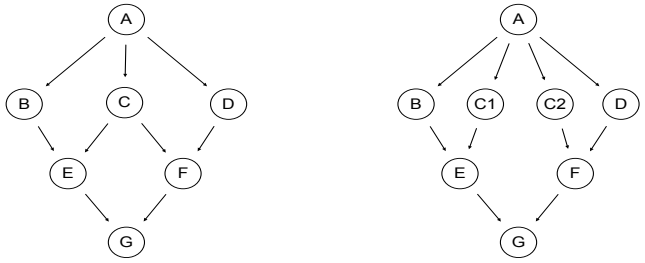


Fig. 2. Unstructured workflow and resulting structured workflow

The complete modeling process consists of several steps as shown in the Fig. 3. Firstly, partially defined prerequisite relationships have to be merged into an integrated graph.

The efficient way of doing the initial integration is to represent the graph with an adjacency matrix. Vertices have to be uniquely mapped to positive numbers ranging from 1 and the total number of the vertices. Creating adjacency matrix eases the detection of (in terms of prerequisites) illicit situations, such as the existence of graph cycles of length 1 and 2. In order to find cycles of length more than 2, algorithms shown in [10] or [20] can be used. The latter is suitable when short cycles are expected.

As the goal is to find at least one cycle and since the topological order is necessary for the next steps of the process, the better approach would be to use the algorithm for incremental cycle detection and topological ordering shown in [6].

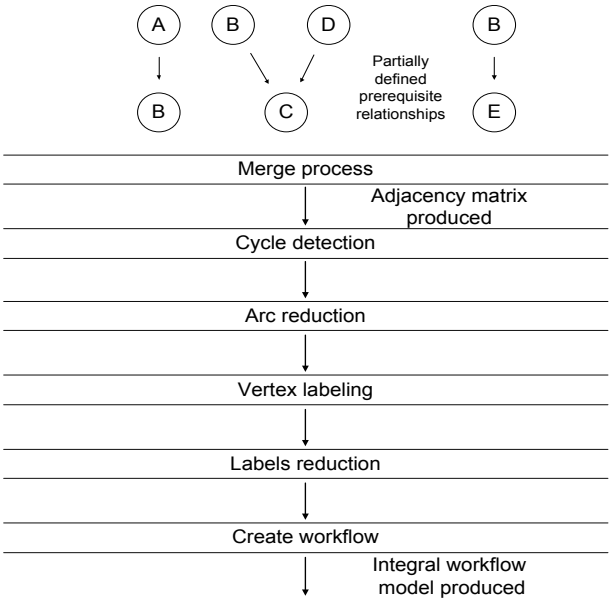


Fig. 3. Phases of the workflow creation algorithm

After creating the adjacency matrix and completing the detection of cycles, the algorithm proceeds with arcs removal. Due to transitivity of prerequisite relation, there is a possibility that some of the arcs could be removed. In a nutshell, if for some arc *a* = *(A,B)* an alternative path from vertex *A* to vertex *B* exists, then the arc *a* can be removed because it is redundant. For example arc *a* = *(A,B)* in the **Fig. 4** is removed since *A* is a transitive predecessor of *B* via *C* and *D*.[1] This step is similar to the transitive reduction of a graph [3].

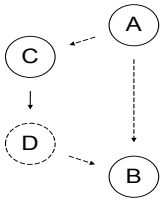


Fig. 4. Removing a redundant arc from A to B

[1] In terms of logic, the situation from **Fig. 4** can be expressed as B=>A, B=>D, D=>C, C=>A, which is equivalent to B=>D, D=>C, C=>A. (B=>A is a surplus)

An alternative to cycle detection and transitive prerequisites removal is the use of algorithms for Boolean satisfiability problem (SAT). Each arc *a=(i,j)* means that *i* is a prerequisite for *j* and such dependency can be logically represented as *j* entails *i*. The logic implication *j=>i* is evaluated to false only in case that *j* is true and *i* is false which corresponds to the situation where some activity is started without its prerequisite being finished. All other combinations (both false, both true, only the prerequisite has finished) are considered to be valid. Other options can be the algorithms presented in [16] and [26], but as prerequisite relationships produce only and-splits and and-joins the initial approach is enough.

Depending on the arrangement of the prerequisites it might happen that a graph is not connected which is not suitable for the algorithm steps that should follow. Therefore, after the reduction of arcs algorithm, two new vertices have to be added to the graph: *Start* and *End*. *Start* vertex will be connected with all vertices having number of inward arcs equal to zero in such a way that *Start* is their predecessor. All vertices having their outward degree equal to zero will be connected with the *End* vertex in such way that *End* is their successor. Consequently, it is guaranteed that the graph is connected that is essential for the vertex labeling algorithm, illustrated in the next section.

4. The Vertex Labeling Algorithm

The vertex labeling algorithm can be applied to a directed graph $G = (V, A)$ with the set of vertices V and set of directed arcs A according to the following presumptions:

- Graph G is connected and there are no cycles in the graph
- There is only one start vertex $v \in V$ such that its indegree $\deg^-(v) = 0$
- There is only one end vertex $v \in V$ such that its outdegree $\deg^+(v) = 0$
- For each pair of vertices $p \in V$ and $q \in V$ such that exists an arc between them ($a \in A$, $a = (p, q)$) there is no other path $p v_1 v_2 \dots v_n q$ ($v_i \in V$) from p to q in graph G.

Cycle detection, arcs reduction and the introduction of *Start* and *End* vertices ensure that the previous presumptions are satisfied.

Definition 1. Vertex label is a string that contains only natural numbers and dots.

Set of all possible labels is marked with $\mathcal{L}$. Mapping $\ell: V \rightarrow L$, where $L \subset \mathcal{L}$, assigns one or more labels to each vertex in the graph. The vertex labeling algorithm is presented in the **Table 1**.

Labeling starts from the *End* vertex that receives label 1 and is added to the set of “open” vertices. Open vertex is a vertex that still has not forwarded all its labels to its predecessors. The labeling algorithm will in each step take a

new vertex from the set of open vertices. The chosen vertex will forward its labels to its predecessors according to the following rules.

- If the vertex has only one predecessor then all its labels are added to the predecessor"s label set without being changed.
- If the current vertex has N predecessors (N ≥ 2) then each label is concatenated with *.i* where *i* is a predecessor"s order number. In this way the first predecessor in the order will receive all vertex labels concatenated with *.1*. For the second predecessor *.2* will be concatenated and same principle respectively applies to all other predecessors until the *N*-th predecessor which receives labels concatenated with *.N*. The current vertex will be removed from the set of open vertices and its predecessors will be added to the set.

Table 1. Vertex labeling algorithm

1. Assign label '1' to *End* vertex and add it to the open vertices set
   ℓ (*End*) := {'1'} , O := {*End*}

2. Let *curr* be the last vertex in the topological order of set O and let P be a set of its predecessors in the graph.
   $curr$:= $w \in$ O such that $v \prec w$, $\forall v \in$ O, $v \neq w$
   P := { v | $\exists a \in A$ such that a = (v, $curr$)}

   if |P| = 1 then
   $\ell(v)$:= $\ell(v) \cup \ell(curr)$ where $v \in$ P
   if |P| > 1 then
   for each $v \in$ P do
   $\ell(v)$:= $\ell(v) \cup concat(\ell(curr),$ ".orderNumber")
   where *concat* is function that will add text from the second parameter as a suffix to each label in label set in the first parameter.
   orderNumber represents the order number of each predecessor in P (possible values are 1 to |P|)
   O := (O ∪ P) \ {curr}

3. If |O| > 0 repeat step 2.

If a random or depth first algorithm is applied in choosing a new vertex in each step of the algorithm, the chosen vertex could be added to and removed from the set of open vertices several times depending of the vertex order and the number of its successors. This can be solved by tagging the labels as they are forwarded. Instead, we choose to take vertices in a reversed topological order, which ensures that at the time of dealing with a vertex all its successors have already forwarded their labels.

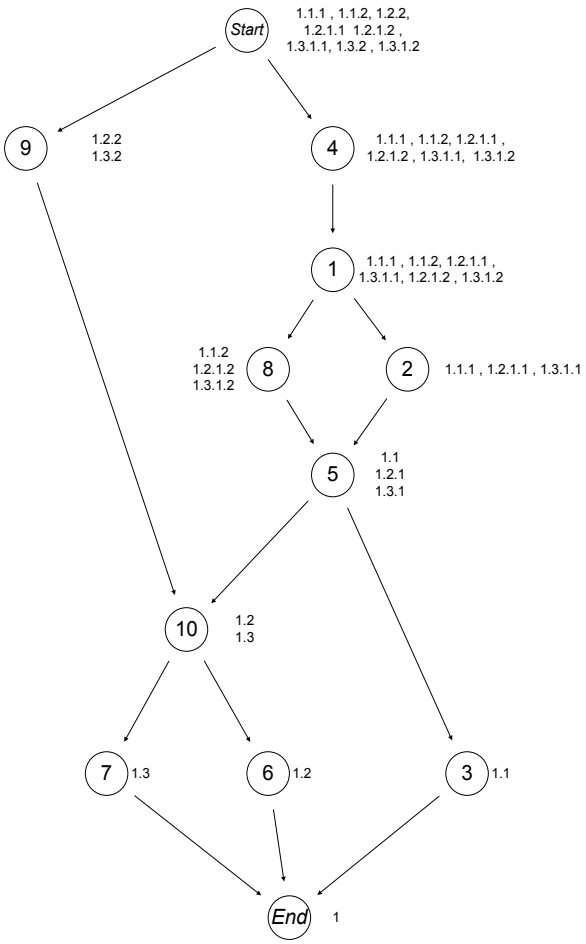


Fig. 5. An illustration of the vertex labeling algorithm operation

An example of the vertex labeling algorithm is shown in the **Fig. 5**. Presuming the topological order was *Start*, 9, 4, 1, 8, 2, 5, 10, 7, 6, 3, *End*, in the first step of the algorithm, label 1 is assigned to *End* vertex and *End* is added to the set of open vertices. Vertex *End* has 3 predecessors – 3, 6 and 7 respectively. As *End* has more than one predecessor, label 1 is suffixed with .1 and added to the label set of vertex 3, to label set of vertex 6 with suffix .2 and to label set of vertex 7 with suffix .3. After that, vertex *End* is removed from the set of open vertices and vertices 3, 6 and 7 are added to that set. Vertex 3 is the last in the topological order of all open vertices and it is processed in the next step. It has only one predecessor (vertex 5) and its labels (the only label is 1.1) are added to vertex 5 without modification. Vertex 3 is removed from the set of open vertices and vertex 5 is added to the set. The process proceeds further with vertex 6 (that copies its label to vertex 10) and after that with vertex 7. At this point the set of open vertices consists of

vertices 5 and 10. Vertex 10 is processed because it is behind vertex 5 in the topological order. Vertex 10 has two predecessors and each of them receives labels from vertex 10. This way, vertex 5 receives 1.2.1 and 1.3.1 and vertex 9 receives 1.2.2 and 1.3.2. Step 2 is repeated until the *Start* vertex with no predecessors is processed making the set of open vertices empty.

5. The Label Reduction Algorithm

Since the cardinality of a vertex label set is the number of the clones of the vertex while producing a WF-model, it is reasonable to reduce the cardinality of each set when possible. A brief look at the **Fig. 5** leads to a conclusion that vertices 1 and 5 should have the same labels. There is no arc from vertex 1 that is not joined in vertex 5 and there is no inbound arc for vertex 5 that cannot be traced back to vertex 1. For example, a part of the WF-model is presented in the **Fig. 6** where parallel activities occur after vertex 1 followed by activity created for vertex 5 that was added after the synchronization point. Therefore label set of vertex 1 can be reduced to have same labels as vertex 5.

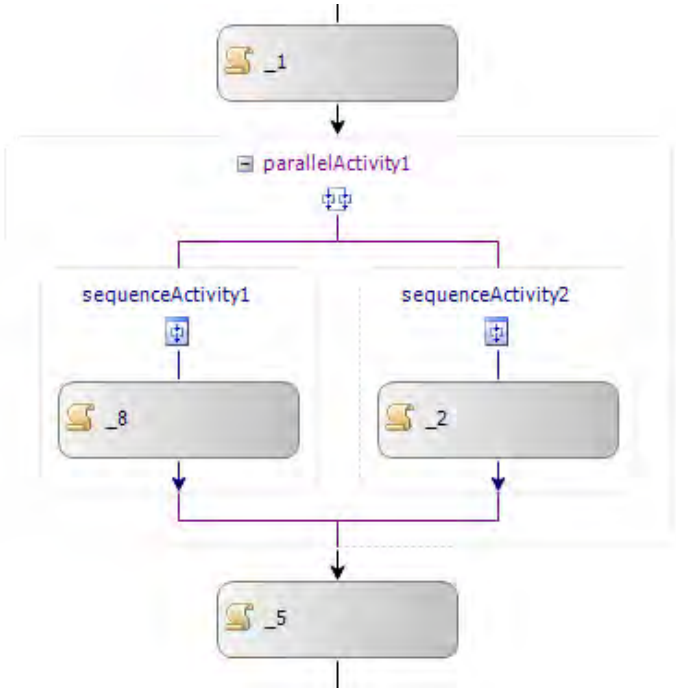


Fig. 6. WF-model for excerpt of graph from Fig. 5

Definition 2. Function *level*: $\mathcal{L} \rightarrow \mathbb{N}$ (where $\mathcal{L}$ is the set of all possible labels) is defined as the number of dots inside a label.

In our example, *level* (1) is 0 and *level* (1.3.1.1) is 3. The relationship between labels will be expressed as a parent-child and an ancestor-successor relation. For instance, label 1.3.1.1 is a child of label 1.3.1 and a successor of label 1. Vice versa, label 1 is an ancestor of labels 1.3.1 and 1.3.1.1, while label 1.3.1 is a parent of label 1.3.1.1.

Definition 3. Label l is a parent of label m if *level(m) = level(l)+1* and m and l are the same until the last dot in both labels.

Definition 4. Label l is an ancestor of label m if l is a parent of m or there exists a chain of labels $l_1 \ldots l_n$ such that l is a parent of l_1, $l_1$is a parent of l_2,… and l_n is a parent of m.

The basic idea behind the label reduction algorithm is to replace labels with their common parent. The replacements are done for vertices having all children of a particular parent label. For example, if all children of label 1.2 in a graph are labels 1.2.1, 1.2.2, 1.2.3 then for all vertices that contain all those three labels, those three labels are replaced with 1.2.

The label reduction algorithm groups labels by value of level function and starts checking labels backwards from the next to the last level until it comes to the first level where the return value of function *level* for the first level is equal to zero. For each label l in current level (marked as set U in the following algorithm), set S is a set containing all children of the label l. The next step is finding all vertices (set V') that contain all labels from the set S. This way, vertices containing all children of the label l are found and for those vertices the replacement can be done. In the end, for each vertex from set V', all labels from the set S are removed from its label set and label l is added.

The algorithm continues until all labels on all levels have been checked, i.e. until label 1 (the only label at the level 0) is checked. Formal definition of the algorithm follows in the **Table 2**.

Table 2. Label reduction algorithm

1. curr_level := $\max_{l \in \mathcal{L}} level(l) - 1$

2. Let U be the set of all labels from current level
   U: = { $l \in \mathcal{L}$ | level(l) = curr_level}
   for each label l from set U do
   S := { m | $m \in \mathcal{L}$ such that l is a parent of *m*}
   If S ≠ ∅ then
   V' := { v | $v \in V$ such that $m \in \ell(v)$, ∀ $m \in$ S}
   for each $v \in$ V' do
   $\ell(v)$:= ($\ell(v)$ \ S) ∪ { l }
3. curr_level := curr_level - 1
   If curr_level ≥ 0 repeat step 2.

For the graph from the **Fig. 5** label reduction occur for the vertices 1, 4 and *Start*. Labels 1.3.1.1 and 1.3.1.2 are replaced with 1.3.1, labels 1.2.1.1 and 1.2.1.2 are replaced with 1.2.1 and labels 1.1.1 and 1.1.2 are replaced with 1.1. Next replacements occur only for the vertex *Start*. Newly added labels 1.3.1 and label 1.3.2 are replaced with 1.3. Newly added labels 1.2.1 and 1.2.2 are replaced with 1.2. Therefore, a new replacement can be done. Labels 1.1, 1.2 and 1.3 are replaced with 1.

After the algorithm ends, vertex *Start* must only contain label 1. **Fig. 7** shows the graph after the label reduction has been applied to the graph in the **Fig. 5**.

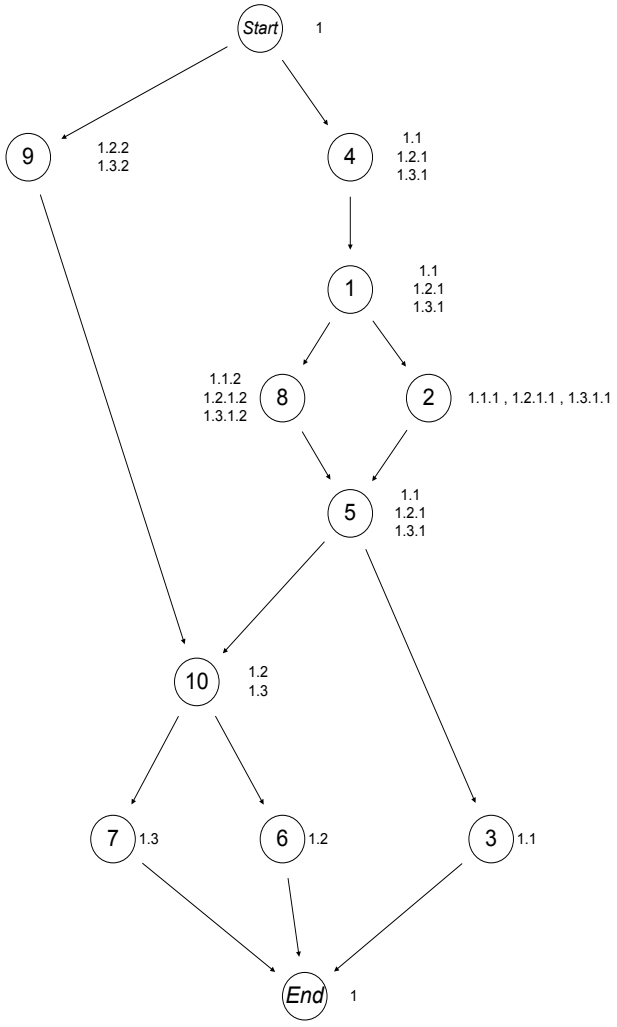


Fig. 7. An example of a graph after the reduction of labels

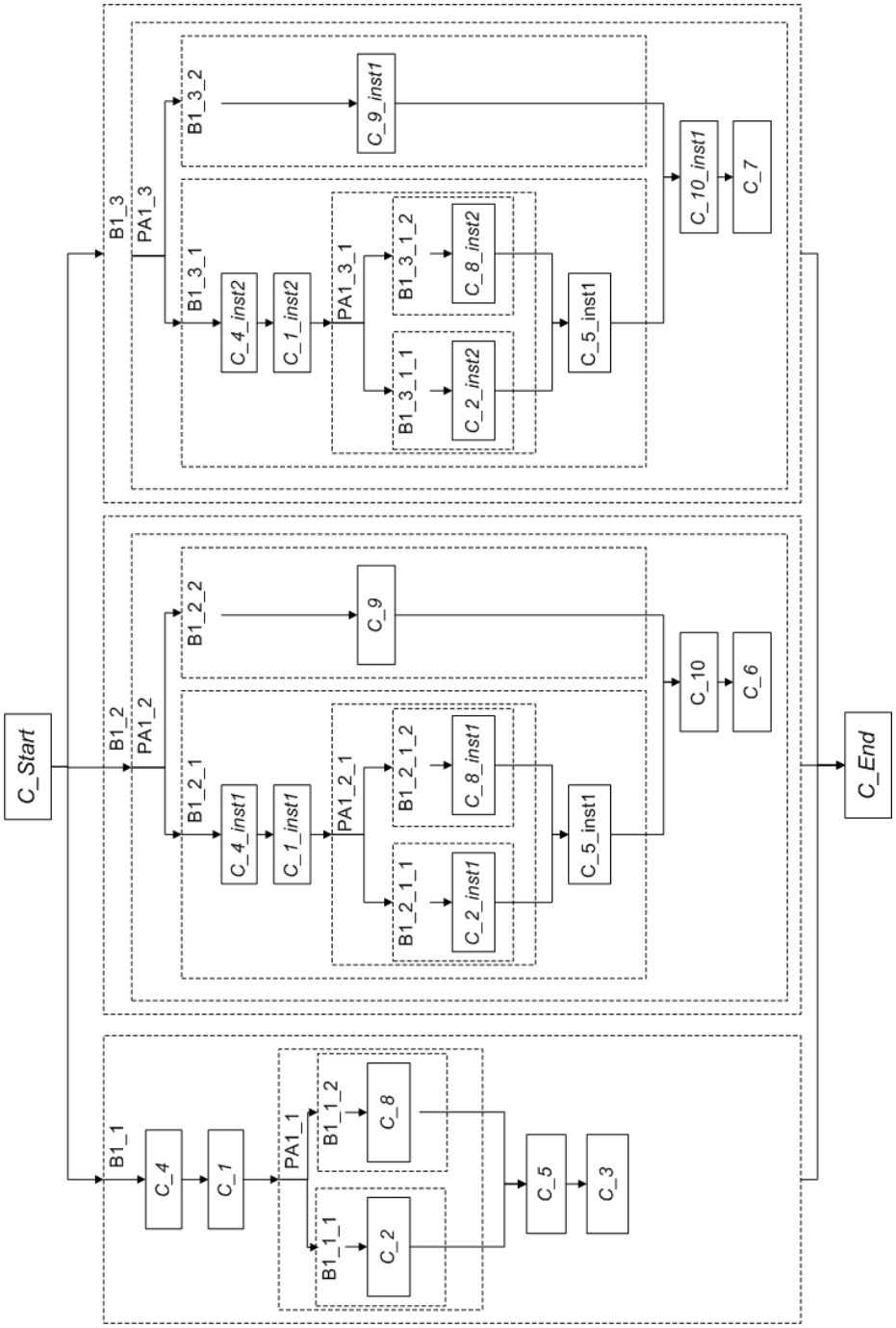


Fig. 8. WF-model for the labeled graph from Fig. 7

After the labels have been reduced, a WF-model can be created. Some vertices will be cloned and the corresponding WF-elements will be added as several parallel branches. WF-model for the previous example is presented in the **Fig. 8** where the elements are named with C_{vertex number}, parallel activities are named with PA_{label name} and branches of parallel activities are named with B_{label name}. In case a vertex had to be cloned, inst_{clone instance number} is appended to the element name in order to ensure that the element names remain unique. The principle of the WF-model creation is described in the next section.

6. WF-model Creation

Each element of a created WF-model will be either an activity that represents a clone of a vertex in the labeled graph or a built-in activity (*ParallelActivity* or *SequenceActivity*). In every WF-model at least one (main) sequence must exist, which contains all elements having label 1. For any other label corresponding new parallel activities and parallel branches will be created and assigned to the particular label. For example, vertex with the label 1.3.1.2 would be added in a branch assigned to the label 1.3.1.2. If such a branch does not exist it is created as a branch of a parallel activity assigned to the label 1.3.1. If such a parallel activity does not exist, it is created in the branch assigned to the label 1.3.1. This procedure is recursive and finishes on the main sequence assigned to the label 1. Names of the activities must be unique and follow variable naming rules in which parallel activities are prefixed with PA_ and branches of parallel activities take prefix B_.

For the graph from the **Fig. 7** a label is uniquely assigned to a particular parallel branch in **Fig. 8** and branch where an activity for a vertex will be added can be uniquely determined. It is important to note that a label does not have to uniquely identify a branch as it can be seen in **Fig. 9** and **Fig. 10**. Label is rather a indicator where a particular element would be nested within the workflow relative to its current position. The main benefit of this principle is that label set can be further reduced.

Vertices from the graph are processed in the topological order, which ensures that all merge points are added after its parents. For instance, a topological sort for the graph in **Fig. 9** can be *Start*,1,2,5,3,4,6,7,8,*End*. The algorithm starts by adding vertex 1 with label 1 (WF-element has name C_1) to the main sequence activity. Afterwards, an element representing vertex 2 with label 1.1 is processed and must be added to the branch assigned to label 1.1. As such branch still does not exist and a parallel branch assigned to label 1 does not exist, a new parallel activity (named PA_1) is added to the main sequence and assigned to label 1. Subsequently branch of a parallel activity PA_1 is created, named B_1_1 and assigned to label 1.1 upon which WF element for vertex 2 is added to the newly created branch.

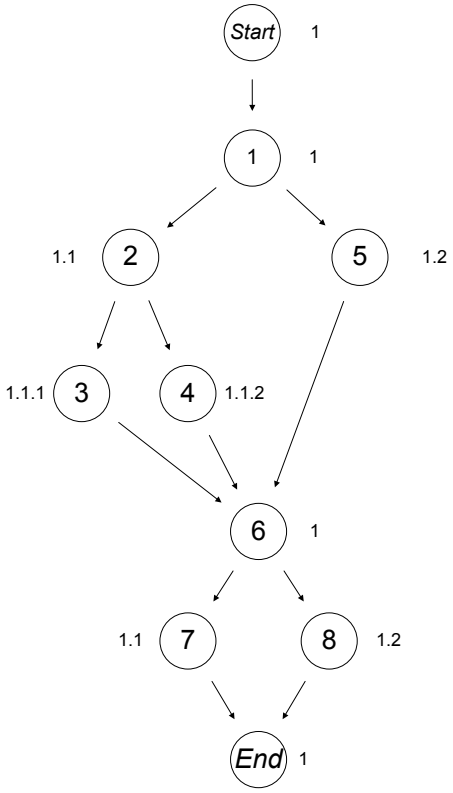


Fig. 9. A simple graph illustrating the creation of a WF-model

Similarly, element that represents vertex 5 is added to the branch B_1_2 assigned to label 1.2. Vertex 3 has label 1.1.1, which means that there must exist a branch assigned to label 1.1.1 as a branch of a parallel activity assigned to 1.1. As both do not initially exist, they will be created. The same procedure follows for all other vertices until vertex 6 is reached. Vertex 6 is the synchronization point of branches 1.1 and 1.2. As it has label 1 it will be added as the next child of the main sequence (after parallel activities). It is important to note that appearance of a label that is an ancestor of an existing label causes that all mappings between successor labels and parallel activities and branches get removed. These way vertices 7 and 8 are not added to the "old" branches B_1_1 and B_1_2 respectively, but a new parallel activity PA_1_inst1 is created containing branches B_1_1_inst1 and B_1_2_inst1. PA_1_inst1 is added to the main sequence after the element C_6. The new mapping for labels 1.1 and 1.2 is active until the next occurrence of label 1. WF-model of the graph in the **Fig. 9** is shown in the **Fig. 10**.

If a vertex has more than one label, previously described steps are repeated for each label.

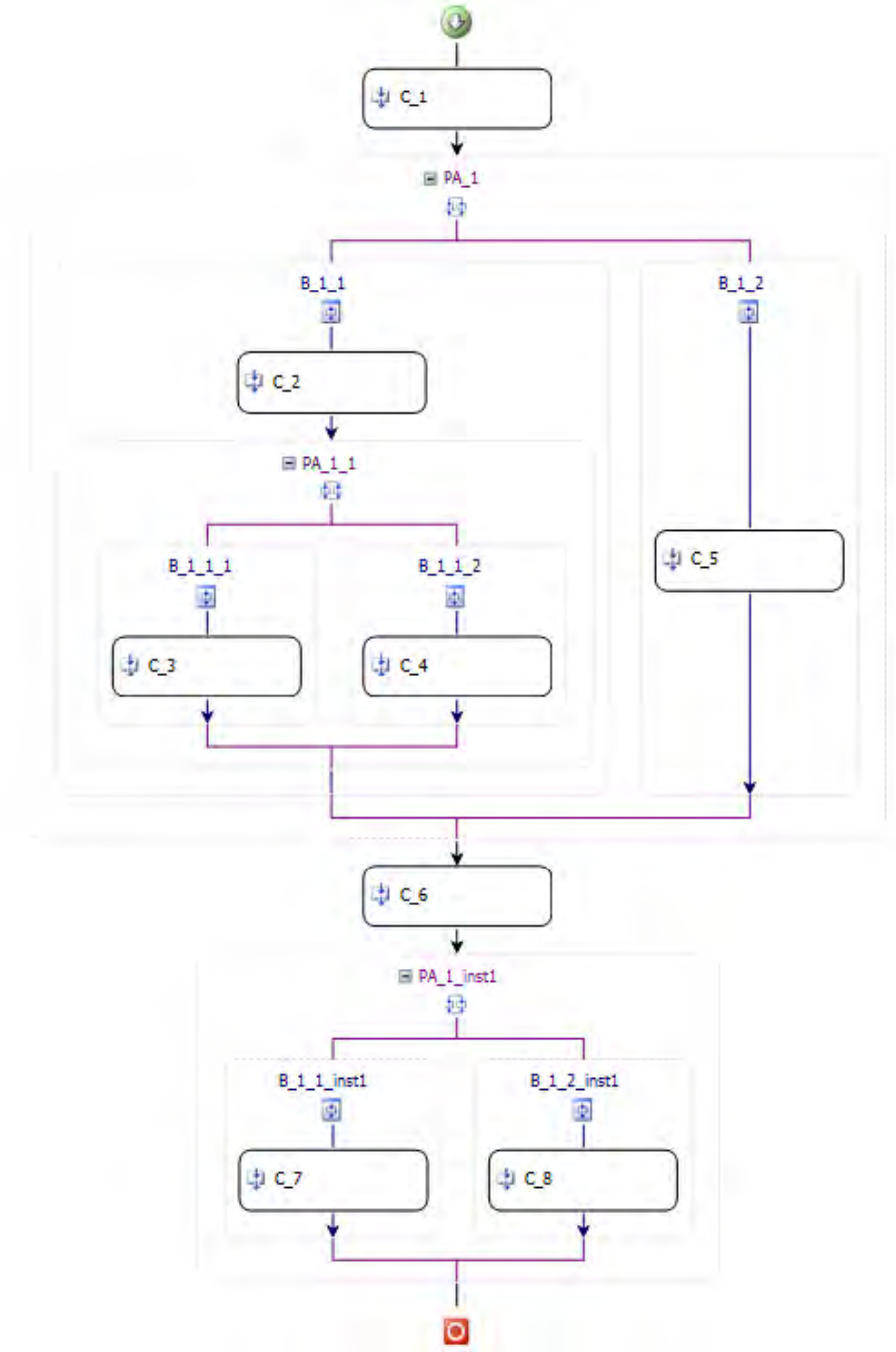


Fig. 10. An WF-model for the labeled graph in Fig. 9

Pseudo code for the WF-model creation algorithm is given in the **Table 3**. Prior to the formal definition of the algorithm two mappings must are defined.

Definition 5.

$p : \mathcal{L} \rightarrow$ PA is mapping between a particular label and the assigned parallel activity, where $\mathcal{L}$ is a set of all labels and PA is set of all parallel activities in WF-model

$b : \mathcal{L} \rightarrow$ B is mapping between a particular label and the assigned branch, where $\mathcal{L}$ is a set of all labels and B is a set of all branches for parallel activities in WF-model

with $b(1)$ initially defined in order to assign the main sequence to label 1.

Table 3. The WF-model creation algorithm based on a labeled graph

Let $G(V, A)$ is a graph labeled according to the algorithms from **Table 1** and **Table 2**

```
for i:=1 to i≤ |V| do
        v := i^th element from the topological order of the graph G

        for each label l ∈ ℓ(v) do
                D := { m | m ∈ ℒ such that l is an ancestor of m}

                if D ≠ ∅ then
                   for each m ∈ D do
                           p(m) := undefined
                           b(m) := undefined

                if b(l) is not defined
                        createBranch(l)

                add element C_v[2] in branch b(l)
```

Recursive function *createBranch* creates a branch for a given label. It depends on the function for creating parallel activity. Both functions are described in the next table.

[2] Name of the element can contain suffix *_instX* where X is the number of elements already created for vertex v

Table 4. The algorithm for creating a branch and parallel activity for a given label

createBranch(l)
    if $p(parent(l))$ is not defined
        createParallelActivity($parent(l)$)
    $b(l)$:= create branch B_ l[3] as a branch of parallel activity $p(parent(l))$

createParallelActivity(m)
    if $b(m)$ is not defined
        createBranch(m)
    $p(m)$:= create parallel activity PA_m as a next activity in branch $b(m)$

7. Correctness of the Workflow Creation Algorithm

In order to prove the correctness of the workflow creation algorithm it is necessary to prove two main claims. The first one is that algorithms for graph integration, arc reduction, vertex labeling and label reducing preserve all of the initial prerequisite relationships and create no new prerequisites. The second one is that the WF-model creation algorithm creates correct sequences and parallel activities.

By looking into the steps of the complete process of workflow creation it becomes clear that the first two steps only transform the partially defined prerequisites into a graph representation without any modifications. The third step (arc reduction) removes arcs representing prerequisite relationships that are already present through transitivity of some other prerequisites. The graph is modified but the meaning has not been changed – all prerequisites are kept either in the original relationship or as a set of transitive prerequisite relationships.

To show the correctness for the remaining algorithm steps (labeling, label reduction and WF-model creation) the following claims should be proven:

- All vertices are in correct order, which means that all prerequisite relationships are satisfied.
- Execution of each element depends only on its prerequisites; i.e. there is no waiting for elements which are not prerequisites.

From the design of the WF-model it is obvious that in order to have two elements run in parallel, it is imperative that their labels are not in an ancestor-successor relationship. The vice versa statement does not have to be valid (see **Fig. 10**, labels 1.1 and 1.2). The claim is formally stated through the following theorem.

Theorem 1. The non existence of an ancestor-successor relationship between labels l_x and l_y, where $l_x \neq l_y$, is required in order to have a WF-

[3] Name of the element can contain suffix *_instX* where X is the number of elements already created for vertex v

element for vertex X with label l_x and a WF-element for vertex Y with label l_y both running in parallel.

Proof: Without loss of generality it can be assumed that vertex X is before vertex Y in the topological order. There are two possible cases that must be discussed. In the first case there is no vertex between X and Y in the topological order having a label which is a common ancestor of labels l_x and l_y. The second case is that such vertex exists which shows that the presumption about non existence of the ancestor-successor relationship is not enough to run the elements in parallel.

Case I. According to the algorithm presented in the **Table 3**, the branches of parallel activities are created in a way that a branch B assigned to label l is branch of the parallel activity PA assigned to label $parent(l)$. Parallel activity PA is added to the branch assigned to label $parent(l)$ which is a branch of the parallel activity assigned to label $parent(parent(l))$ etc. Creating activity for vertex X with label l_x will create (if such does not already exists) a branch assigned to the label l_x. The same principle applies to the vertex Y with the label l_y. If $l_x = l_y$ then it is the same branch and X and Y are in the same branch which also means they are in the same execution line. If l_x is an ancestor of l_y then, according to previously described algorithm, the branch assigned to the label l_y is contained inside the branch assigned to the label l_x. If l_y is an ancestor of the label l_x it means that branch assigned to the label l_y already exists and activity representing a clone of the element Y is added to the end of that branch which means that Y is a synchronization point for branches contained in the branch assigned to l_y putting X and Y in the same execution line. If l_x and l_y are not in an ancestor-successor relationship then it is obvious that they can be run in parallel.

Case II. Appearance of label l that is a common ancestor of labels l_x and l_y, according to the algorithm from the **Table 3**, causes all assignments between branches and successors of the label l to be removed. As such, assignments for labels l_x and l_y are also removed. The clone of vertex with label l, such as one between X and Y, is a synchronization point for all branches that have been assigned to the labels l_x and l_y. This fact causes that when an element for the vertex Y is going to be created it gets added to the newly created branch assigned to l_y and that branch is a part of the branch assigned to the label l and, as such, is in the same line of execution with X.

■

As an outcome from the proof of the Theorem 1, Corollary 2 follows directly.

Corollary 2. Clone of a vertex X with label l_x and clone of a vertex Y with label l_y run in parallel if $l_x \neq l_y$, labels l_x and l_y are not in an ancestor-successor relationship and there are no vertices between X and Y in a topological order with a label that is common ancestor of labels l_x and l_y.

By observing not only a single clone, but all clones of a vertex it can be stated that two vertices are in the same line of execution if for each clone of one vertex exists a clone of the second vertex such that those two clones are in the same line of execution. If not, these vertices can be run in parallel. For example, for the graph from the Fig. 7 vertices 4 and 10 are in the same line of execution because for each label of vertex 10 (labels are 1.2 and 1.3) exists a label of vertex 4 (labels are 1.1, 1.2.1, 1.3.1) in an ancestor-successor relationship with that label. Vertices 9 and 3 represent an opposite situation. With Lemma 4 it will be shown that it is enough to find just one pair of labels in an ancestor-successor relationship so to conclude that the two vertices are in the same line of execution.

Corollary 3. For a WF-model created immediately after the vertex labeling algorithm, two vertices X and Y are in the same line of execution if and only if for each pair of labels $l_x \in \ell$ (X) and $l_y \in \ell$ (Y) label l_x is equal to l_y or label l_x is an ancestor of label l_y or label l_y is ancestor of label l_x.

Proof: The first direction follows directly from the Corollary 2 because there are no vertices between X and Y in a topological order having a label which is a common ancestor of labels l_x and l_y. Therefore l_x and l_y must be in an ancestor-successor relationship. The second direction follows from the Theorem 1, due to fact that nonexistence of an ancestor-successor relationship between labels is required to have two clones run in parallel. If all labels are in an ancestor-successor relationship then no pair of clones of vertices X and Y can be run in parallel, hence vertices X and Y are in the same line of execution.

■

Lemma 4. After the vertex labeling algorithm and before the labels reduction algorithm, for each two vertices X and Y, such that X is predecessor of Y, and for each label $l_y \in \ell$ (Y) exists a label $l_x \in \ell$ (X) such that either $l_y = l_x$ or l_y is an ancestor of l_x.

Proof: Case I. Let X be an immediate predecessor of Y. If X is the only immediate predecessor of Y then all labels of vertex Y are added to set of labels of vertex X without any modification and in this case is ℓ (Y) $\subset \ell$ (X). From this it immediately follows that for each label $l_y \in \ell$ (Y) exists a label $l_x \in \ell$ (X) such that $l_y = l_x$. If the vertex Y has two or more immediate predecessors and the vertex X is the i-th predecessor of Y then each label of the vertex Y is suffixed with *.i* while adding to label set of the vertex X. Therefore, for each label $l_y \in \ell$ (Y) there is a label $l_y.i$ in ℓ (X).

Case II. Let X be a non-immediate predecessor of Y. Then there exists a path $Xv_1v_2..v_nY$ in the graph. Labels from the vertex Y are transferred to v_n, from the vertex v_n to the vertex v_{n-1} and so on until the vertex X. In each step one of the two possible situations from the first case can be applied. Therefore for each label $l_y \in \ell$ (Y) either exists a label $l_x \in \ell$ (X) such that $l_y = l_x$ (in case all vertices in the path had only one immediate predecessor) or l_y

is an ancestor of l_x as a consequence of successive suffixing with dot and the order number of a particular predecessor.

■

Lemma 5. After the vertex labeling algorithm two vertices cannot be in the same line of execution if one vertex is not a predecessor of the other.

Proof: Let vertices X and Y be in the same line of execution, but such that neither X is a predecessor of Y, nor Y is a predecessor of X. Let vertex Z be the first (in the topological order) common successor of the vertices X and Y. Such a vertex must exist because the graph is a connected graph having at least vertex *End* as such a vertex. Since X and Y are according to the presumption in different ancestry lines, after transferring labels from the vertex Z to its predecessors, ancestry lines toward vertex X have received labels of the vertex Z with the suffix *.i*, and ancestry line toward the vertex Y has received labels of the vertex Z with the suffix *.j*. As there was no label reduction, vertices X and Y have labels that are not in an ancestor-successor relationship and according to the Corollary 3 cannot be in the same line of execution which is in the contradiction with the assumption. Therefore X must be a predecessor of Y or Y must be a predecessor of X.

■

Lemma 4 shows that before the label reduction all predecessors and successors of a vertex (and according to the Lemma 5 only these) are in the same line of execution with the vertex due to the existence of an ancestor-successor relationship between their labels. It has to be shown that the label reduction algorithm keeps the execution line.

Lemma 6. Two vertices X and Y are in the same line of execution after the label reduction algorithm if and only if they were in the same line of execution before the label reduction algorithm.

Proof: ⇒ Let vertices X and Y be in the same line of execution after the label reduction algorithm. If it is assumed that X and Y were not in the same execution line before the label reduction, then their lines of execution split at their common ancestor and join in the first (in the topological order) common successor. As labels coming from the common successor are propagated to two different lines of execution with different suffixes, neither one element in the different ancestry lines (including X and Y) cannot have all children of a label l from the common successor"s label set and that is true for each label l of the common successor. Therefore X and Y cannot reduce their label set to have same labels as those in the label set in the other execution line or to become their ancestors or successors. According to the Lemma 3 this means that they were in the different lines of execution and no label reduction could occur.

⇐ Let vertices X and Y be in the same line of execution before the label reduction algorithm and let, without loss of generality, X be a predecessor of Y. This means that before the label reduction each label $l_y \in \ell$ (Y) was equal to a label $l_x \in \ell$ (X) or l_y was an ancestor of the label l_x. Label reduction algorithm can change the label set of the vertex X and/or the vertex Y and

three different mutual relationships between the vertices in the same line of execution have to be observed.

In the first one, every path from the vertex *Start* to the vertex Y includes the vertex X and every path from the vertex *Start* to the vertex *End* such that includes the vertex X also includes the vertex Y (e.g. vertices 1 and 5 from **Fig. 5**). In this case, after the label reduction algorithm vertices X and Y will have the same labels because X contains all children of labels originated by coping labels of the vertex Y to its predecessors.

In the second case each path from the vertex *Start* to the vertex Y includes the vertex X, but there exists a path from the vertex *Start* to the vertex *End* that includes the vertex X and does not include the vertex Y (e.g. vertices 1 and 2 from the **Fig. 5**). If the vertex X had contained labels that have the same parent as the labels of the vertex Y, X"s label set would be reduced and those labels would be replaced with their parents. In the successive reduction it may happen that for a label $l_y \in \ell$ (Y) its corresponding label $l_x \in \ell$ (X) is swapped with a label that is an ancestor of l_x. Therefore the new label is also in an ancestor-successor relationship with the label l_y, and according to the Lemma 3, vertices are still in the same line of execution.

The third case is represented with a situation in which there exists a path from the vertex *Start* to the vertex Y which does not include the vertex X (e.g. vertices 4 and 7 from the **Fig. 5**). (The existence of the vertex Y in any path from the vertex *Start* through X to the vertex *End* is irrelevant.) In this case the vertex X in the label reduction algorithm cannot remove a label $l_x \in \ell$ (X) that is a successor of a label $l_y \in \ell$ (Y), because there exists at least one label that is a successor of the label l_y in another ancestry line and as such is not an element of ℓ(X) and therefore the relationship between X and Y has not been changed.

■

Theorem 7. Every clone of a graph"s vertices transformed into WF-activities depends only on its predecessors in the graph and no clones can be executed before all predecessor clones have been finished.

Proof: The Lemma 4 proves that all labels of a vertex are propagated to its predecessors and no predecessors of the vertex are omitted. This means that all predecessors are in the same line of execution before the label reduction. Lemma 5 proves that for each vertex, a vertex in the same line of execution cannot exist if it was not a predecessor of a successor of the vertex. Lemma 6 ensures that the label reduction algorithm keeps the line of execution, which means that all predecessors are preserved and no new dependencies have been added. Mutual vertical order of the clones inside the WF-model is ensured using topological ordering.

■

8. Algorithms Complexity

Complexity of the presented algorithms depends on the number of (different) labels, which is dependent not only on the number of vertices and arcs in a graph, but also on the graph's structure. As the worst case for a particular algorithm step can lead to significant simplification of another step in the process, in this section we give main guidelines on how to calculate the number of labels and which implementations to use in order to reduce the complexity of the presented algorithms. It is important to stress out that the given complexity is the upper bound and it can be far from the average complexity.

In the step 2 of the vertex labeling algorithm from the **Table 1**, each vertex having indegree greater than one creates new labels in the graph. A contribution of the vertex to the number of new labels in a graph is a product of the cardinality of the vertex label set and the number of the inbound arcs. The cardinality of a vertex label set represents the number of different paths from that vertex to the *End* vertex and can be calculated as a sum of cardinalities of label sets of its successors. For a vertex v in a graph $G = (V, A)$ this contribution of a vertex v can be expressed with the following formula

$$c(v) = \begin{cases} deg^-(v) * |\ell(v)| & , deg^-(v) > 1, v \neq End \\ deg^-(End) & , deg^-(v) > 1, v = End \\ 0 & , deg^-(v) = 1 \end{cases}$$

where

$$|\ell(v)| = \sum_{w \in S} |\ell(w)|$$

and $S = \{ w \mid \exists\, a \in A$ such that $a = (v, w)\}$.

This way, total number of different labels in the graph is equal to the sum of contributions of all vertices incremented by one due to the label 1 initially added to graph and assigned to the vertex *End*.

$$|\mathcal{L}| = 1 + \sum_{v \in V} c(v)$$

In the vertex labeling algorithm from the **Table 1**, a sub-step of the step 2 is executed $|A|$ times, where $|A|$ is the number of arcs in the graph. The complexity of the algorithm then depends on the implementation of relationship between labels and vertices, implementation of the union operation (with concatenation) and the number of labels in a vertex.

As it is shown later in this section, it is convenient to implement vertex label set as a hash table and store all labels in a tree, where each node additionally contains a pointer to the workflow branch assigned to the label and a hash table of pointers to vertices that have the label in its label set. If the tree is balanced then the addition of a new node to the tree has complexity of $O(\log|\mathcal{L}|)$ and the addition of a new vertex pointer into the hash table for the retrieved node can be done in constant time. As insertion of a new vertex in

the set of open vertices in such a way that vertices are always sorted by topological order and producing the union of labels is linear in relation to the number of vertices and the number of labels respectively, it can be claimed that the complexity of the vertex labeling algorithm is $O(|A| * (|\mathcal{L}| + |A|))$.

With labels stored in the tree, the label reduction algorithm can be implemented as an inorder tree traversal in which for each non-leaf node a slightly modified step 2 from the **Table 2** is executed. In that case finding the set V' for label l in step 2 from the **Table 2** is, complexity-wise, equal to finding the intersection of all hash tables with vertices pointers to the children of the node traversed and each intersection can be executed with linear complexity. Replacement of children labels with label l for all vertices from the set V' is done with the following single step: for a vertex from intersection and for a child label m of the label l, m is deleted from the vertex label set, and vertex pointer is deleted from the hash table in the node m. As the deletion from hash table can be done in constant time and as this occurs for each vertex and each label in each step of a traversal process, the complexity is $O(|V| * |\mathcal{L}|^2)$.

A single step of the inner loop in the WF-model creation algorithm consists of removing the existing mapping, creating a new branch and adding an element to the branch. This step is executed $\sum_{v \in V} |\ell(v)|$ times. If the tree is balanced, finding an assigned branch to a label is done in logarithmic complexity. When a mapping between a label and a branch has to be removed, the pointer to the branch has to be deleted from the tree. Instead of deleting pointers to the branches in entire subtree, to save on complexity it is enough to delete just the pointer in root of the subtree. This implies that the creation of a new branch has to be modified in such a way that, except assigning new pointer to a node, the deletion of all pointers in the child has to be done. According to this, it can be stated that complexity of the WF-creation algorithm is $O\left(\sum_{v \in V} |\ell(v)| * \log|\mathcal{L}|\right)$.

9. Possible Improvements of the Algorithm

Although the algorithm from the **Table 2** reduces the number of clones, there are some drawbacks to it. The **Fig. 8** shows that the two rightmost branches with labels *B_1_2* and *B_1_3* are almost the same, except for the elements *C_6* and *C_7*. The presented WF-model can be improved by deferring the parallel split just before the elements *C_6* and *C_7*. In that case all previous elements can be added to the common branch. Hence, the possible improvement consists of merging the parts of parallel branches that share the same parent parallel activity. In our example this means that branches assigned to labels 1.2 and 1.3 are the same (i.e. 1.3 can be replaced with 1.2) for all vertices before the vertex 10 (including vertex 10). Therefore, for the labels in those vertices, starting pattern 1.3 could be replaced with 1.2. For all vertices after the vertex 10 label 1.2 should be replaced with 1.2.1 and 1.3 should be replaced with 1.2.2.

It turns out that the more “regular” a graph is (in terms of adequately paired and-splits and and-joins) the surplus of the clones becomes more obvious. By looking at the **Fig. 11** one can notice that after the label reduction, label set of the vertex 3 has been reduced from {1.1, 1.2} to {1}, but the same thing does not apply to the vertices 1 and 2. Their label sets can be reduced by just replacing starting patterns of the labels which will, in turn, produce the graph shown in the **Fig. 12**.

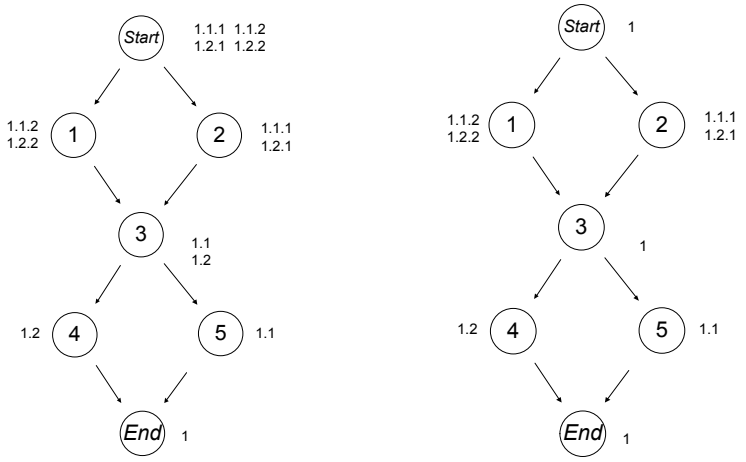


Fig. 11. A graph after the vertex labeling and the same graph after the label reduction algorithm

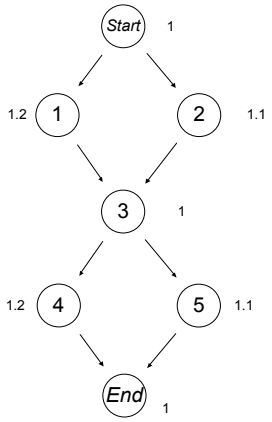


Fig. 12. Optimal labeling solution for the graph on the left hand side in the Fig. 11

Proof of these claims is based on a proof of the Theorem 8 which deals with an improvement that can be done for these vertices where the label set has not been reduced but contains two or more labels that have same parent as the vertex 10 in the Fig. 7. Replacing more labels with the same parent

starts from the vertex containing those labels and propagates replacements upwards to the *Start* vertex or until the first vertex without these labels.

Theorem 8. Let the label set of a vertex v contains a subset O={$l_1, \ldots, l_n$} of labels that have same parent and let l_1 be the first among them in the lexicographic order. If the starting pattern l_iX is replaced with l_1X for the labels of the vertex v and the labels of v„s predecessors and the starting pattern l_j is replaced with $l_i.k$ in the labels of v„s successors, where k is the position of l_i inside the set O, then it can be claimed the line of execution has been kept.

Proof: Without the loss of generality it can be claimed that the vertex v contains two labels *l.i* i *l.j*, where *l* is the mutual prefix of the labels and i<j. All predecessors of v, among other labels received from their other successors, have labels with the patterns *l.i.suffix* i *l.j.suffix*, where suffix is a string of dots and numbers that were concatenated during the labeling algorithm. (Optionally, v can have other labels that do not start with *l* and, for those labels, in v„s predecessors will exist labels, with an appended suffix, that are the successors of those labels so v is in the same line of execution with its predecessors). By replacing prefix *l.j* with *l.i* in the vertex v and in its predecessors, it is possible that a label set of a particular vertex gets reduced, but line of execution will be kept because for each label of the vertex v there exists a label in his predecessors in an ancestor-successor relationship with label from the vertex v. It remains to be shown that replacing prefix *l.i* with *l.i.1* and *l.j* with *l.i.2* in all labels in a v„s successor keeps the line of execution. The vertex v has labels *l.i* and *l.j* which means that at least one successor of the vertex v containing label *l* exists. That successor has at least one more predecessor and that predecessor contains label *l.k* (in any other situation the label reduction algorithm would reduce labels *l.i* i *l.j* with label *l* in the vertex v). The replacement of *l.i* with *l.i.1* and *l.j* with *l.i.2* implies that branches assigned to labels *l.i.1* i *l.i.2* join one step before the join with a branch assigned to *l.k* (and any other branches).

If set O had more than two labels, the proof is similar.

■

Although this improvement can reduce the number of the labels, it significantly raises complexity as it adds labels of the higher value of a level function after the level has already been processed. Appearance of those labels requires algorithm reset and several re-runs.

10. Conclusion

Dividing prerequisite relationships into partially defined components helps maintaining a relationship between elements and increases readability. Nevertheless, as shown in the introductory example, even for simple models it may be impossible to transform the integrated graph directly into a real workflow model with an existing workflow modeling language because such

workflow models do not have to be structured. As an opposite to developing proprietary workflow management software to support unstructured models, an approach consisting of using existing workflow management software is proposed. The approach consists of element cloning (duplication) and a workflow wrapper ensuring the clones are shown as unique elements in the runtime and no data duplication occurs. The description of the wrapper is out of the scope of this paper.

The paper presented the steps for the integration of prerequisite relationships into directed graphs and algorithms for vertex labeling and label reduction. Initial algorithms and enhancements are presented and their correctness has been proved. Furthermore, the complexity of the algorithms has been discussed and the worst case complexity has been given, with the remark that finding average complexities would require extensive tests since the worst case in one of the algorithm"s steps can significantly reduce complexity of another step.

References

1. van der Aalst, W., van Hee, K.: Workflow management. Models, Methods and Systems. MIT Press, Cambridge, Massachusetts, USA. (2004)
2. van der Aalst, W., Lassen, K.: Translating Unstructured Workflow Processes to Readable BPEL: Theory and Implementation. Information and Software Technology, Vol. 50, No. 3, 131-159. (2008)
3. Aho, A.V., Garey, M.R., Ullman, J.D.: The Transitive Reduction of a Directed Graph. SIAM Journal on Computing, Vol.1, No 2, 131-137. (1972)
4. Cesarini, M., Monga, M., Tedesco, R.: Carrying on the e-Learning process with a Workflow Management Engine. In Proceedings of the 2004 ACM Symposium on Applied Computing. Nicosia, Cyprus, 940-945. (2004)
5. Gruhn, V., Laue, R.: Good and bad excuses for unstructured business process models. In Proceedings of 12th European Conference on Pattern Languages of Programs, Irsee, Germany. (2007)
6. Haeupler, B., Telikepalli, K.: Mathew, R.; Siddhartha S.; Tarjan, R.E.: Incremental Cycle Detection, Topological Ordering, and Strong Component Maintenance. (2008). [Online]. Available http://people.csail.mit.edu/haeupler/incremental-topological-ordering-journal.pdf (October 2010)
7. Hausser, R., Friess, M., Küster, J.M., Vanhatalo, J.: Combining Analysis of Unstructured Workflows with Transformation to Structured Workflows. In Proceedings of the 10th IEEE International Enterprise Distributed Object Computing Conference, Hong Kong, China, 129-140. (2006)
8. Hlaoui, Y.B., Ayed, L.J.B.: An Interactive Composition of UML-AD for the Modelling of Workflow Applications. Ubiquitous Computing and Communication Journal, Vol. 4, No. 3, 599-608. (2009)
9. Holl, A., Valentin G.: Structured business process modeling (SBPM). Information Systems Research in Scandinavia (IRIS 27), Falkenberg, Sweden. (CD-ROM). (2004)
10. Johnson, D.B.: Finding all the elementary circuits of a directed graph. SIAM Journal on Computing, Vol. 4, No. 1, 77-84. (1975)

11. Kiepuszewski, B., ter Hofstede, A., Bussler, C.: On Structured Workflow Modelling. Lecture Notes in Computer Science, Vol. 1789. Springer-Verlag, Berlin Heidelberg, 431-445. (2000)
12. Kiepuszewski, B.: Expressiveness and Suitability of Languages for Control Flow Modelling in Workflows. PhD Thesis, Queensland University of Technology, Brisbane, Australia. (2003)
13. Lassen, K., van der Aalst, W.: WorkflowNet2BPEL4WS: A Tool for Translating Unstructured Workflow Processes to Readable BPEL. Lecture Notes in Computer Science, Vol. 4275. Springer-Verlag, Berlin Heidelberg, 127-144. (2006)
14. Lin, L., Ho, C., Sadiq, W., Orlowska, M.E.: Using Workflow Technology to Manage Flexible e-Learning Services. Educational Technology & Society, Vol. 5, No. 4, 116-123. (2002)
15. Liu, R., Kumar, A.: An analysis and taxonomy of unstructured workflows. Lecture Notes in Computer Science, Vol. 3649. Springer-Verlag, Berlin Heidelberg, 268-284. (2005)
16. Mendling, J., Simon, C.: Business Process Design by View Integration. Lecture Notes in Computer Science, Vol. 4103. Springer-Verlag, Berlin Heidelberg, 55-64. (2006)
17. Microsoft Patterns and Practices: Microsoft Application Architecture Guide, 2nd Edition, Microsoft Press, Redmond, USA. (2009)
18. Milašinović, B., Fertalj, K., Nižetić, I.: On some Problems while Writing an Engine for Flow Control in Workflow Management Software. In Proceedings of the 29th International Conference on Information Technology Interfaces, Cavtat, Croatia, 489-494. (2007)
19. Milašinović, B., Fertalj, K.: Using partially defined workflows for course modelling in a learning management system. ICIT-2009, The 4th International Conference on Information Technology, Amman, Jordan. (2009).
20. Nishizawa, K.: A method to Find Elements of Cycles in an Incomplete Directed Graph and Its Applications – Binary AHP and Petri Nets. Computers & Mathematics with Applications, Vol. 33, No. 9, 33-46. (1997)
21. Polyvyanyy, A., García-Bañuelos, L., Dumas, M.: Structuring Acyclic Process Models. Lecture Notes in Computer Science, Vol. 6336, Springer-Verlag, Berlin Heidelberg 276-293. (2010)
22. Sadiq, S., Sadiq, W., Orlowska, M.: Workflow Driven e-Learning – Beyond Collaborative Environments. International NAISO Congress on Networked Learning in a Global Environment, Challenges and Solutions for Virtual Education. Berlin, Germany. (2002)
23. Windows Workflow Foundation. [Online]. Available: http://msdn.microsoft.com/en-us/netframework/aa663328.aspx (October 2010)
24. Workflow Management Coalition. Terminology & Glossary: Document Number WFMC-TC-1011 - Document Status - Issue 3.0 [Online]. Available: http://www.wfmc.org/Download-document/WFMC-TC-1011-Ver-3-Terminology-and-Glossary-English.html (October 2010)
25. Xi, S., Yong, J.: New Data Integration Workflow Design for e-Learning. Lecture Notes in Computer Science, Vol. 4402, Springer-Verlag, Berlin Heidelberg 699-707. (2007)
26. Ye, Y., Roy, K.: A graph-based synthesis algorithm for AND/XOR networks. In Proceedings of the 34th Annual Design Automation Conference, Anaheim, California, USA, 107-112. (1997)

27. Yong, J.: Workflow-based e-learning platform. In Proceedings of the 9th International Conference on Computer Supported Cooperative Work in Design, Coventry, UK, Vol. 2, 1002-1007. (2005)

Krešimir Fertalj is a full professor at the Department of Applied Computing at the Faculty of Electrical Engineering and Computing, University of Zagreb. Currently he lectures a couple of computing courses on undergraduate, graduate and doctoral studies. His professional and scientific interest is in computer-aided software engineering, complex information systems and in project management. He participated in a number of information system designs, implementations and evaluations. Fertalj is member of ACM, IEEE, PMI, and Croatian Academy of Engineering.

Boris Milašinović is a research assistant at the Department of Applied Computing at the Faculty of Electrical Engineering and Computing, University of Zagreb. He graduated in 2001 at the Department of Mathematics at Faculty of Science, University of Zagreb. He received MSc degree in 2006 and Ph. D. degree in 2010 in Computing at the Faculty of Electrical Engineering and Computing. His main research interests include software development and workflow management. Currently he lectures a couple of undergraduate courses in Computing and he is teaching assistant to one graduate course in Computing.

Received: November 20, 2010; Accepted: August 5, 2011

DOI:10.2298/CSIS110406047V

Improving Program Comprehension by Automatic Metamodel Abstraction

Michal Vagač[1] and Ján Kollár[2]

[1] Department of Informatics, Faculty of Natural Sciences, Matej Bel University
Tajovského 40, 974 01 Banská Bystrica, Slovakia
michal.vagac@gmail.com
[2] Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Košice
Letná 9, 042 00 Košice, Slovakia
Jan.Kollar@tuke.sk

Abstract. The maintenance of a software system represents an important part in its lifetime. In general, each software system is the subject of different kinds of changes. Bug fixes and a new functionality extensions are the most common reasons for a change. Usually, a change is accomplished by source code modifications. To make such a modification, correct understanding the current state of a system is required.
This paper presents the innovative approach to the simplification of program comprehension. Based on the presented method, the affected software system is analysed and metamodel for the selected feature is created. The feature represents functional aspect of a system being the subject of the analysis and change. The main benefit is that by focusing on well known (and precisely described) parts of program implementation, it is possible to create metamodel for implementation parts automatically. The level of metamodel is at a higher level of abstraction than implementation.

Keywords: Aspect-oriented programming, feature location, metalevel architectures, program comprehension, reverse engineering, software change.

1. Introduction

Software systems help us with many everyday tasks. Since none software system is perfect, sooner or later there is demand for a change. Software systems often model a real world situation. That is why changes in real world result in state, in which software system (which was satisfactory in the past) is becoming insufficient. To bring such system to the sufficient state again, it has to be changed. The most common reasons for a change are bug fixes and additions of a new functionality. It is more common to change the existing system than to create a new from scratch. Software system maintenance and evolution consumes up to 80 percent of system's lifetime [16].

To perform a maintenance, it is necessary to understand details related to the requested change correctly. Without sufficient knowledge, it is possible to

break functionality of a system. Therefore a program comprehension is the prerequisite for program maintenance. After understanding the request for a change, the relevant implementation code fragments have to be identified. Just after that these fragments can be safely modified. As a software system is growing, it is more and more difficult to locate code intended for maintenance. This problem raises also with fluctuation of developers.

In this paper, we propose a new architecture for improving a program comprehension, based on utilization of well known classes used in object-oriented program. We claim that for these classes it is possible to define accurate localization algorithm. This algorithm, in combination with a predefined feature knowledge base, maps the selected program feature at a higher level of abstraction automatically. Moreover, this mapping is performed in runtime.

The paper is organized as follows. Section 2 describes the proposed methodology for automatic abstraction of selected feature. Section 3 presents the experiment developed using Java programming language. Section 4 covers the related works. Finally, section 5 concludes the paper and presents future work.

2. Feature Metamodel at Higher Level of Abstraction

Existing tools for dynamic analysis and visualization of system execution are mostly generic solutions. Visualization result is difficult to use, since it is closer to implementation level than to application domain level. To get application domain specific visualization, link between application domain concept and code implementation must exist. This link between different levels of abstraction is not explicitly represented in a program code. When reading and understanding program by a developer, this link is built up using previous developer's experiences and knowledge (about problem domain, programming techniques, algorithms, data structures, etc.). To reproduce this activity automatically, a kind of *knowledge base* describing link between different levels of abstraction must be established.

In this paper, we will focus on how to abstract systems constructed in object-oriented manner. A program developed in object-oriented language is typically defined by group of classes and their instances – objects. Objects communicate to each other by sending messages. Relationships between classes and objects are defined by program code. Let's define V as a set of all existing classes (1) and P as a set of classes used in object-oriented program (2). Figure 1 represents a program created in object-oriented language.

$$V = \{v_1, v_2, \ldots, v_n\} \tag{1}$$

$$P = \{p_1, p_2, \ldots, p_m\}, P \subset V \tag{2}$$

To understand object-oriented program, a developer has to read used classes and understand their relations and meanings. By term understanding we mean *"ability of explaining the program, its structure, its behaviour, its effects on its operational context, and its relationships to application domain in terms that are*

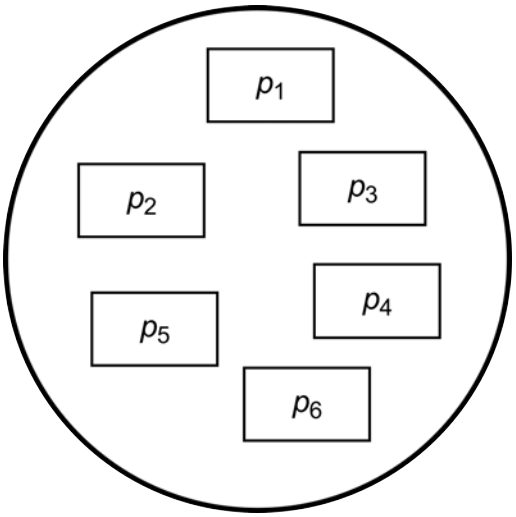


Fig. 1. Object-oriented program using classes p_i

qualitatively different from the tokens used to construct the source code of the program" [2]. It is essential to move implementation level knowledge to a higher level of abstraction. As mentioned above, to make the translation from one level to another, extensive knowledge – a *knowledge base* – is required. Complete knowledge base should contain information about all classes, all application domains and about their relationship. Since usually new classes are defined, it is impossible to prepare complete knowledge base.

However, there is a group of software components, which are well known and their amount is limited – components defined in software libraries. Defining knowledge base which contains information about software libraries, it is possible automatically create the transformation between the implementation level and higher level of abstraction, i.e. such that implements features comprised in libraries.

Modern object oriented software development platforms provides a comprehensive set of its own standard class libraries. Let's designate K as a set of *known classes*, i.e. all classes with known names and meaning. K is a subset of set V, of all classes. (3). Let's define K' as a set of known classes, which are used in the program (4).

$$K = \{k_1, k_2, \ldots, k_r\}, K \subset V \tag{3}$$

$$K' = \{k'_1, k'_2, \ldots, k'_s\}, K' = P \cap K \tag{4}$$

When using classes from standard library, the program becomes easier readable and understandable (Fig. 2).

Let's define *aspect of the program* as a group of known classes used in the program, which relates to one logical part (from higher level abstraction point of

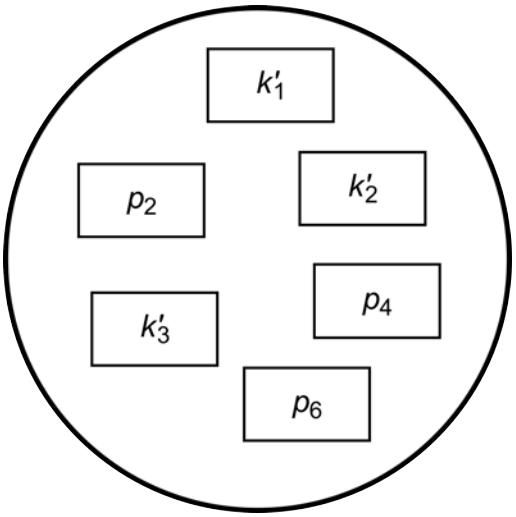


Fig. 2. Object-oriented program using classes p_i and well known classes k_i

view). Aspect of the program is a subset of program's features, since it contains only features based on known classes. Since the program may contain several aspects, A is subset of K' (5). By defining a knowledge base for known classes and by analyzing program for the way of using these classes, it is possible to define a function f, which maps an aspect of the program to model M, which can be created at a higher level of abstraction than implementation level (Fig. 3, equation 6).

$$A = \{a_1, a_2, \ldots, a_t\}, A \subset K' \tag{5}$$

$$M = f(A) \tag{6}$$

In implementation, the knowledge base and the function f will be presented by combination of tracking algorithm and metamodel builder algorithm. Both algorithms will be specific for each feature.

Proposed system architecture is described in fig. 4. Since there are two separate systems (base system and tool), where one react about the other, it is convenient to use a metalevel architecture. Base level is represented by a legacy software system. This system is a subject to analyse. Metalevel represents the tool. Base level system is automatically enhanced with code tracing system runtime execution. Based on tracing results and utilizing knowledge base, the metasystem automatically builds a metamodel of specified feature. This metamodel describes feature implementation at a higher level of abstraction.

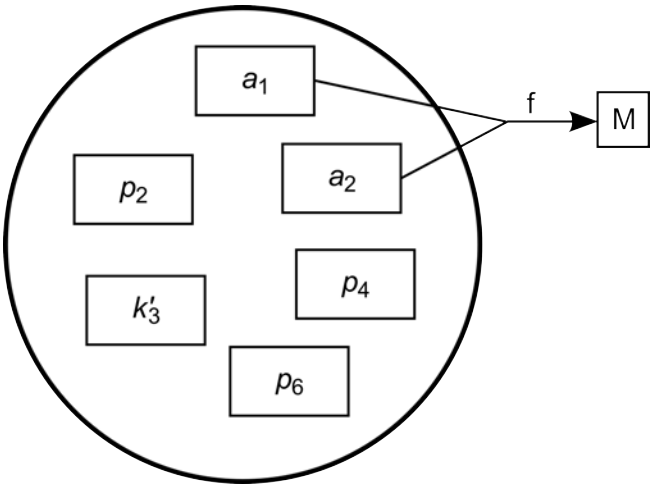


Fig. 3. Object-oriented program using classes p_i and known classes k_i and a_i. By examining the way of use of classes a_i it is possible to create model M at a higher level of abstraction

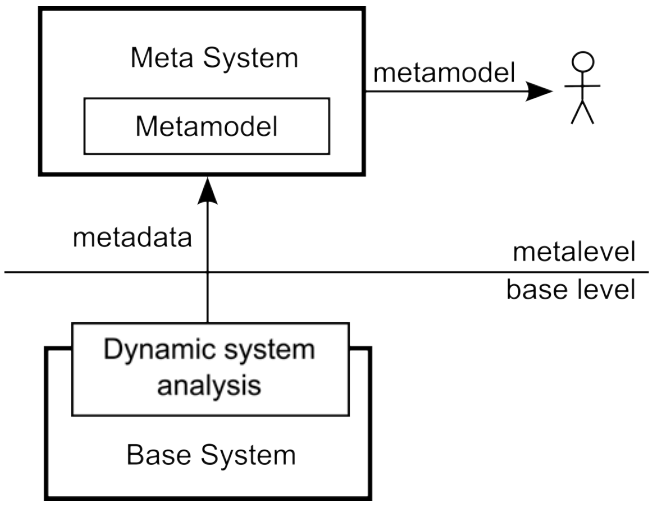


Fig. 4. Metamodel creation using aspect-oriented approach

3. Tool for Feature Metamodel Building

To validate usefulness of our proposal, we have constructed experimental tool, which automatically builds up metamodel for several predefined features. The experiment is performed using Java programming language. This language was chosen because of its wide use and its rich standard library (Java Class Library).

To extend base level application with tracking code, we decided to use aspect-oriented approach. AOP ability for extending existing code with a new functionality is very helpful for techniques depending on metadata – it is possible to extend base system with monitoring code [25]. This new code will create execution traces used to build metamodel.

Process of extending base level application with monitoring code is defined in knowledge base in form of AOP aspects. For each feature, the knowledge base contains information about its implementation – classes and methods used, as well as the way of using these classes and methods. Aspects defined in the knowledge base trace execution and usage of specified classes and methods at the base level of application. According to the traced information, algorithm (also defined in the knowledge base) builds up corresponding metamodel. Aspects and algorithm building metamodel are specific for each modelled feature.

One scenario supported by the experiment is modelling the feature for input/output streams. A stream can be defined as a sequence of data. A stream can represent many different kinds of sources and destinations, including disk files, devices, other programs, and memory arrays. Basic classes used to work with streams are abstract classes *java.io.InputStream*, *java.io.OutputStream*, that work with bytes, and *java.io.Reader* and *java.io.Writer*, that work on characters. Different extensions of these classes allow different kinds of stream transformations. Among others, class *java.io.ByteArrayInputStream* allows reading bytes from byte array, *java.io.FileInputStream* allows reading bytes from file, *java.io.FilterInputStream* allows stream transformations,
java.util.zip.GZIPInputStream decompresses stream data, etc. In the knowledge base used in experiment, the description of these classes is defined. One recognized way of using these classes is chaining their instances (instance of one class is used as constructor parameter for another one). This is common way for defining the chain of classes used to process the specific stream. For example, stream is read from a file, then it is decrypted and finally decompressed.

Described feature of input/output streams is modelled as the chain of filters used with a stream (Fig. 5).

Fig. 5. Metamodel for input/output streams.

To build such metamodel automatically, all invocations of mentioned classes' constructors must be traced. Metamodel is created according to traced sequence of constructor invocations. Algorithm building metamodel must consider only related invocations (connected through constructor argument). Finally, the tool displays graphical representation of created metamodel (Fig. 6).

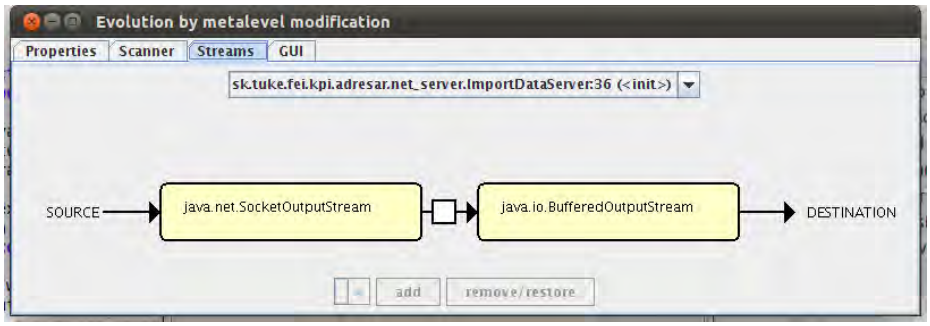


Fig. 6. Data streams metamodel

4. Related Works

The work presented in this paper addresses the issue for improving program comprehension by automatic metamodel abstraction.

According to a new design principle of *open implementation* a software module allows its clients to control its implementation strategy [13].

Open implementation contains besides main interface (providing functionality) also meta-interface through which a client tunes the implementation underlying the primary interface. The wide use of open implementation principle is *metalevel architecture*, see [15].

In general, a metalevel architecture consists of different levels, where one level is controlled by another one. From viewpoint of program represented as a set of objects, it is possible to define several terms in the area of metalevel architectures. Application describing a problem being solved is located at *domain level*. *Domain objects* are objects of this application. These objects describe the problem being solved. *Domain object protocol* defines operations (called *domain operations*) provided by domain object.

Except a domain level there exists a *metalevel*, which provides a space for metaobjects. *Metaobjects* describe, control, implement or modify domain objects. In case of multilevel architecture, metaobject can control another metaobjects. A *metaobject protocol* (MOP) is object-oriented interface allowing communication between objects at domain level and objects at metalevel. It defines application programming interface which can be used to work with metaobjects. Finally, *metaobject operation* is operation from metaobject protocol.

The technology of aspect-oriented programming which allows modularization of *crosscuting concerns* [14], ability for adding a new functionality to an existing program code [25] has been used in our solution as the basic technology for our runtime abstraction.

For maintenance tasks, the first step to be done before a change itself is identifying and understanding program code relevant to the requested change. Identifying parts of source code, that correspond to functional part of program, is known as feature or concept location [32]. *Feature* or *concept* can be defined as cohesive set of functionality [31]. Each feature represents a well understood abstraction of a system's problem domain [28]. It exists at runtime as a collaboration of objects, exchanging messages to achieve a specific goal. The main difference between concepts and features is, that the user can exercise the latter (hence the notion of concept is more general than the notion of feature) [18]. In this work, we focused on features. Features are usually described by the requirements of a software system. Typically, users formulate their requirements, change requests or error reports in terms of features [21].

Finding relations between features and source code takes important part in our program comprehension. Feature location is the process of identifying mappings between domain level concepts and their implementations in source code (it identifies code fragments that implements specific feature) [24]. The maintenance request, expressed usually in natural language and using the domain level terminology is the input of mapping and a set of components that implement the feature [4] is the output. The input and output of the location process belong to different levels of abstraction (domain level on one side, implementation level on the other side). To make the transformation from one level to another, extensive knowledge is required (problem domain, programming techniques, algorithms, data structures, etc.). Since features are not explicitly represented in source code, the features identification is a difficult task. It is traditionally an intuitive and informal process, based on past experiences.

In general, feature location can be static, dynamic or hybrid. The static techniques of feature identification doesn't need execution of subject program – these techniques instead focus on searching in source code. The feature (or concept) location process can be defined as follows [19]:

1. feature formulation (usually in natural language)
2. query formulation and execution based on the intermediary representation
3. investigation of results

First approaches simply utilized pattern matching tools (such as Unix utility grep). This basic principle was improved in several ways ([4] [7] [20] [33] [29] [10] [11]). Improvements include extending the search process with usage of advanced information retrieval methods, or creating different ways of graphical visualisation of query results.

The dynamic technique of feature identification analyzes execution traces of subject program. These techniques are important especially for object-oriented and dynamic languages. With these, it is nearly impossible to obtain a complete

understanding of the system only by inspecting the source code [28]. Object-oriented characteristics such as inheritance and polymorphism make it difficult to understand runtime behavior of the system only by inspecting source code. Finally, hybrid (combined) feature location techniques utilize both - execution trace and source code information. Usually, static information is used to filter the execution traces.

Works [23] [30] [17] use Java Debug Interface to collect execution traces. Using this approach is complicated and leads to performance penalty – so we decided for the use of AOP to create execution traces. In [8] a set of instrumentation aspects is defined that add code to a given Java program to collect enough tracing information such that the program can be reverse engineered. In this approach, every method invocation is intercepted and may be recorded. Since recording each method invocation would yield too much data to be analyzed, the recorder may be customized using a filter expression. However, to define filter expression, implementation level knowledge is required.

Common approach to handle a big amount of execution trace data is its visualization. Standards provided by execution trace analysis are sequence diagrams and mapping method calls to source files. Sequence diagrams, while originally devised as a notation used during analysis and design, can be also very useful in program comprehension – through the visualization of execution call traces. Reverse-engineered sequence diagrams based on dynamic call traces are typically very large, therefore several techniques to handle theirs size were introduced [1].

Works [12] [3] use aspect-oriented programming to instrument executed program with a new code, which collects information about program execution. Collected data are presented as UML sequential diagrams. Paper [22] summarizes experiences with the development of a reverse engineering tool for UML sequence diagrams. Author states that the development of a tool supporting the reconstruction of the behavior of a running software system must address the major areas of data collection, representation of this data in a suitable metamodel, and finally its graphical representation. Work [1] provides an overview of sequence diagram tools.

Paper [9] proposes a reverse-engineering tool suite to build precise class diagrams from Java programs. The tool uses both static and dynamic data to infer relationships among classes and interfaces. Work [3] describes class diagrams used to create metamodels. These metamodels holds collected data and model resulting diagrams.

Besides UML diagrams, several tools uses own specific way of data visualization. Paper [5] proposes trace visualization techniques based on the massive sequence and circular bundle view. Paper [6] introduces a novel 3D visualization technique that supports animation of feature behavior. The approach exploits a third dimension to visually represent the dynamic information, namely object instantiations and message sends. Work [26] describes user views of the execution that are specific to the program being understood and to the particular problem.

Described ways of dynamic analysis and visualization of system execution are generic solutions. Such approaches are often difficult to use, since it is impossible to have only one simple general visualization suitable for different kinds of specific behaviours. Therefore to use such tools, user must understand (often complex) visualization output, which is closer to implementation level than to application domain level. As stated in [26] and [27], understanding the software behavior is a unique problem requiring a specialized solution and a visualization.

Our approach comes out from the fact introduced above. One of our aims was to provide feature specific visualization, which will be closer to application domain level than to implementation level. To create such a visualization, feature specific model is built up during dynamic analysis of system execution.

5. Conclusions and Future Work

The paper presents a new approach to automatic feature mapping between different levels of abstraction. The proposed method utilizes knowledge of standard class libraries. The experimental tool based on this method has been developed, to prove the ability for automatic feature visualization at a higher level of abstraction than that of the implementation. Based on the usage of known classes, the tool is able to automatically track down selected predefined feature of existing application and create its metamodel. Since the metamodel is specific for tracked feature, also its visualization can be closer to application domain and better describes given problem (in contrary to generic visualizations). Predefined knowledge base substitutes developer's knowledge.

The development of such automatic tool is not so simple. The main difficulty is associated with knowledge base construction – for each recognized feature there must be aspects defined to trace feature implementation and algorithms to model traced implementation details in metamodel. The level of difficulty of aspects and algorithms depends on the specific feature. On the other side, once the tool is developed, it can be used to locate predefined features from any existing application using compatible version of libraries. It is even possible to use the tool with application with no source code available (in case of using load-time weaving of aspects). The prerequisite is just the compatibility of application programming interface.

Acknowledgments. This work was supported by VEGA Grant No. 1/0015/10 Principles and methods of semantic enrichment and adaptation of knowledge-based languages for automatic software development.

References

1. Bennett, C., Myers, D., Storey, M.A., German, D.M., Ouellet, D., Salois, M., Charland, P.: A survey and evaluation of tool features for understanding reverse-engineered sequence diagrams. J. Softw. Maint. Evol. 20, 291–315 (July 2008), http://portal.acm.org/citation.cfm?id=1400155.1400156

2. Biggerstaff, T.J., Mitbander, B.G., Webster, D.E.: Program understanding and the concept assignment problem. Commun. ACM 37, 72–82 (May 1994), http://doi.acm.org/10.1145/175290.175300
3. Briand, L.C., Labiche, Y., Leduc, J.: Toward the reverse engineering of uml sequence diagrams for distributed java software. IEEE Trans. Softw. Eng. 32, 642–663 (September 2006), http://portal.acm.org/citation.cfm?id=1248725.1248762
4. Chen, K., Rajlich, V.: Case study of feature location using dependence graph. In: Proceedings of the 8th International Workshop on Program Comprehension. pp. 241–. IWPC '00, IEEE Computer Society, Washington, DC, USA (2000), http://portal.acm.org/citation.cfm?id=518049.856972
5. Cornelissen, B., Zaidman, A., Holten, D., Moonen, L., van Deursen, A., van Wijk, J.J.: Execution trace analysis through massive sequence and circular bundle views. J. Syst. Softw. 81, 2252–2268 (December 2008), http://portal.acm.org/citation.cfm?id=1454787.1454981
6. Greevy, O., Lanza, M., Wysseier, C.: Visualizing live software systems in 3d. In: Proceedings of the 2006 ACM symposium on Software visualization. pp. 47–56. SoftVis '06, ACM, New York, NY, USA (2006), http://doi.acm.org/10.1145/1148493.1148501
7. Griswold, W.G., Yuan, J.J., Kato, Y.: Exploiting the map metaphor in a tool for software evolution. In: Proceedings of the 23rd International Conference on Software Engineering. pp. 265–274. ICSE '01, IEEE Computer Society, Washington, DC, USA (2001), http://portal.acm.org/citation.cfm?id=381473.381501
8. Gschwind, T., Oberleitner, J.: Improving dynamic data analysis with aspect-oriented programming. In: Proceedings of the Seventh European Conference on Software Maintenance and Reengineering. pp. 259–. IEEE Computer Society, Washington, DC, USA (2003), http://portal.acm.org/citation.cfm?id=872754.873577
9. Guéhéneuc, Y.G.: A reverse engineering tool for precise class diagrams. In: Proceedings of the 2004 conference of the Centre for Advanced Studies on Collaborative research. pp. 28–41. CASCON '04, IBM Press (2004), http://portal.acm.org/citation.cfm?id=1034914.1034917
10. Hill, E.: Developing natural language-based program analyses and tools to expedite software maintenance. In: Companion of the 30th international conference on Software engineering. pp. 1015–1018. ICSE Companion '08, ACM, New York, NY, USA (2008), http://doi.acm.org/10.1145/1370175.1370226
11. Hill, E., Pollock, L., Vijay-Shanker, K.: Automatically capturing source code context of nl-queries for software maintenance and reuse. In: Proceedings of the 31st International Conference on Software Engineering. pp. 232–242. ICSE '09, IEEE Computer Society, Washington, DC, USA (2009), http://dx.doi.org/10.1109/ICSE.2009.5070524
12. Khaled, R., Noble, J., Biddle, R.: Inspectj: program monitoring for visualisation using aspectj. In: Proceedings of the 26th Australasian computer science conference - Volume 16. pp. 359–368. ACSC '03, Australian Computer Society, Inc., Darlinghurst, Australia, Australia (2003), http://portal.acm.org/citation.cfm?id=783106.783147
13. Kiczales, G.: Beyond the black box: Open implementation. IEEE Softw. 13, 8–11 (January 1996), http://portal.acm.org/citation.cfm?id=624611.625543
14. Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C.V., Loingtier, J.M., Irwin, J.: Aspect-oriented programming. In: ECOOP. pp. 220–242 (1997)
15. Kočí, R.: Methods and Tools for Implementation of Open Simulation Systems. Ph.D. thesis, Brno University of Technology, Brno, Czech Republic (2004), http://www.fit.vutbr.cz/research/view_pub.php?id=7613

16. Lehman, M.M., Ramil, J.F., Kahen, G.: A paradigm for the behavioural modelling of software processes using system dynamics. Technical report 2001/8, Imperial College, Department of Computing, London, United Kingdom (September 2001)
17. Leroux, H., Réquilé-Romanczuk, A., Mingins, C.: Jacot: a tool to dynamically visualise the execution of concurrent java programs. In: Proceedings of the 2nd international conference on Principles and practice of programming in Java. pp. 201–206. PPPJ '03, Computer Science Press, Inc., New York, NY, USA (2003), http://portal.acm.org/citation.cfm?id=957289.957349
18. Liu, D., Marcus, A., Poshyvanyk, D., Rajlich, V.: Feature location via information retrieval based filtering of a single scenario execution trace. In: Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering. pp. 234–243. ASE '07, ACM, New York, NY, USA (2007), http://doi.acm.org/10.1145/1321631.1321667
19. Marcus, A., Rajlich, V., Buchta, J., Petrenko, M., Sergeyev, A.: Static techniques for concept location in object-oriented code. In: Proceedings of the 13th International Workshop on Program Comprehension. pp. 33–42. IEEE Computer Society, Washington, DC, USA (2005), http://portal.acm.org/citation.cfm?id=1058432.1059343
20. Marcus, A., Sergeyev, A., Rajlich, V., Maletic, J.I.: An information retrieval approach to concept location in source code. In: Proceedings of the 11th Working Conference on Reverse Engineering. pp. 214–223. IEEE Computer Society, Washington, DC, USA (2004), http://portal.acm.org/citation.cfm?id=1038267.1039053
21. Mehta, A., Heineman, G.T.: Evolving legacy systems features using regression test cases and components. In: Proceedings of the 4th International Workshop on Principles of Software Evolution. pp. 190–193. IWPSE '01, ACM, New York, NY, USA (2001), http://doi.acm.org/10.1145/602461.602507
22. Merdes, M., Dorsch, D.: Experiences with the development of a reverse engineering tool for uml sequence diagrams: a case study in modern java development. In: Proceedings of the 4th international symposium on Principles and practice of programming in Java. pp. 125–134. PPPJ '06, ACM, New York, NY, USA (2006), http://doi.acm.org/10.1145/1168054.1168072
23. Oechsle, R., Schmitt, T.: Javavis: Automatic program visualization with object and sequence diagrams using the java debug interface (jdi). In: Revised Lectures on Software Visualization, International Seminar. pp. 176–190. Springer-Verlag, London, UK (2002), http://portal.acm.org/citation.cfm?id=647382.724668
24. Olszak, A., Jørgensen, B.N.: Remodularizing java programs for comprehension of features. In: Proceedings of the First International Workshop on Feature-Oriented Software Development. pp. 19–26. FOSD '09, ACM, New York, NY, USA (2009), http://doi.acm.org/10.1145/1629716.1629722
25. Oriol, M., Cazzola, W., Chiba, S., Saake, G.: Object-oriented technology. ecoop 2008 workshop reader. chap. Getting Farther on Software Evolution via AOP and Reflection, pp. 63–69. RAM-SE'08, Springer-Verlag, Berlin, Heidelberg (2009), http://dx.doi.org/10.1007/978-3-642-02047-6_7
26. Reiss, S.P.: Visualizing program execution using user abstractions. In: Proceedings of the 2006 ACM symposium on Software visualization. pp. 125–134. SoftVis '06, ACM, New York, NY, USA (2006), http://doi.acm.org/10.1145/1148493.1148512
27. Reiss, S.P.: Visual representations of executing programs. J. Vis. Lang. Comput. 18, 126–148 (April 2007), http://portal.acm.org/citation.cfm?id=1230158.1230422
28. Röthlisberger, D., Greevy, O., Nierstrasz, O.: Feature driven browsing. In: Proceedings of the 2007 international conference on Dynamic languages: in conjunction with the 15th International Smalltalk Joint Conference 2007. pp. 79–100. ICDL '07, ACM, New York, NY, USA (2007), http://doi.acm.org/10.1145/1352678.1352684

29. Shepherd, D., Fry, Z.P., Hill, E., Pollock, L., Vijay-Shanker, K.: Using natural language program analysis to locate and understand action-oriented concerns. In: Proceedings of the 6th international conference on Aspect-oriented software development. pp. 212–224. AOSD '07, ACM, New York, NY, USA (2007), http://doi.acm.org/10.1145/1218563.1218587
30. Sundararaman, J., Back, G.: Hdpv: interactive, faithful, in-vivo runtime state visualization for c/c++ and java. In: Proceedings of the 4th ACM symposium on Software visualization. pp. 47–56. SoftVis '08, ACM, New York, NY, USA (2008), http://doi.acm.org/10.1145/1409720.1409729
31. Turner, C.R., Fuggetta, A., Lavazza, L., Wolf, A.L.: A conceptual basis for feature engineering. J. Syst. Softw. 49, 3–15 (December 1999), http://portal.acm.org/citation.cfm?id=340287.351492
32. Wilde, N., Scully, M.C.: Software reconnaissance: mapping program features to code. Journal of Software Maintenance 7, 49–62 (January 1995), http://portal.acm.org/citation.cfm?id=249936.249939
33. Zhao, W., Zhang, L., Liu, Y., Sun, J., Yang, F.: Sniafl: Towards a static noninteractive approach to feature location. ACM Trans. Softw. Eng. Methodol. 15, 195–226 (April 2006), http://doi.acm.org/10.1145/1131421.1131424

Michal Vagač is Assistant Professor at Department of Informatics, Faculty of Natural Sciences, Matej Bel University, and PhD student at Department of Computers and Informatics, Technical university of Košice, Slovakia. He received his MSc. in Computer Science, in 2001. The subject of his research is metamodeling, metaprogramming, programming paradigms, and dynamic software systems adaptation.

Ján Kollár is Full Professor of Informatics at Department of Computers and Informatics, Technical university of Košice, Slovakia. He received his M.Sc. summa cum laude in 1978 and his Ph.D. in Computer Science in 1991. In 1978-1981 he was with the Institute of Electrical Machines in Košice. In 1982-1991 he was with Institute of Computer Science at the P.J. Šafárik University in Košice. Since 1992 he is with the Department of Computer and Informatics at the Technical University of Košice. In 1985 he spent 3 months in the Joint Institute of Nuclear Research in Dubna, USSR. In 1990 he spent 2 months at the Department of Computer Science at Reading University, UK. He was involved in research projects dealing with real-time systems, the design of microprogramming languages, image processing and remote sensing, dataflow systems, implementation of programming languages, and high performance computing. He is the author of process functional programming paradigm. Currently his research area covers formal languages and automata, programming paradigms, implementation of programming languages, functional programming, and adaptive software and language evolution.

Received: April 6, 2011; Accepted: August 17, 2011.

DOI:10.2298/CSIS110318069B

An Approach to Automated Conceptual Database Design Based on the UML Activity Diagram

Drazen Brdjanin and Slavko Maric

University of Banja Luka, Faculty of Electrical Engineering
Patre 5, 78000 Banja Luka, Bosnia and Herzegovina
{bdrazen,ms}@etfbl.net

Abstract. This paper presents an approach to the automated design of the initial conceptual database model. The UML activity diagram, as a frequently used business process modeling notation, is used as the starting point for the automated generation of the UML class diagram representing the conceptual database model. Formal rules for automated generation cover the automatic extraction of business objects and business process participants, as well as the automatic generation of corresponding classes and their associations. Based on these rules we have implemented an automatic generator and evaluated it on a real business model.

Keywords: Activity Diagram, Business Process Model, Class Diagram, Conceptual Database Model, UML, Topcased, ADBdesign.

1. Introduction

The database design process mainly undergoes the following phases: requirements analysis, conceptual design, logical design and physical design. The main goal of conceptual database design is to provide an overall description of data on a high level of abstraction in the entire system. The corresponding model is usually called the *conceptual database model* (CDM) and represents a semantic data model. The related literature is more focused on conceptual design than on other design phases, considering it to be the most important design phase since the following phases are usually straightforward transformations of the model from the previous one. Because of that, the automatization of CDM design has been the subject of research for many years.

Starting with Chen's eleven heuristic rules [10] for the translation of information requirements specified in a natural language (English) into an E-R diagram, a lot of research has been done in the field of *natural language processing* (NLP) on extracting knowledge from requirements specifications and automating CDM design. At present, a natural language is the most frequently used language for requirements specifications and the majority of approaches to automated CDM design are NLP-based approaches. However, the effectiveness and limitations of NLP-based approaches are usually deeply related to the source language since they depend on the size and complexity of the grammar used and the scope of lexicon. Consequently, the utilization of linguistics-based

approaches, which are presently mainly related to the English and German languages, is questionable for other languages with more complex morphology and some non-NLP-based alternatives are more desirable.

Currently, there are several non-NLP-based alternatives to the automated CDM design, such as approaches taking a *collection of forms* [36, 3, 11, 12, 21] or *diagrammatic requirements models* [1] as the basis for automated design instead of requirements specifications expressed in a natural language. There are also several proposals taking *business process models* [15, 19, 32, 31, 7] as the starting basis, but with modest achievements in automated CDM generation. A more detailed overview of the related work is provided later in the paper.

The fact that business modeling and database design often use different notations, which usually don't conform to the same or common metamodel, poses a particular problem and challenge in the automated CDM design based on a business model. Business modeling is usually characterized by process-oriented notations, such as IDEF0 [24], EPC [35], Petri nets [30], BPMN [39], etc. On the other hand, the E-R notation, introduced by Chen [9], or one of its modifications such as IE [22], is traditionally used for conceptual database modeling, while the development of UML has seen an increasing use of the UML class diagram as well [23]. Different, i.e. non-harmonized notations are one of the reasons for a fairly small number of papers in the field. One of the ways to overcome this problem is the use of UML for both business modeling and database design. In this paper we present an approach to the automated design of the *initial CDM*[1] using the unified UML-based notation.

Another reason for the fairly small number of papers in the field, besides different and/or non-harmonized notations, is related to the semantic capacity of business models which has not been completely explored yet and should be exploited for the automated CDM design. Inspired by [15] and some subsequent proposals [4, 31, 37, 7, 6, 5], in this paper we explore the semantic capacity of the business process model represented by the UML activity diagram (AD), and define the formal rules for automated design of the initial CDM represented by the UML class diagram (CD). The proposed approach is based on: (i) automatic extraction of business entities (*participants* and *objects*) from the source AD, and (ii) automatic generation of corresponding classes and their associations (*participant-object* and *object-object* associations) in the target CD.

This paper is structured as follows. After the introduction, the second section presents the AD as the source model, while the third section presents the CD as the target model. The fourth section presents an analysis of the semantic capacity of the AD, and provides the formal rules for automated generation of the initial CDM. The experimental results, including qualitative and quantitative evaluation of the implemented automated CDM generator, are given in the fifth section. The sixth section presents the related work. Finally, we conclude the paper and highlight the directions for further research.

[1] Since the presented approach is currently focused on the automated generation of proper structure of the target data model that could be refined manually afterwards, we use the term *initial CDM*.

2. Detailed Activity Diagram

The AD is a widely accepted business process modeling notation [20] and there are a large number of papers dealing with the formalization of semantics and the suitability analysis of the AD for business modeling [34]. The very rich semantics of the AD allows modeling business processes at different levels of abstraction and detail, from the *macroactivity diagram*, used in developing the initial business model for the rough specification of a business process, to the *detailed activity diagram*, used for the detailed specification of a business process in later stages of business modeling.

In this paper we consider a *detailed activity diagram* (DAD) which represents activities at least at *complete* level (related excerpt from the UML superstructure [28] is shown in Fig. 1), i.e.

- each step in the realization of the given business process is represented by a corresponding *action node* (`OpaqueAction`) and will be shortly referred to as *action* in the rest of the paper;
- each action is performed by a particular business process participant that plays some business role modeled by a corresponding *activity partition* (usually called *swimlane*). In the rest of the paper, a business process participant is shortly referred to as *participant*. It can be *external* (e.g. buyer, customer, etc.) or *internal* (e.g. business worker, working group, organization unit, etc.). Since the responsibility of a particular participant (i.e. corresponding business role) is modeled with one swimlane, we have as many swimlanes as there are different participants involved in the process represented by the given DAD;
- each action may have a number of inputs and/or outputs represented by *object nodes* (`CentralBufferNode`) that can be in different *states* in the given business process. In the rest of the paper they are referred to as *input objects* and *output objects*, respectively;
- objects and actions are connected with *object flows* (`ObjectFlow`). An object flow is a kind of *activity edge* which is directed from an input object toward the corresponding action (*input object flow*) or from an action toward the corresponding output object (*output object flow*); and
- each object flow has a *weight* attribute, whose value is by default one. In UML semantics, the object flow weight represents the minimum number of *tokens* that must traverse the edge at the same time. We assume that the weight of an object flow represents the total number of objects required for an action if they are input objects, or the total number of objects created in an action if they are output objects. An unlimited weight (*) is used if the number of input/output objects is not a constant.

The fact that an action is performed by some participant is represented in both corresponding DAD elements, i.e. the `inPartition` attribute of the corresponding action contains the identifier of the related swimlane, while the `node` attribute of the given swimlane contains the identifiers of all actions performed by the given participant.

The fact that an action has an input object is represented in all three corresponding DAD elements, i.e. the `source` and `target` attributes of the given input object flow contain the identifiers of the given object and action, respectively, while the `outgoing` attribute of the given object and the `incoming` attribute of the given action contain the identifier of the given input object flow.

The fact that an action has an output object is similarly represented, i.e. the `source` and `target` attributes of the given output object flow contain the identifiers of the given action and object, respectively, while the `outgoing` attribute of the given action and the `incoming` attribute of the given object contain the identifier of the given output object flow.

Based on the previous description of business process representation by the AD, the DAD can be formally defined as follows.

Definition 1. ***(Detailed activity diagram)*** *Let $\mathcal{P}, \mathcal{A}, \mathcal{O}$ and $\mathcal{F}$ be sets of participants, actions, objects and object flows in some business process, respectively. The detailed activity diagram, denoted by $\mathcal{DAD}(\mathcal{P}, \mathcal{A}, \mathcal{O}, \mathcal{F})$, is a UML activity diagram with the following properties:*

(1) each action $a \in \mathcal{A}$ is performed by only one participant $p \in \mathcal{P}$, i.e.

$$\forall p \in \mathcal{P}, \forall a \in \mathcal{A} \mid a \in node(p) \quad \Rightarrow \quad inPartition(a) = \{p\};$$

(2) each input object flow $if \in \mathcal{F}_{\mathcal{I}} \subseteq \mathcal{F}$ is an object flow directed from exactly one input object $io \in \mathcal{O}_{\mathcal{I}} \subseteq \mathcal{O}$ toward exactly one action $a \in \mathcal{A}$, i.e.

$$\forall io \in \mathcal{O}_{\mathcal{I}}, \forall a \in \mathcal{A}, \forall if \in \mathcal{F}_{\mathcal{I}} \mid if \in outgoing(io) \wedge if \in incoming(a) \Rightarrow \quad source(if) = io \wedge target(if) = a;$$

(3) each output object flow $of \in \mathcal{F}_{\mathcal{O}} \subseteq \mathcal{F}$ is an object flow directed from exactly one action $a \in \mathcal{A}$ toward exactly one output object $oo \in \mathcal{O}_{\mathcal{O}} \subseteq \mathcal{O}$, i.e.

$$\forall a \in \mathcal{A}, \forall oo \in \mathcal{O}_{\mathcal{O}}, \forall of \in \mathcal{F}_{\mathcal{O}} \mid of \in outgoing(a) \wedge of \in incoming(oo) \Rightarrow \quad source(of) = a \wedge target(of) = oo.$$

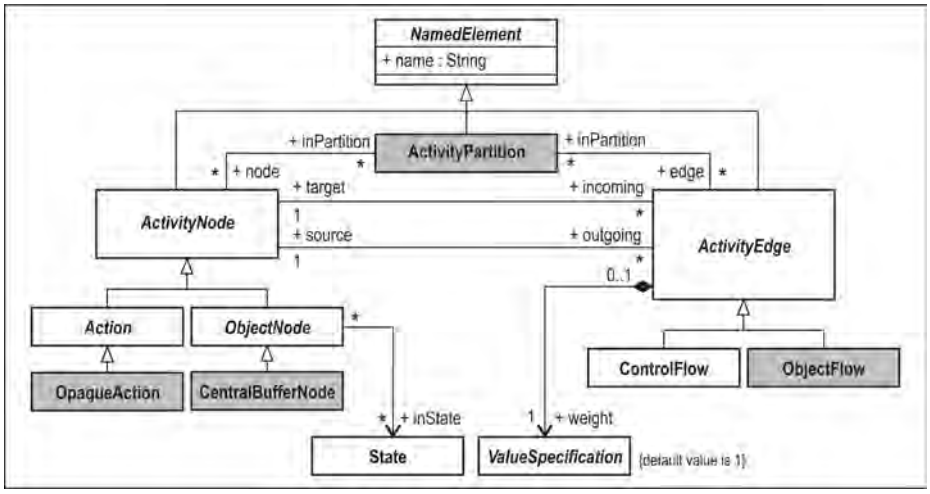


Fig. 1. UML metamodel [28] excerpt used for DAD representation

Figure 2 depicts the sample DAD that will be used throughout this paper as the source business process model. Although it represents a simplified version of the DAD representing the typical business process of order acquisition followed by the delivery of ordered items, it is sufficiently illustrative to cover the most important concepts and basic rules for automated CDM design.

Although the presented workflow is quite intuitive, we still provide a short description. The given business process starts with the customer's request for a new order delivery. Based on that request, the commercial creates a corresponding new order header containing all relevant data and allows the customer to specify order details based on catalog items. After that, the commercial confirms the specified order details and makes the decision. If the order is acceptable, the order header will be marked as accepted. Otherwise, it will be canceled. After the order is accepted, the stockman collects stock items for all confirmed order details. All prepared items will be controlled and packed into a

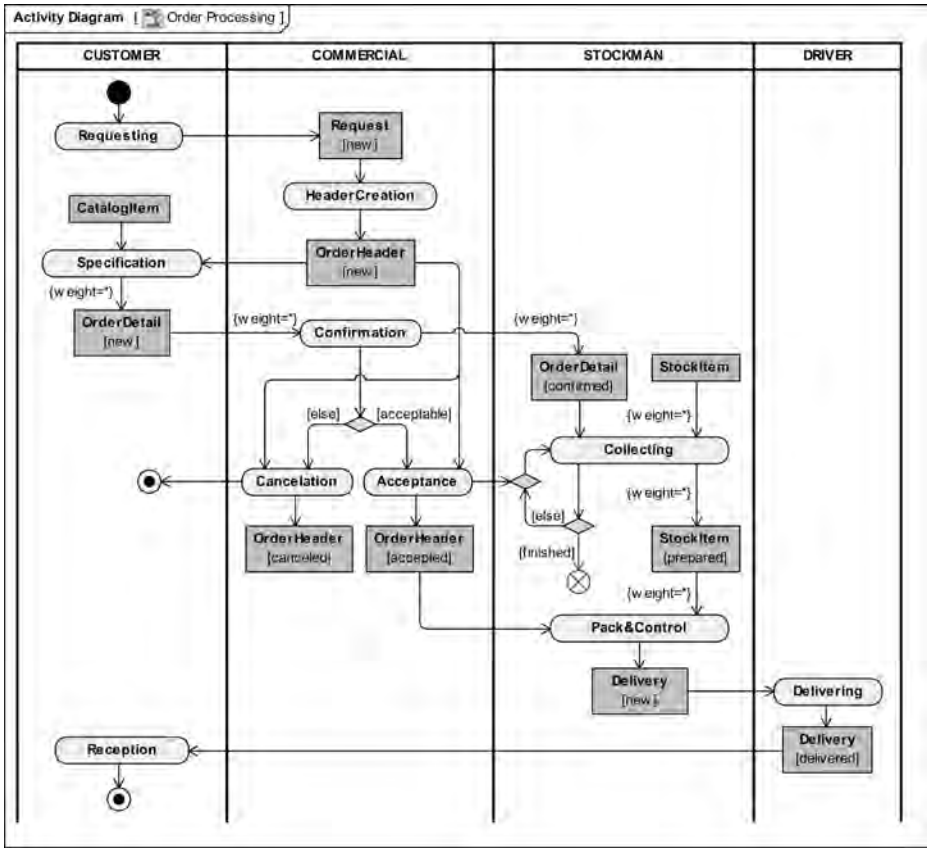


Fig. 2. Sample DAD used in the paper as the source business process model

single delivery and delivered by the driver. Finally, the given business process ends with the reception of the delivery by the corresponding customer.

3. Class Diagram as the Initial CDM

We use the CD to represent the CDM in an attempt to unify notations for both business modeling and CDM design. The UML metamodel excerpt from the UML infrastructure [27], required for the representation of the CDM, is shown in Fig. 3.

Since the target CD is to represent the conceptual model of a relational database, each generated class will not contain operations, but only a set of owned attributes (data members). Since the presented approach is currently focused on automated generation of proper structure of the target model, each generated class will, if necessary, contain only one attribute named `id`, which represents a primary key. Additionally, all generated associations will be binary associations.

Definition 2. ***(Conceptual database model)*** *Let $\mathcal{E}$ and $\mathcal{R}$ be sets of classes and their associations, respectively. The conceptual database model, denoted by $\mathcal{CDM}(\mathcal{E},\mathcal{R})$, is a UML class diagram with the following properties:*

(1) each entity set is modeled by a corresponding class $e \in \mathcal{E}$ of the same name, whose each `ownedAttribute` (`Property`) corresponds to an attribute of the given entity set, and

(2) each relationship is modeled by a corresponding binary association $r \in \mathcal{R}$ of the same name, whose two `memberEnd` attributes (`Property`) represent `source` and `target` association ends with the appropriate multiplicity corresponding to respective relationship cardinalities.

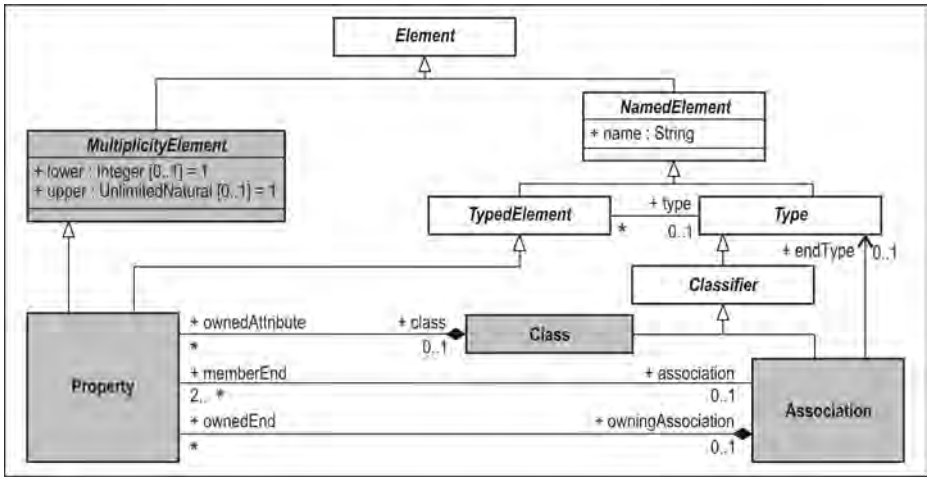


Fig. 3. UML metamodel [27] excerpt for CDM representation

4. Semantic capacity of DAD and rules for CDM design

This section presents an analysis of the semantic capacity of the DAD for automated generation of the CDM. Formal rules cover the extraction of business entities (participants and objects) from the DAD and automated generation of corresponding classes and their associations. At present, we distinguish two groups of associations. The first group are *participant-object* associations generated based on the actions performed by a particular participant on concrete object(s). The second group are *object-object* associations generated for the actions that have both input and output objects.

4.1. Automated extraction of participants

Each business role in the given business process is represented by a corresponding swimlane in the DAD and there are as many swimlanes as there are different types of participants involved in the process. For example, in the sample DAD there are four different roles. The `CUSTOMER` is an external role, while the `COMMERCIAL`, `STOCKMAN` and `DRIVER` are internal roles.

Each role is performed by a particular participant. With time, i.e. with multiple executions of the given business process, the same role may/will be performed by many different participants. External roles are a typical example. In the sample DAD, the `CUSTOMER` is an abstraction of all customers. For example, in one instance of the business process we have one particular customer, but in some other subsequent execution, the customer could be some other person or company. Similarly, the given business system may have many workers/units performing the same internal role. For example, the given business system may have many drivers. In each execution of the sample business process, one of them plays the `DRIVER` role and delivers items to the particular customer. In some other case, some other driver will deliver ordered items to the same or some other customer.

To conclude, each role is an abstraction of a number of entities of the same type. That implies that each role should be represented by the corresponding entity type in the target CDM, i.e. each swimlane (*participant*) from the DAD is to be mapped into the corresponding class of the same name in the target CDM, which is formally expressed by the next rule.

Rule 1. ***(Extraction of participants)*** *Rule $\mathcal{T}_{\mathcal{P}}$ maps participant $p \in \mathcal{P}$ into class $e_{\mathcal{P}} \in \mathcal{E}_{\mathcal{P}} \subseteq \mathcal{E}$:*

$$\mathcal{T}_{\mathcal{P}} : \mathcal{DAD}(\mathcal{P}, \mathcal{A}, \mathcal{O}, \mathcal{F}) \mapsto \mathcal{CDM}(\mathcal{E}, \mathcal{R}) \overset{def}{\Longleftrightarrow} e_{\mathcal{P}} = \mathcal{T}_{\mathcal{P}}\big(p\big)$$
$$\mathcal{T}_{\mathcal{P}}\big(p\big) \overset{def}{=} e_{\mathcal{P}} \mid \big(name(e_{\mathcal{P}}) = name(p)\big).$$

Set $\mathcal{E}_{\mathcal{P}}$ of classes generated for all $p \in \mathcal{P}$ is as follows:

$$\mathcal{E}_{\mathcal{P}} = \Big\{ e_{\mathcal{P}} \in \mathcal{E} \mid e_{\mathcal{P}} = \mathcal{T}_{\mathcal{P}}(p),\ p \in \mathcal{P} \Big\}.$$

Application of the previous rule to the sample DAD is to result in the creation of four classes: `CUSTOMER`, `COMMERCIAL`, `STOCKMAN` and `DRIVER`.

4.2. Automated extraction of objects

During the execution of a business process, participants perform actions and use different objects. We distinguish two categories of objects.

The first category is *generated* objects, i.e. objects created in the given process. For example, the `Request` object is one of generated objects in the sample DAD, since it is created by a particular customer. In each subsequent execution of the given business process, one new request will be created by the same or some other customer. Other generated objects in the sample DAD are: `OrderHeader`, `OrderDetail` and `Delivery`.

After they are created, generated objects may appear to be the input objects in other actions that may use them to create some other generated objects (e.g. an order header is created based on the customer's request) or change their states (e.g. after the specification, each order detail is to be confirmed by the commercial).

The second category is *existing* objects, i.e. objects that are not created in the given process but in some other process. They exist independently of the given business process. Such objects can be used for the creation of generated objects. For example, in the sample DAD, the `CatalogItem` is one type of existing objects which are used for the order details specification by the customer (each order detail is specified based on one catalog item). Existing objects may also be modified (may change their states) by the execution of some action, like the `StockItem` objects which are collected by the stockman and later packed into the delivery.

Based on the previous considerations we can formally define existing and generated objects, since that is important for further DAD analysis.

Definition 3. ***(Existing object)*** *Existing object $eo \in \mathcal{O}_{\mathcal{X}} \subseteq \mathcal{O}$ is an object which is not created in the given process but in some other process, i.e.*

$$\forall eo \in \mathcal{O}_{\mathcal{X}} \quad \Rightarrow \quad \nexists of \in \mathcal{F}_{\mathcal{O}} \mid target(of) = eo,$$

or alternatively

$$eo \in \mathcal{O}_{\mathcal{X}} = \{o \in \mathcal{O} \mid incoming(o) = \emptyset\}.$$

Definition 4. ***(Generated object)*** *Generated object $go \in \mathcal{O}_{\mathcal{G}} \subseteq \mathcal{O}$ is an object created in the given process, that is an object which is not "existing", i.e.*

$$go \in \mathcal{O}_{\mathcal{G}} = \{o \in \mathcal{O} \mid o \notin \mathcal{O}_{\mathcal{X}}\}.$$

In each execution of the given business process, participants deal with one or many (specified by the *weight* attribute of the corresponding object flow) instances of the specified *objects*. Even in case of manipulation with one single instance, in the course of time, i.e. with multiple executions of the given business process, many instances of particular object type will be used. To conclude, each *object* is an abstraction of many entities of the same type. That implies that each *object* should be represented by the corresponding entity type in the target CDM, i.e. each *generated* and each *existing object* from the DAD is to be mapped into a corresponding class of the same name in the target CDM, which is formally expressed by the next rule.

Rule 2. ***(Extraction of objects)*** *Rule* $\mathcal{T}_{\mathcal{O}}$ *maps object* $o \in \mathcal{O}$ *into class* $e_{\mathcal{O}} \in \mathcal{E}_{\mathcal{O}} \subseteq \mathcal{E}$*:*

$$\mathcal{T}_{\mathcal{O}} : \mathcal{DAD}(\mathcal{P}, \mathcal{A}, \mathcal{O}, \mathcal{F}) \mapsto \mathcal{CDM}(\mathcal{E}, \mathcal{R}) \overset{def}{\Longleftrightarrow} e_{\mathcal{O}} = \mathcal{T}_{\mathcal{O}}\big(o\big)$$
$$\mathcal{T}_{\mathcal{O}}\big(o\big) \overset{def}{=} e_{\mathcal{O}} \mid \big(name(e_{\mathcal{O}}) = name(o)\big).$$

Set $\mathcal{E}_{\mathcal{O}}$ *of classes created for all generated and existing objects* $o \in \mathcal{O}_{\mathcal{G}} \cup \mathcal{O}_{\mathcal{X}}$ *in the given business process is as follows:*

$$\mathcal{E}_{\mathcal{O}} = \Big\{ e_{\mathcal{O}} \in \mathcal{E} \mid e_{\mathcal{O}} = \mathcal{T}_{\mathcal{O}}(o),\ o \in \mathcal{O}_{\mathcal{G}} \cup \mathcal{O}_{\mathcal{X}} \Big\}.$$

Application of the previous rule to the sample DAD is to result in the creation of four classes for generated objects: `Request`, `OrderHeader`, `OrderDetail` and `Delivery`, as well as two classes for existing objects: `CatalogItem` and `StockItem`.

4.3. Automated generation of *participant-object* associations

The first group of associations is *participant-object* associations which can be generated based on the actions that participants perform on objects. There are several typical patterns in the DAD that should be analyzed taking into account the distinction between generated and existing objects.

Participant-generated object **associations.** Firstly, we will consider the associations of classes representing participants and generated objects.

As defined earlier, a generated object is an object which is created in some action performed by a particular participant. In the given business process, some action may result in the creation of one or many generated objects of the same type (Fig. 4a), as specified by the weight attribute of the corresponding output object flow. In the course of time, i.e. with multiple executions of the given business process, the same or some other participant playing the same role will create many generated objects of the same type. Each generated object depends on exactly one participant that performs the given action and creates such object(s). This implies that such action is to be mapped from the DAD into the binary association of classes corresponding to the particular participant and generated object in the target CDM with the "$1:\star$" cardinality. For example, in the sample DAD, some customer requests a new delivery by issuing a new request. With time, the same customer may have many requests and all these requests will be related only to this customer.

The first rule for automated generation of associations between classes corresponding to participants and generated objects defines the mapping of actions that create objects into corresponding associations. Consequently, these associations in the target CDM represent the fact that some objects are created by a particular participant.[2]

[2] Each of these associations may have their own attributes (i.e. timestamp, etc.). Due to the "1:*" cardinality, all these attributes will be attributes of the classes corresponding to the generated objects. Presently, we are focused on the proper structure of the target CDM and automated generation of attributes will be part of our future work.

Rule 3. ***(Creation of objects)*** *Let action* $a \in \mathcal{A}$*, performed by participant* $p \in \mathcal{P}$*, create object* $o \in \mathcal{O}_{\mathcal{G}}$*. Rule* $\mathcal{T}_{\mathcal{PGO}}$ *maps triplet* $\langle p, a, o\rangle$ *into association* $r_{\mathcal{PGO}} \in \mathcal{R}_{\mathcal{PGO}} \subseteq \mathcal{R}$ *between classes* $e_{\mathcal{P}}$ *and* $e_{\mathcal{G}}$ *corresponding to the given participant and generated object, respectively:*

$$\mathcal{T}_{\mathcal{PGO}} : \mathcal{DAD}(\mathcal{P}, \mathcal{A}, \mathcal{O}, \mathcal{F}) \mapsto \mathcal{CDM}(\mathcal{E}, \mathcal{R}) \overset{def}{\Longleftrightarrow} r_{\mathcal{PGO}} = \mathcal{T}_{\mathcal{PGO}}\big(\langle p, a, o\rangle\big)$$

$$\mathcal{T}_{\mathcal{PGO}}\big(\langle p, a, o\rangle\big) \overset{def}{=} r_{\mathcal{PGO}} \mid \Big(name(r_{\mathcal{PGO}}) = name(a) \wedge \big(memberEnd(r_{\mathcal{PGO}}) = \{source, target\} \mid type(source) = e_{\mathcal{P}} \wedge multiplicity(source) = 1 \wedge type(target) = e_{\mathcal{G}} \wedge multiplicity(target) = *\big)\Big).$$

Let $\mathcal{O}_{\mathcal{GO}}(a) \subseteq \mathcal{O}$ *be the set of all generated objects of action* $a \in \mathcal{A}$*. Then set* $\mathcal{R}_{\mathcal{PGO}}(a)$ *of all "participant-generated object" associations, generated for all* $o \in \mathcal{O}_{\mathcal{GO}}(a)$ *for the given action* $a \in \mathcal{A}$*, is as follows:*

$$\mathcal{R}_{\mathcal{PGO}}(a) = \Big\{ r_{\mathcal{PGO}} \in \mathcal{R} \mid r_{\mathcal{PGO}} = \mathcal{T}_{\mathcal{PGO}}(\langle p, a, o\rangle),\ o \in \mathcal{O}_{\mathcal{GO}}(a) \Big\}.$$

Total set $\mathcal{R}_{\mathcal{PGO}}$ *of "participant-generated object" associations representing facts of object creation for the entire DAD represents the union of all* $\mathcal{R}_{\mathcal{PGO}}(a)$ *sets*

$$\mathcal{R}_{\mathcal{PGO}} = \bigcup_{a \in \mathcal{A}} \mathcal{R}_{\mathcal{PGO}}(a).$$

Application of the previous rule to the sample DAD will result in the creation of associations for the following triplets: `<CUSTOMER,Requesting,Request>`, `<COMMERCIAL,HeaderCreation,OrderHeader>`, `<CUSTOMER,Specification,OrderDetail>` and `<STOCKMAN,Pack&Control,Delivery>`.

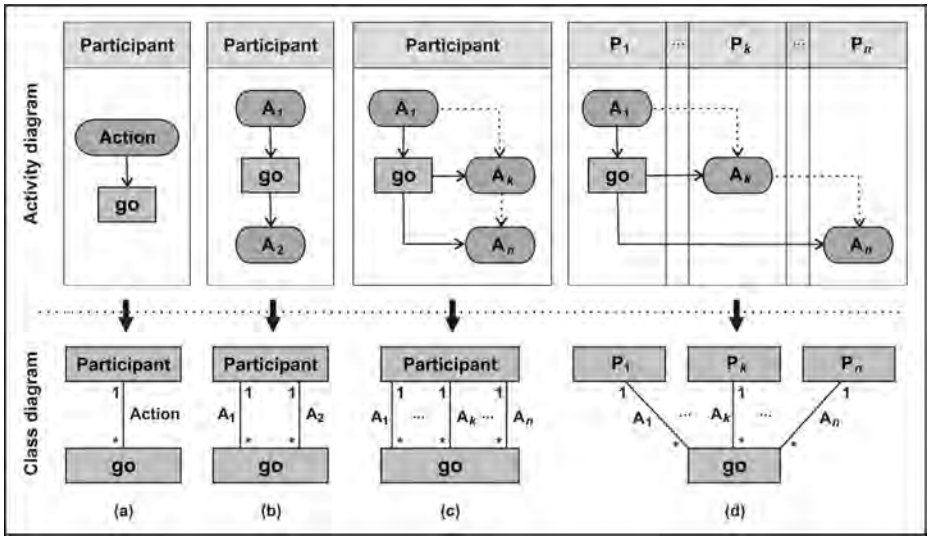


Fig. 4. Typical DAD patterns for generated objects and mapping into CDM

Each generated object may appear to be the input object in some other action performed by the same participant (Fig. 4b). In the course of time, the given participant will perform such action many times and each time on a different input object generated in some previous action. This implies that such action is to be mapped from the DAD into the binary association of classes corresponding to the given participant and given input object in the target CDM with the "$1:*$" cardinality. For example, in the sample DAD, the customer receives delivery. With time, the same customer may receive many deliveries and all these deliveries will be related only to this customer.

More generally, each generated object may constitute the input object in several different actions performed by the same participant (Fig. 4c) or some other participants (Fig. 4d). For example, in the sample DAD, the order header is the input object in several subsequent actions: it is used for specification of order details (performed by the customer), it is the subject of cancelation or acceptance (performed by the commercial), and it is also used for preparing delivery (performed by the stockman). Based on the previous considerations, each of these actions with the same input generated object(s), is to be mapped into the binary association of classes corresponding to the particular participant and given input object(s) in the target CDM with the "$1:*$" cardinality.

The second rule for the automated generation of associations between classes corresponding to participants and generated objects defines the mapping of actions with generated objects as inputs into corresponding associations. Consequently, these associations in the target CDM represent the fact that some generated objects are used by a particular participant.

Rule 4. ***(Usage of generated objects)*** *Let action $a \in \mathcal{A}$, performed by participant $p \in \mathcal{P}$, have generated input object $o \in \mathcal{O}_{\mathcal{G}}$. Rule $\mathcal{T}_{\mathcal{PGI}}$ maps triplet $\langle p, a, o\rangle$ into association $r_{\mathcal{PGI}} \in \mathcal{R}_{\mathcal{PGI}} \subseteq \mathcal{R}$ between classes $e_{\mathcal{P}}$ and $e_{\mathcal{G}}$ corresponding to the given participant and input object, respectively:*

$$\mathcal{T}_{\mathcal{PGI}} : \mathcal{DAD}(\mathcal{P}, \mathcal{A}, \mathcal{O}, \mathcal{F}) \mapsto \mathcal{CDM}(\mathcal{E}, \mathcal{R}) \overset{def}{\Longleftrightarrow} r_{\mathcal{PGI}} = \mathcal{T}_{\mathcal{PGI}}\big(\langle p, a, o\rangle\big)$$

$$\mathcal{T}_{\mathcal{PGI}}\big(\langle p, a, o\rangle\big) \overset{def}{=} r_{\mathcal{PGI}} \mid \Big(name(r_{\mathcal{PGI}}) = name(a) \wedge \big(memberEnd(r_{\mathcal{PGI}}) = \{source, target\} \mid type(source) = e_{\mathcal{P}} \wedge multiplicity(source) = 1 \wedge type(target) = e_{\mathcal{G}} \wedge multiplicity(target) = *\big)\Big).$$

Let $\mathcal{O}_{\mathcal{GI}}(a) \subseteq \mathcal{O}$ be the set of all generated input objects of action $a \in \mathcal{A}$. Then set $\mathcal{R}_{\mathcal{PGI}}(a)$ of all "participant-generated object" associations, generated for all $o \in \mathcal{O}_{\mathcal{GI}}(a)$ for the given action $a \in \mathcal{A}$, is as follows:

$$\mathcal{R}_{\mathcal{PGI}}(a) = \Big\{ r_{\mathcal{PGI}} \in \mathcal{R} \mid r_{\mathcal{PGI}} = \mathcal{T}_{\mathcal{PGI}}(\langle p, a, o\rangle),\ o \in \mathcal{O}_{\mathcal{GI}}(a) \Big\}.$$

Total set $\mathcal{R}_{\mathcal{PGI}}$ of "participant-generated object" associations representing facts of generated objects usages for the entire DAD represents the union of all $\mathcal{R}_{\mathcal{PGI}}(a)$ sets

$$\mathcal{R}_{\mathcal{PGI}} = \bigcup_{a \in \mathcal{A}} \mathcal{R}_{\mathcal{PGI}}(a).$$

Finally, total set $\mathcal{R}_{\mathcal{PG}}$ of "participant-generated object" associations for the entire DAD represents the union of the $\mathcal{R}_{\mathcal{PGO}}$ and $\mathcal{R}_{\mathcal{PGI}}$ sets

$$\mathcal{R}_{\mathcal{PG}} = \mathcal{R}_{\mathcal{PGO}} \cup \mathcal{R}_{\mathcal{PGI}}.$$

Some action having a generated object as an input object may change its state and such object can be used in other subsequent actions. For example, in the sample DAD, the commercial accepts an order header changing its state from `new` to `accepted` and that accepted order header is later used for preparing delivery. It is the same generated object, but in different states. Different states are represented by different attribute values and don't affect the structure of the target CDM. That implies that the previous rule is relevant for all generated input objects regardless of their states[3]. In this way, application of the previous rule to the sample DAD will result in the creation of associations for the following triplets: `<COMMERCIAL,HeaderCreation,Request>`, `<CUSTOMER,Specification,OrderHeader>`, `<COMMERCIAL,Confirmation,OrderDetail>`, `<COMMERCIAL,Cancelation,OrderHeader>`, `<COMMERCIAL,Acceptance,OrderHeader>`, `<STOCKMAN,Collecting,OrderDetail>`, `<STOCKMAN,Pack&Control,OrderHeader>`, `<DRIVER,Delivering,Delivery>` and `<CUSTOMER,Reception,Delivery>`.

***Participant-existing object* associations.** As earlier defined, existing objects are objects which are not created in the given business process, but in some other process. We distinguish three typical usages of existing objects.

Firstly, they may be used as input objects in actions without changing their state, i.e. actions which result in creating generated objects based on existing objects. Such existing object in the sample DAD is the `CatalogItem` used for the specification of order details. The statement that some participant uses some existing object for the creation of some generated object contains three main facts: (i) participant creates some new object, (ii) some new object will be generated based on some existing object, and (iii) participant uses some existing object. Each fact can be represented by a corresponding association. However, these three facts are not mutually independent and just two associations will suffice. The third association will be redundant (if we know that the given customer specified the order details based on catalog items, we indirectly know that (s)he used those catalog items). The first fact has been already covered by the rule for the automated generation of *participant-generated object* associations. The second fact is more significant than the third fact and will be represented by an *object-object* association[4]. Consequently, there is no need for the automated generation of association of classes corresponding to the given participant and such existing object.

[3] It is possible that the state of some generated object is not specified at all. For example, some object is naturally in state `new` after the creation, but that doesn't need to be explicitly specified.

[4] Automated generation of *object-object* associations is the subject of the next section.

The second typical usage of existing objects is the *activation*. It represents the fact that some existing object(s) constitute(s) the input in some action that changes its/their state. This implies that the given action has the input and output objects of the same name. The input object state (initial state of the given existing object(s) before the execution of the given action) is typically not depicted, while the output object state (state of the given object(s) after the action execution) is typically depicted, as illustrated in Fig. 5. Collecting stock items is an example of the existing object activation in the sample DAD. The stockman collects stock items and prepares them for packing and delivering. (S)he takes the items from the stock and changes their state into `prepared`.

The term *activation* is used for two reasons: (i) such objects exist independently of the given business process in some *idle* state before the *activation*, and (ii) typically (but not necessarily), *activation* represents the initial usage of such existing objects in the business process, since the *activated* existing objects, after the *activation*, usually constitute the input objects in subsequent actions.

Definition 5. ***(Existing object activation)*** *Let $\mathcal{O}_{\mathcal{I}}(a)$ and $\mathcal{O}_{\mathcal{O}}(a)$ represent the sets of input and output objects of action $a \in \mathcal{A}$, respectively. Action a, performed by participant $p \in \mathcal{P}$, represents the activation $\alpha(p, a, eo)$ of existing object $eo \in \mathcal{O}_{\mathcal{I}}(a)$ if there is an output object $oo \in \mathcal{O}_{\mathcal{O}}(a)$ such that $name(oo) = name(eo)$.*

By performing an activation, the given participant may activate one or many existing objects of the given type (specified by the weight attribute of the corresponding object flows). With time, the same participant may activate many existing objects. On the other hand, the same existing object(s) is/are activated by one participant in the given business process, but with time, it is possible that the same existing object will be activated many times by many different participants (some book in the library may be borrowed many times by many library members). Consequently, if we consider *activation* as a relationship between participants and existing objects, then its cardinality is "$\star : \star$". Since activation typically has its own attributes (e.g. state attributes, activation timestamp, etc.), we will not represent it as an association whose source and target end multiplicities equal "$\star$", but as a separate class associated with both corresponding classes (participant and object) by two "$\star : 1$" associations (Fig. 5).

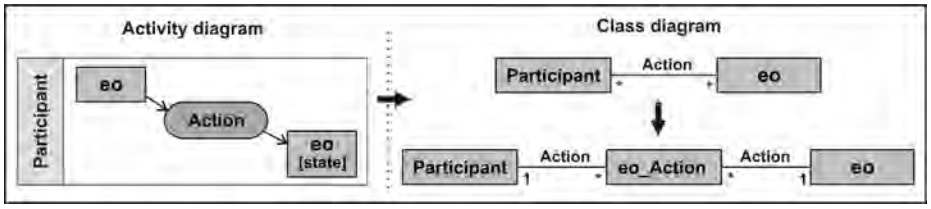


Fig. 5. Existing object activation and corresponding CDM representation

Rule 5. ***(Activation of existing objects)*** *Composite rule $\mathcal{T}_{\mathcal{PEA}}$, consisting of three rules $\mathcal{T}^{(1)}_{\mathcal{PEA}}$, $\mathcal{T}^{(2)}_{\mathcal{PEA}}$ and $\mathcal{T}^{(3)}_{\mathcal{PEA}}$, defines the mapping of activation $\alpha(p,a,eo)$ into a corresponding activation class and its associations.*

Rule $\mathcal{T}^{(1)}_{\mathcal{PEA}}$ maps activation $\alpha(p,a,eo)$ into activation class $e_{\mathcal{A}} \in \mathcal{E}_{\mathcal{A}} \subseteq \mathcal{E}$:

$$\mathcal{T}^{(1)}_{\mathcal{PEA}} : \mathcal{DAD}(\mathcal{P},\mathcal{A},\mathcal{O},\mathcal{F}) \mapsto \mathcal{CDM}(\mathcal{E},\mathcal{R}) \overset{def}{\Longleftrightarrow} e_{\mathcal{A}} = \mathcal{T}^{(1)}_{\mathcal{PEA}}\big(\alpha(p,a,eo)\big)$$
$$\mathcal{T}^{(1)}_{\mathcal{PEA}}\big(\alpha(p,a,eo)\big) \overset{def}{=} e_{\mathcal{A}} \mid \big(name(e_{\mathcal{A}}) = name(eo) + name(a)\big).$$

Rule $\mathcal{T}^{(2)}_{\mathcal{PEA}}$ maps activation $\alpha(p,a,eo)$ into association $r_{\mathcal{PA}} \in \mathcal{R}_{\mathcal{PA}} \subseteq \mathcal{R}$ between classes $e_{\mathcal{P}}$ and $e_{\mathcal{A}}$ corresponding to the given participant and activation, respectively:

$$\mathcal{T}^{(2)}_{\mathcal{PEA}} : \mathcal{DAD}(\mathcal{P},\mathcal{A},\mathcal{O},\mathcal{F}) \mapsto \mathcal{CDM}(\mathcal{E},\mathcal{R}) \overset{def}{\Longleftrightarrow} r_{\mathcal{PA}} = \mathcal{T}^{(2)}_{\mathcal{PEA}}\big(\alpha(p,a,eo)\big)$$
$$\mathcal{T}^{(2)}_{\mathcal{PEA}}\big(\alpha(p,a,eo)\big) \overset{def}{=} r_{\mathcal{PA}} \mid \Big(name(r_{\mathcal{PA}}) = name(a) \wedge \big(memberEnd(r_{\mathcal{PA}}) = \{source, target\} \mid type(source) = e_{\mathcal{P}} \wedge multiplicity(source) = 1 \wedge type(target) = e_{\mathcal{A}} \wedge multiplicity(target) = *\big)\Big).$$

Rule $\mathcal{T}^{(3)}_{\mathcal{PEA}}$ maps activation $\alpha(p,a,eo)$ into association $r_{\mathcal{EA}} \in \mathcal{R}_{\mathcal{EA}} \subseteq \mathcal{R}$ between classes $e_{\mathcal{X}}$ and $e_{\mathcal{A}}$ corresponding to the given existing object and activation, respectively:

$$\mathcal{T}^{(3)}_{\mathcal{PEA}} : \mathcal{DAD}(\mathcal{P},\mathcal{A},\mathcal{O},\mathcal{F}) \mapsto \mathcal{CDM}(\mathcal{E},\mathcal{R}) \overset{def}{\Longleftrightarrow} r_{\mathcal{EA}} = \mathcal{T}^{(3)}_{\mathcal{PEA}}\big(\langle p, \alpha(a,eo)\rangle\big)$$
$$\mathcal{T}^{(3)}_{\mathcal{PEA}}\big(\alpha(p,a,eo)\big) \overset{def}{=} r_{\mathcal{EA}} \mid \Big(name(r_{\mathcal{EA}}) = name(a) \wedge \big(memberEnd(r_{\mathcal{EA}}) = \{source, target\} \mid type(source) = e_{\mathcal{X}} \wedge multiplicity(source) = 1 \wedge type(target) = e_{\mathcal{A}} \wedge multiplicity(target) = *\big)\Big).$$

Let $\mathcal{X}$ be the set of all activations $\alpha(p,a,eo)$ in the given $\mathcal{DAD}(\mathcal{P},\mathcal{A},\mathcal{O},\mathcal{F})$. The $\mathcal{E}_{\mathcal{A}}$, $\mathcal{R}_{\mathcal{PA}}$ and $\mathcal{R}_{\mathcal{EA}}$ sets of classes and corresponding associations, created for all activations $\alpha(p,a,eo) \in \mathcal{X}$, are as follows:

$$\mathcal{E}_{\mathcal{A}} = \Big\{e_{\mathcal{A}} \in \mathcal{E} \mid e_{\mathcal{A}} = \mathcal{T}^{(1)}_{\mathcal{PEA}}\big(\alpha(p,a,eo)\big),\ \alpha(p,a,eo) \in \mathcal{X}\Big\},$$

$$\mathcal{R}_{\mathcal{PA}} = \Big\{r_{\mathcal{PA}} \in \mathcal{R} \mid r_{\mathcal{PA}} = \mathcal{T}^{(2)}_{\mathcal{PEA}}\big(\alpha(p,a,eo)\big),\ \alpha(p,a,eo) \in \mathcal{X}\Big\},$$

$$\mathcal{R}_{\mathcal{EA}} = \Big\{r_{\mathcal{EA}} \in \mathcal{R} \mid r_{\mathcal{EA}} = \mathcal{T}^{(3)}_{\mathcal{PEA}}\big(\alpha(p,a,eo)\big),\ \alpha(p,a,eo) \in \mathcal{X}\Big\}.$$

There is only one activation (`<STOCKMAN,Collecting,StockItem>`) in the sample DAD. Application of the $\mathcal{T}_{\mathcal{PEA}}$ rule will result in the creation of an activation class (`StockItem_Collecting`) as well as two associations with classes corresponding to the given participant (`<STOCKMAN,StockItem_Collecting>`) and existing object (`<StockItem,StockItem_Collecting>`).

The third typical usage of an existing object appears after activation when the activated existing object constitutes the input object in some subsequent actions performed by the same and/or some other participant(s), as illustrated

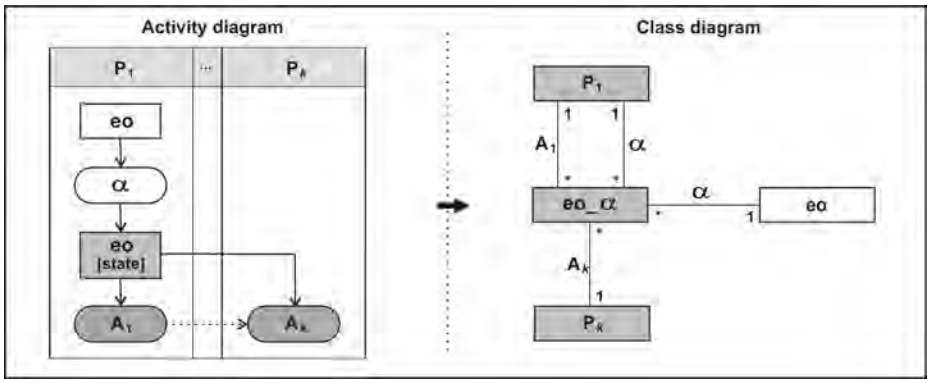


Fig. 6. Usage of activated existing object and mapping into CDM

in Fig. 6. For example (sample DAD), after collecting, stock items are prepared (i.e. activated from the stock) for packing and constitute the input objects in the subsequent `Pack&Control` action.

If an activated existing object constitute the input object in some subsequent action, then the given activated object is related to exactly one participant who performs that action. With time, the given participant will perform such action many times. Each time it may be performed on the same existing object, but also on some other activated existing object of the same type (e.g. the same library member may borrow the same book many times, as well as many different books). This implies that the cardinality of the relationship between the given participant and activated existing object is to be "`1:*`". To be able to define the formal rule that maps such action into the association of corresponding classes, the target class that corresponds to the given activated existing object is to be determined. The given participant performs the action with some concrete existing input object that was previously activated, i.e. (s)he is not related to any object of the given type, but to the particular activated object. This implies that the target class should be the class corresponding to the activation of the given object, but not the class representing all objects of the given type[5].

[5] Associated activation class, representing the fact of activation of existing objects of the given type, is a weak entity type since each activation existentially depends on some concrete existing object. The primary key of this entity set will consist of the existing object identifier (i.e. primary key of the existing object) and some discriminator (e.g. activation timestamp). Each activation also depends on some concrete participant, but not existentially (if we suppose a slightly different business process metamodel that enables models containing only actions and objects, but without participants, activation will depend only on the existing object, i.e there is no activation without some existing object). Since the associated activation class contains complete identity information about the activated objects, as well as a set of other attributes representing the state of the activated objects, we can conclude that the associated activation class constitutes the relevant representation of the activated existing objects.

Consequently, an action representing the usage of activated existing object(s) in the DAD is to be mapped into the binary association of classes corresponding to the particular participant and activation of the given input existing object(s) in the target CDM with the "`1:*`" cardinality (Fig. 6).

Rule 6. ***(Usage of activated existing objects)*** *Let action $a \in \mathcal{A}$, performed by participant $p \in \mathcal{P}$, have activated existing input object $o \in \mathcal{O}_{\mathcal{X}}$. Rule $\mathcal{T}_{\mathcal{PEI}}$ maps triplet $\langle p, a, o \rangle$ into association $r_{\mathcal{PEI}} \in \mathcal{R}_{\mathcal{PEI}} \subseteq \mathcal{R}$ between classes $e_{\mathcal{P}}$ and $e_{\mathcal{A}}$ corresponding to the given participant and activation of the existing input object, respectively:*

$$\mathcal{T}_{\mathcal{PEI}} : \mathcal{DAD}(\mathcal{P}, \mathcal{A}, \mathcal{O}, \mathcal{F}) \mapsto \mathcal{CDM}(\mathcal{E}, \mathcal{R}) \overset{def}{\iff} r_{\mathcal{PEI}} = \mathcal{T}_{\mathcal{PEI}}\big(\langle p, a, o \rangle\big)$$

$$\mathcal{T}_{\mathcal{PEI}}\big(\langle p, a, o \rangle\big) \overset{def}{=} r_{\mathcal{PEI}} \mid \Big(name(r_{\mathcal{PEI}}) = name(a) \wedge \big(memberEnd(r_{\mathcal{PEI}}) = \{source, target\} \mid type(source) = e_{\mathcal{P}} \wedge multiplicity(source) = 1 \wedge type(target) = e_{\mathcal{A}} \wedge multiplicity(target) = *\big)\Big).$$

Let $\mathcal{O}_{\mathcal{A}}(a) \subseteq \mathcal{O}$ be the set of all activated existing input objects of action $a \in \mathcal{A}$. Then set $\mathcal{R}_{\mathcal{PEI}}(a)$ of all "participant-existing object" associations, generated for all $o \in \mathcal{O}_{\mathcal{A}}(a)$ for the given action $a \in \mathcal{A}$, is as follows:

$$\mathcal{R}_{\mathcal{PEI}}(a) = \Big\{ r_{\mathcal{PEI}} \in \mathcal{R} \mid r_{\mathcal{PEI}} = \mathcal{T}_{\mathcal{PEI}}(\langle p, a, o \rangle),\ o \in \mathcal{O}_{\mathcal{A}}(a) \Big\}.$$

Total set $\mathcal{R}_{\mathcal{PEI}}$ of "participant-existing object" associations representing facts of activated existing objects usages, represents the union of all $\mathcal{R}_{\mathcal{PEI}}(a)$ sets

$$\mathcal{R}_{\mathcal{PEI}} = \bigcup_{a \in \mathcal{A}} \mathcal{R}_{\mathcal{PEI}}(a).$$

Total set $\mathcal{R}_{\mathcal{PE}}$ of "participant-existing object" associations represents the union of $\mathcal{R}_{\mathcal{PEA}}$ and $\mathcal{R}_{\mathcal{PEI}}$ sets. Finally, total set $\mathcal{R}_{\mathcal{PO}}$ of "participant-object" associations for the entire DAD represents the union of all "participant-generated object" and "participant-existing object" associations, i.e.

$$\mathcal{R}_{\mathcal{PO}} = \mathcal{R}_{\mathcal{PG}} \cup \mathcal{R}_{\mathcal{PE}} = \mathcal{R}_{\mathcal{PGO}} \cup \mathcal{R}_{\mathcal{PGI}} \cup \mathcal{R}_{\mathcal{PEA}} \cup \mathcal{R}_{\mathcal{PEI}}.$$

The previous rule is relevant for the `<STOCKMAN,Pack&Control,StockItem>` triplet in the sample DAD. Its application will result in the creation of one association in the target CDM between classes `STOCKMAN` and `StockItem_Collecting` that correspond to the given participant and existing object activation.

Some action having an activated existing object as the input object may change its state and such object can be used in other subsequent actions. This case is not depicted in the sample DAD, but it is possible. By dividing the `Pack&Control` action into two separate subsequent actions, `Check` and `Pack`, each prepared stock item will be firstly checked before packing, i.e. action `Check` will change the state of each activated object from `prepared` to `checked`. It is the same activated existing object, but in different states. Since different states are represented by different attribute values, this case doesn't affect the structure of the target CDM and there is no need for additional associations (analogy with generated objects usage).

4.4. Automated generation of *object-object* associations

The second group of associations is *object-object* associations, i.e. associations between classes representing *business objects*. They can be automatically generated based on actions that have both input and output objects by direct mapping of these actions into the respective associations.

Actions with input and output objects may be classified based on the number of different types of input and output objects. There are several possible situations (Fig. 7) and they are as follows: (i) *single input - single output* (SISO) actions, (ii) *multi input - single output* (MISO) actions, (iii) *single input - multi output* (SIMO) actions, and (iv) *multi input - multi output* (MIMO) actions.

In the sample DAD, an example of SISO actions is the `HeaderCreation` action, since it has input object(s) of exactly one type (`Request`) and output object(s) of exactly one type (`OrderHeader`). In the given example, each execution of this action takes exactly one customer's request and creates exactly one order header. However, generally it is possible that some SISO action takes more than one input object of the same type and/or has more than one output object of the same type, which is denoted by the `weight` attribute. Although the `Confirmation`, `Cancelation`, `Acceptance` and `Delivering` actions have the input and output objects of exactly one type, they don't belong to the SISO category since each of these actions has input and output objects of the same type (they only change the objects' states). Such actions have already been covered by the previous rules (**Rule 4** and **Rule 6**).

The `Specification` action represents the first example of MISO actions in the sample DAD. This action, which has input objects of two different types (`OrderHeader` and `CatalogItem`), results in the creation of one or many objects of the third type (`OrderDetail`). The second MISO action is the `Pack&Control` action, also having input objects of two different types (`OrderHeader` and `StockItem`) and resulting in the creation of one object of the third type (`Delivery`). Although the `Collecting` action has input objects of two different types (`OrderDetail` and `StockItem`), it doesn't belong to the MISO category. It represents the activation of the `StockItem` objects (already covered by **Rule 5**) and actually belongs to the SISO category (one or many stock items are prepared for the each confirmed order detail).

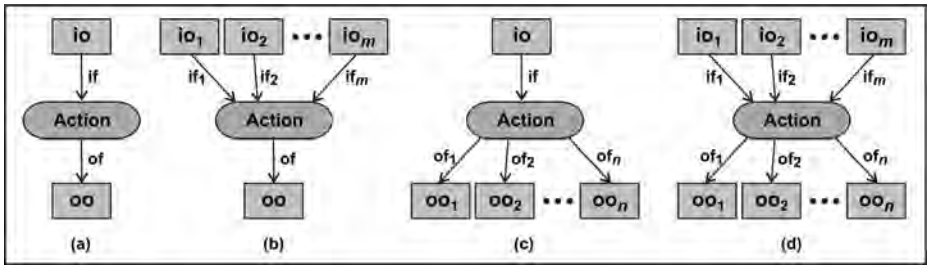


Fig. 7. Classification of actions: (a) SISO, (b) MISO, (c) SIMO, and (d) MIMO

There are no SIMO and MIMO examples in the sample DAD, but they are possible. For example, in the production unit of the given business system, separation of some mixture into its components constitutes a SIMO action. Such action has object(s) of exactly one type (mixture) and as many different types of output objects as there are different types of components in the mixture. If we add machine(s) required for the separation (machine, as an existing object, will constitute an additional input object), or some previously prepared containers for packing separated components (prepared containers, be they generated and/or activated existing objects, will also constitute additional input objects), then such SIMO action becomes a MIMO action.

Generally, input objects may be: (i) *generated* ($io_{\mathcal{G}} \in \mathcal{O}_{\mathcal{G}}$), (ii) *activated existing* ($io_{\mathcal{A}} \in \mathcal{O}^{a}_{\mathcal{X}}$), i.e. existing objects that have been already activated by some previous action in the given business process, and (iii) *non-activated existing* ($io_{\mathcal{X}} \in \mathcal{O}^{n}_{\mathcal{X}}$), i.e. existing objects that have not been activated in the given business process. On the other hand, output objects may be: (i) *generated* ($oo_{\mathcal{G}}$), and (ii) *activated existing* ($oo_{\mathcal{A}}$). Classes $e_{\mathcal{G}} \in \mathcal{E}_{\mathcal{G}}$ and $e_{\mathcal{X}} \in \mathcal{E}_{\mathcal{X}}$ in the target CDM, corresponding to *generated* and *non-activated existing* objects, respectively, are created by **Rule 2** (extraction of objects), while classes $e_{\mathcal{A}} \in \mathcal{E}_{\mathcal{A}}$, corresponding to *activated existing* objects, are created by **Rule 5** (activation of existing objects). These classes will constitute the source and target classes in the mapping of SISO, MISO, SIMO and MIMO actions into the corresponding *object-object* associations.

SISO actions. Firstly we will consider SISO actions. SISO cases and the corresponding transformation rules are illustrated in Fig. 8.

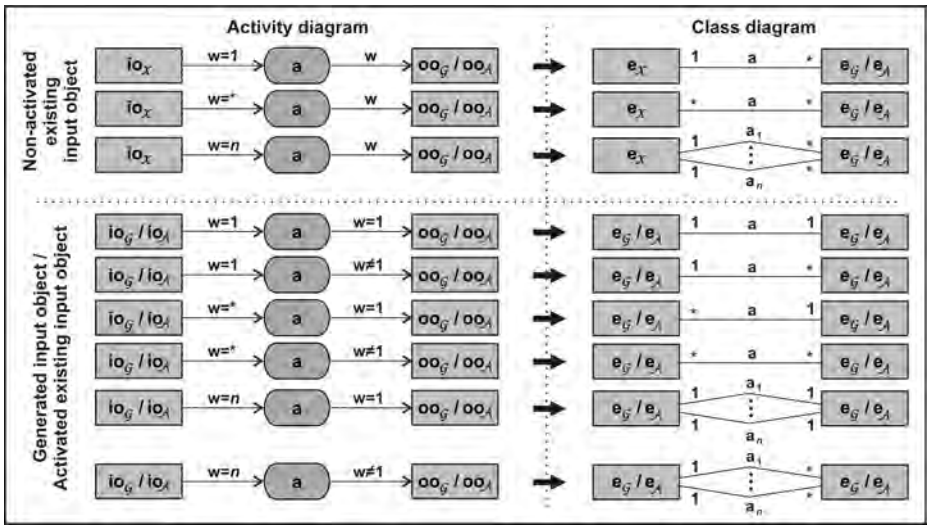


Fig. 8. Illustration of transformation rules for SISO actions

Each output object (be it generated or activated existing) depends on as many input objects as is the *weight* of the given input object flow. Thus, if the weight of the corresponding input object flow equals "1", then the given output object depends on exactly one input object and the multiplicity of the respective source association end (corresponding to the input object) is exactly "1". If the input weight equals "*", then the given output object depends on many input objects and the multiplicity of the respective source association end is "*". If the input weight is a literal *n* (*n* >1), then the given output object depends on exactly *n* input objects (like personal data containing data about two cities, where the first city represents the place of birth, while the second city represents the place of residence). In that case we have exactly *n* associations, where each association has the source end multiplicity equal to "1".

If the input object(s) is/are non-activated existing object(s), i.e. $io_{\mathcal{X}}$, then the target end multiplicity (which corresponds to the output object) of each association always equals "*" and doesn't depend on the weight of the output object flow, because even in cases when the output weight is exactly "1", with time, the same existing input object may be used in the creation/activation of many output objects.

If the input object(s) is/are generated or activated existing object(s), i.e. $io_{\mathcal{G}}$ or $io_{\mathcal{A}}$, then the target end multiplicity depends only on the weight of the output object flow. If the output weight is exactly "1", then exactly one output object depends on $io_{\mathcal{G}}$ or $io_{\mathcal{A}}$ and the target end multiplicity should be exactly "1". Otherwise, the target end multiplicity should be "*" because more than one output object depend on given input object(s).

Rule 7. ***(Object-object associations for SISO actions)*** *Let $a \in \mathcal{A}$ be a SISO action with input object(s) $io \in \mathcal{O}_{\mathcal{G}} \cup \mathcal{O}^a_{\mathcal{X}} \cup \mathcal{O}^n_{\mathcal{X}}$ and output object(s) $oo \in \mathcal{O}_{\mathcal{O}}$, where $if \in \mathcal{F}_{\mathcal{I}}$ and $of \in \mathcal{F}_{\mathcal{O}}$ represent corresponding input and output object flows whose weights are denoted by w_{if} and w_{of}, respectively. Rule $\mathcal{T}^{siso}_{\mathcal{OO}}$ maps SISO tuple $\langle io, if, a, of, oo\rangle$ into set $\mathcal{R}^{siso}_{\mathcal{OO}}(a)$ containing exactly $n \in \mathbb{N}$ associations between classes $e_{\mathcal{IO}}$ and $e_{\mathcal{OO}}$:*

$$\mathcal{T}^{siso}_{\mathcal{OO}} : \mathcal{DAD}(\mathcal{P}, \mathcal{A}, \mathcal{O}, \mathcal{F}) \mapsto \mathcal{CM}(\mathcal{E}, \mathcal{R}) \overset{def}{\Longleftrightarrow} \mathcal{R}^{siso}_{\mathcal{OO}}(a) = \mathcal{T}^{siso}_{\mathcal{OO}}\big(\langle io, if, a, of, oo\rangle\big)$$

$$\mathcal{R}^{siso}_{\mathcal{OO}}(a) = \Big\{ r^{(j)}_{\mathcal{OO}} \in \mathcal{R} \mid r^{(j)}_{\mathcal{OO}} = \mathcal{T}^{*}_{\mathcal{OO}}\big(\langle io, if, a, of, oo\rangle\big), j = 1, ..., n \Big\}$$

$$\mathcal{T}^{*}_{\mathcal{OO}}\big(\langle io, if, a, of, oo\rangle\big) \overset{def}{=} r_{\mathcal{OO}} \mid \Big(name(r_{\mathcal{OO}}) = name(a) \wedge \big(memberEnd(r_{\mathcal{OO}}) = \{source, target\} \mid type(source) = e_{\mathcal{IO}} \wedge multiplicity(source) = m_s \wedge type(target) = e_{\mathcal{OO}} \wedge multiplicity(target) = m_t \big) \Big),$$

where the corresponding source and target classes $e_{\mathcal{IO}}$ and $e_{\mathcal{OO}}$ are given with:

$$e_{\mathcal{IO}} = \begin{cases} e_{\mathcal{G}}, io \in \mathcal{O}_{\mathcal{G}} \\ e_{\mathcal{X}}, io \in \mathcal{O}^n_{\mathcal{X}} \\ e_{\mathcal{A}}, io \in \mathcal{O}^a_{\mathcal{X}} \end{cases} \qquad e_{\mathcal{OO}} = \begin{cases} e_{\mathcal{G}}, oo \in \mathcal{O}_{\mathcal{G}} \\ e_{\mathcal{A}}, oo \in \mathcal{O}^a_{\mathcal{X}} \end{cases},$$

while the corresponding source and target association end multiplicities and the total number of associations are as follows:

$$m_s = \begin{cases} *, w_{if} = * \\ 1, otherwise \end{cases} \quad m_t = \begin{cases} *, w_{of} \neq 1 \vee io \in \mathcal{O}^n_{\mathcal{X}} \\ 1, otherwise \end{cases} \quad n = \begin{cases} 1, w_{if} \in \{1, *\} \\ w_{if}, otherwise \end{cases}.$$

Application of the previous rule to the first sample SISO action (`Header-Creation`) will result in the creation of a binary association between the classes corresponding to the input and output objects (`Request` and `OrderHeader`) with the "1:1" cardinality since both input and output objects are generated and the weight of corresponding object flows equals "1". Application to the *de facto* SISO action (`Collecting`) will result in the creation of a binary association between the class corresponding to the input object (`OrderDetail`) and class corresponding to the activation of the output object (`StockItem_Collecting`) with the "1:*" cardinality, since the input is a generated object and the output is an activated existing object, while the weights of the corresponding object flows are "1" and "*", respectively.

MISO actions. Each MISO action has input objects of more than one type and it creates output objects of exactly one type. Since all input objects are required for the start of an action, we assume that the output object(s) depend(s) on all these input objects, i.e. output object(s) is/are directly related to each of the input objects. For example, in the sample DAD (`Pack&Control`), each delivery (represented by output object type `Delivery`) is related to one order (represented by the first input object type `OrderHeader`) and contains many items (represented by the second input object type `StockItem`). Similarly, for the second sample MISO action (`Specification`), all order details (represented by output object type `OrderDetail`) belong to a particular order (represented by the first input object type `OrderHeader`) and each of them is specified based on a concrete catalog item (represented by the second input object type `CatalogItem`). This implies that SISO rule $\mathcal{T}^{siso}_{\mathcal{OO}}$ should be independently applied to each *input object(s) - output object(s)* pair of MISO action, as depicted in Fig. 9 (a).

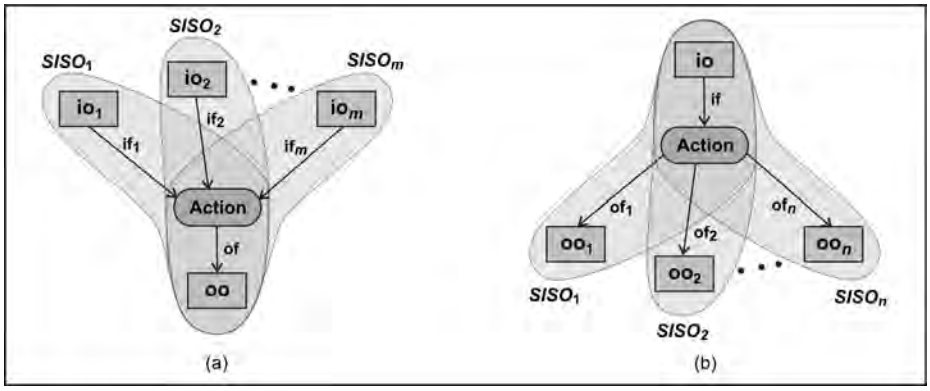


Fig. 9. Application of SISO transformation rule for MISO (a) and SIMO (b) actions

Rule 8. ***(Object-object associations for MISO actions)*** *Let $a \in \mathcal{A}$ be a MISO action with $m \in \mathbb{N}$ different types of input objects $io_1, \ldots, io_m \in \mathcal{O}_\mathcal{I}$ and output object(s) $oo \in \mathcal{O}_\mathcal{O}$, where $if_1, \ldots, if_m \in \mathcal{F}_\mathcal{I}$ and $of \in \mathcal{F}_\mathcal{O}$ constitute the corresponding input and output object flows, respectively. Let $\mathcal{M}(a) = \{\langle io_k, if_k, a, of, oo\rangle, 1 \leq k \leq m\}$ be the set of all SISO tuples for the given action $a \in \mathcal{A}$. Rule $\mathcal{T}_{\mathcal{OO}}^{miso}$ maps the $\mathcal{M}(a)$ set into the $\mathcal{R}_{\mathcal{OO}}^{miso}(a)$ set of corresponding associations for the given action:*

$$\mathcal{T}_{\mathcal{OO}}^{miso} : \mathcal{DAD}(\mathcal{P}, \mathcal{A}, \mathcal{O}, \mathcal{F}) \mapsto \mathcal{CM}(\mathcal{E}, \mathcal{R}) \overset{def}{\Longleftrightarrow} \mathcal{R}_{\mathcal{OO}}^{miso}(a) = \mathcal{T}_{\mathcal{OO}}^{miso}\big(\mathcal{M}(a)\big)$$

$$\mathcal{R}_{\mathcal{OO}}^{miso}(a) \overset{def}{=} \bigcup_{1 \leq k \leq m} \mathcal{R}_{\mathcal{OO}}^{(k)}(a), \quad \mathcal{R}_{\mathcal{OO}}^{(k)}(a) = \mathcal{T}_{\mathcal{OO}}^{siso}\big(\langle io_k, if_k, a, of, oo\rangle\big).$$

The $\mathcal{T}_{\mathcal{OO}}^{miso}$ rule is a general rule relevant to all *single output actions*, since a SISO action is just a special case of MISO actions ($m = 1$).

Application of the previous rule to the first MISO action (`Pack&Control`) in the sample DAD will result in the creation of two binary associations: (i) "1:1" association between classes corresponding to the input (`OrderHeader`) and output object (`Delivery`), and (ii) "*:1" association between the class corresponding to the activation of input objects (`StockItem_Collecting`) and the class corresponding to the output object (`Delivery`). Similarly, application to the second sample SIMO action (`Specification`) will result in the creation of the following two binary associations: (i) "1:*" association between classes corresponding to the input (`OrderHeader`) and output object (`OrderDetail`), and (ii) "1:*" association between classes corresponding to the input (`CatalogItem`) and output object (`OrderDetail`).

SIMO actions. A SIMO action has input objects of exactly one type and output objects of more than one different type. Since the given action is to result in the creation/activation of all output objects, we assume that each output object is directly related to the given input object(s). This implies that SISO rule $\mathcal{T}_{\mathcal{OO}}^{siso}$ should also be independently applied to each *input object(s) - output object(s)* pair of a SIMO action, as illustrated in Fig. 9 (b).

Rule 9. ***(Object-object associations for SIMO actions)*** *Let $a \in \mathcal{A}$ be a SIMO action with input object(s) $io \in \mathcal{O}_\mathcal{I}$ as well as $n \in \mathbb{N}$ different types of output objects $oo_1, \ldots, oo_n \in \mathcal{O}_\mathcal{O}$, where $if \in \mathcal{F}_\mathcal{I}$ and $of_1, \ldots, of_n \in \mathcal{F}_\mathcal{O}$ represent the corresponding input and output object flows, respectively. Let $\mathcal{M}(a) = \{\langle io, if, a, of_k, oo_k\rangle, 1 \leq k \leq n\}$ be the set of all SISO tuples for the given action $a \in \mathcal{A}$. Rule $\mathcal{T}_{\mathcal{OO}}^{simo}$ maps the $\mathcal{M}(a)$ set into the $\mathcal{R}_{\mathcal{OO}}^{simo}(a)$ set of corresponding associations for the given action:*

$$\mathcal{T}_{\mathcal{OO}}^{simo} : \mathcal{DAD}(\mathcal{P}, \mathcal{A}, \mathcal{O}, \mathcal{F}) \mapsto \mathcal{CM}(\mathcal{E}, \mathcal{R}) \overset{def}{\Longleftrightarrow} \mathcal{R}_{\mathcal{OO}}^{simo}(a) = \mathcal{T}_{\mathcal{OO}}^{simo}\big(\mathcal{M}(a)\big)$$

$$\mathcal{R}_{\mathcal{OO}}^{simo}(a) \overset{def}{=} \bigcup_{1 \leq k \leq n} \mathcal{R}_{\mathcal{OO}}^{(k)}(a), \quad \mathcal{R}_{\mathcal{OO}}^{(k)}(a) = \mathcal{T}_{\mathcal{OO}}^{siso}\big(\langle io, if, a, of_k, oo_k\rangle\big).$$

MIMO actions. A MIMO action has $m \in \mathbb{N}$ different types of input and $n \in \mathbb{N}$ different types of output objects.

Following the same analogy as for MISO actions, we can conclude that each output object of MIMO action depends on each input object. This implies that each MIMO action can be considered as a set of n concurrent MISO actions. For example, a supposed MIMO action (separation of components) can be treated as a set of several MISO actions such that each MISO action represent extraction of one component from the input mixture. In this way each separated component depends on the input mixture as well as the separation machine. Similarly, following the same analogy as for SIMO actions, each MIMO action can also be considered as a set of m concurrent SIMO actions.

Since each MISO action can be considered as a set of m SISO actions, each MIMO action can be considered as a set of $m * n$ SISO actions. Consequently, the $\mathcal{T}_{\mathcal{OO}}^{siso}$ rule should be applied to all these SISO tuples of a MIMO action.

Rule 10. ***(Object-object associations for MIMO actions)*** *Let $a \in \mathcal{A}$ be a MIMO action with $m \in \mathbb{N}$ different types of input objects $io_1, ..., io_m \in \mathcal{O}_{\mathcal{I}}$ and $n \in \mathbb{N}$ different types of output objects $oo_1, ..., oo_n \in \mathcal{O}_{\mathcal{O}}$, where $if_1, ..., if_m \in \mathcal{F}_{\mathcal{I}}$ and $of_1, ..., of_n \in \mathcal{F}_{\mathcal{O}}$ constitute the corresponding input and output object flows, respectively. Let $\mathcal{M}(a) = \{\langle io_j, if_j, a, of_k, oo_k\rangle, 1 \leq j \leq m, 1 \leq k \leq n\}$ be the set of all SISO tuples for the given action $a \in \mathcal{A}$. Rule $\mathcal{T}_{\mathcal{OO}}^{mimo}$ maps the $\mathcal{M}(a)$ set into the $\mathcal{R}_{\mathcal{OO}}^{mimo}(a)$ set of corresponding associations for the given action:*

$$\mathcal{T}_{\mathcal{OO}}^{mimo} : \mathcal{DAD}(\mathcal{P}, \mathcal{A}, \mathcal{O}, \mathcal{F}) \mapsto \mathcal{CM}(\mathcal{E}, \mathcal{R}) \overset{def}{\Longleftrightarrow} \mathcal{R}_{\mathcal{OO}}^{mimo}(a) = \mathcal{T}_{\mathcal{OO}}^{mimo}\big(\mathcal{M}(a)\big)$$

$$\mathcal{R}_{\mathcal{OO}}^{mimo}(a) \overset{def}{=} \bigcup_{\substack{1 \leq j \leq m \\ 1 \leq k \leq n}} \mathcal{R}_{\mathcal{OO}}^{(j,k)}(a), \quad \mathcal{R}_{\mathcal{OO}}^{(j,k)}(a) = \mathcal{T}_{\mathcal{OO}}^{siso}\big(\langle io_j, if_j, a, of_k, oo_k\rangle\big).$$

MISO and SIMO actions represent special cases of MIMO actions (for MISO actions $n = 1$, for SIMO actions $m = 1$). Consequently, the aforementioned **Rule 10** (MIMO actions) and basic **Rule 7** (SISO actions) constitute general rules for mapping actions with both input and output objects into the corresponding *object-object* associations.

4.5. Order of application of the rules

The process of the automated CDM design, i.e the proper order of application of the rules, is determined by the mutual dependence of the rules. Generally, the rules that create classes are to be applied before the rules for automated generation of associations since the associations cannot be generated if there are no previously generated classes. However, some rules are mutually independent and can be applied in any order.

More concretely, the rules that create classes for the extracted participants (Rule 1) and objects (Rule 2) are to be applied before the rules for automated generation of associations (Rules 3-10). These two rules for automated generation of classes can be applied in any order due to their mutual independence.

The rules for automated generation of *participant-generated object* (Rules 3-4) and *participant-existing object* (Rules 5-6) associations are also mutually independent and can be applied in any order, too. It is only important that the rule for mapping activation of existing objects (Rule 5) be applied before the rule for the usage of activated existing objects (Rule 6).

The rules for automated generation of *participant-object* associations (particularly Rule 5) are to be applied before the rules for automated generation of *object-object* associations (Rules 7-10) since the rule for mapping activation of existing objects results in the creation of an activation class that may be used in automated generation of *object-object* associations.

Finally, the automated generation of all *object-object* associations can be performed by applying the MIMO rule (Rule 10) that uses the basic SISO rule (Rule 7)[6].

5. Experimental Results

Automated generator. The target automated generator of the initial CDM is implemented as a Topcased[7] [38] plugin named **ADBdesign** whose prototype has already been presented in [7]. This improved release implements all previously formally specified rules.

The functionality of the Topcased platform and the implemented generator is based on the standard UML2 Eclipse plugin as the EMF[8] - based [8] implementation of the UML 2.1 specification for the Eclipse platform, which enables visual UML modeling, as well as the program generation of UML diagrams, and ensures a satisfactory visualization of automatically generated diagrams. Each UML diagram in the Topcased model is represented by two files of the same name, but different extensions. The *.uml* file contains the XMI[9] representation of the diagram, while the *.umldi* file describes its visualization.

The implementation is based on the combination of the DOM XML parser for the source diagram analysis and the UML2 factory [17] for the target diagram generation. The generator processes the source *.uml* file containing the DAD description and generates the target *.uml* file containing the CDM representation. The visualization of the automatically generated CDM, i.e. generation of the corresponding *.umldi* file, is performed by using the integrated Topcased functionality for the automatic visualization.

[6] As previously mentioned, MISO and SIMO actions are special cases of MIMO actions and, consequently, all *object-object* associations can be generated by applying the MIMO rule.

[7] Toolkit in OPen-source for Critical Application & SystEms Development

[8] Eclipse Modeling Framework

[9] XMI (XML Metadata Interchange) [25] is the OMG standard for platform independent metadata interchange enabling the serialization of MOF (Meta-Object Facility)-based models and metamodels into XML and vice versa, the visualization of MOF-based models and metamodels from XML.

Automatically generated sample CDM. The implemented generator has been applied to the sample DAD (Fig. 2). The visualization result of the automatically created CDM is depicted in Fig. 10.

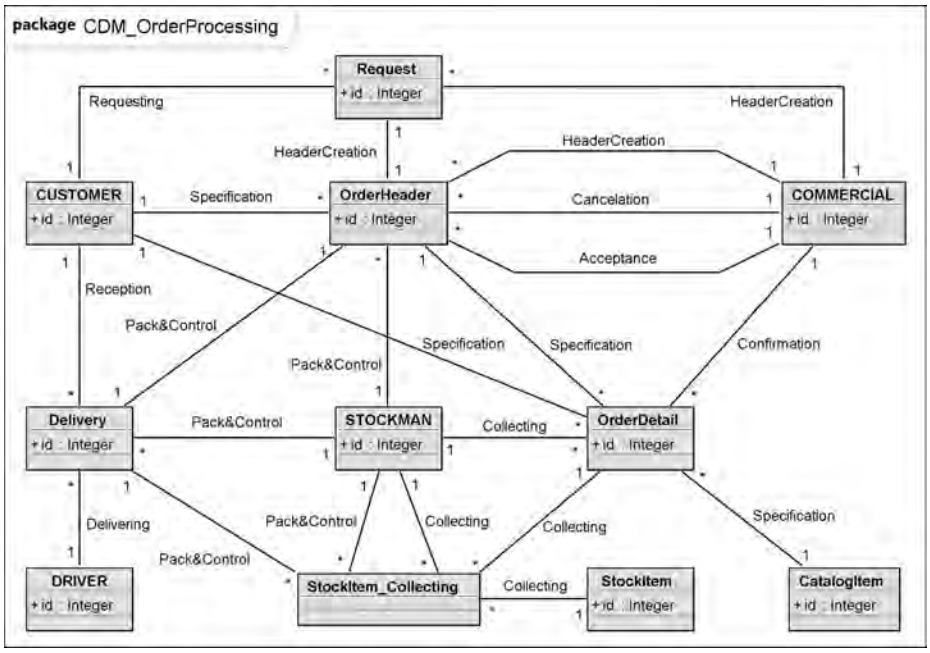


Fig. 10. Automatically generated CDM based on sample DAD

The implemented generator has created all 11 classes, as was expected during the analysis of the sample DAD, and they are as follows: (i) four classes corresponding to the participants, (ii) four classes corresponding to the generated objects, (iii) two classes corresponding to the existing objects, and (iv) one class corresponding to the activation of the existing objects. An overview of all automatically generated classes is given in Table 1.

The implemented generator has created all 22 associations, as was expected based on the analysis of the sample DAD. An overview of all automatically generated associations is given in Table 2.

Table 1. Overview of automatically generated classes based on sample DAD

Rule	Automatically generated classes
#1: $\mathcal{T}_{\mathcal{P}}$ *(participants)*	$\mathcal{E}_{\mathcal{P}}$ = { `CUSTOMER`, `COMMERCIAL`, `STOCKMAN`, `DRIVER` }
#2: $\mathcal{T}_{\mathcal{O}}$ *(generated objects)*	$\mathcal{E}_{\mathcal{G}}$ = { `Request`, `OrderHeader`, `OrderDetail`, `Delivery` }
#2: $\mathcal{T}_{\mathcal{O}}$ *(existing objects)*	$\mathcal{E}_{\mathcal{X}}$ = { `CatalogItem`, `StockItem` }
#5: $\mathcal{T}^{(1)}_{\mathcal{PEA}}$ *(activation of existing objects)*	$\mathcal{E}_{\mathcal{A}}$ = { `StockItem_Collecting` }

The first group of generated associations is *participant-object* associations and they are as follows: (i) four associations corresponding to the creation of generated objects (Rule 3: $\mathcal{T}_{\mathcal{PGO}}$), (ii) nine associations corresponding to the usage of generated objects (Rule 4: $\mathcal{T}_{\mathcal{PGI}}$), (iii) two associations related to the activation of existing objects (Rule 5: $\mathcal{T}^{(2)}_{\mathcal{PEA}}$ and $\mathcal{T}^{(3)}_{\mathcal{PEA}}$), and (iv) one association corresponding to the usage of activated existing objects (Rule 6: $\mathcal{T}_{\mathcal{PEI}}$).

The second group of generated associations is *object-object* associations and they are as follows: (i) two associations corresponding to the SISO actions (Rule 7: $\mathcal{T}^{siso}_{\mathcal{OO}}$), and (ii) four associations corresponding to the MISO actions (Rule 8: $\mathcal{T}^{miso}_{\mathcal{OO}}$).

Table 2. Overview of automatically generated associations based on sample DAD

Rule	Automatically generated associations			
	source	name	target	cardinality
#3 ($\mathcal{T}_{\mathcal{PGO}}$)	CUSTOMER	Requesting	Request	1:⋆
	COMMERCIAL	HeaderCreation	OrderHeader	1:⋆
	CUSTOMER	Specification	OrderDetail	1:⋆
	STOCKMAN	Pack&Control	Delivery	1:⋆
#4 ($\mathcal{T}_{\mathcal{PGI}}$)	COMMERCIAL	HeaderCreation	Request	1:⋆
	CUSTOMER	Specification	OrderHeader	1:⋆
	COMMERCIAL	Confirmation	OrderDetail	1:⋆
	COMMERCIAL	Cancelation	OrderHeader	1:⋆
	COMMERCIAL	Acceptance	OrderHeader	1:⋆
	STOCKMAN	Collecting	OrderDetail	1:⋆
	STOCKMAN	Pack&Control	OrderHeader	1:⋆
	DRIVER	Delivering	Delivery	1:⋆
	CUSTOMER	Reception	Delivery	1:⋆
#5 ($\mathcal{T}^{(2,3)}_{\mathcal{PEA}}$)	STOCKMAN	Collecting	StockItem_Collecting	1:⋆
	StockItem	Collecting	StockItem_Collecting	1:⋆
#6 ($\mathcal{T}_{\mathcal{PEI}}$)	STOCKMAN	Pack&Control	StockItem_Collecting	1:⋆
#7 ($\mathcal{T}^{siso}_{\mathcal{OO}}$)	Request	HeaderCreation	OrderHeader	1:1
	OrderDetail	Collecting	StockItem_Collecting	1:⋆
#8 ($\mathcal{T}^{miso}_{\mathcal{OO}}$)	OrderHeader	Specification	OrderDetail	1:⋆
	CatalogItem	Specification	OrderDetail	1:⋆
	OrderHeader	Pack&Control	Delivery	1:1
	StockItem_Collecting	Pack&Control	Delivery	⋆:1

Qualitative evaluation. The fact that the implemented generator has generated all classes and associations, as was expected during the identification of semantic capacity of the DAD and application of formal rules to the sample DAD, proves that the generator has been implemented in accordance with formal rules, but doesn't prove that the automatically generated CDM is appro-

priate for the given business system, nor does it show the degree of compliance with some manually designed CDM for the same business system. Hence, an evaluation of the automatically generated CDM will be more desirable and more important as well.

Presently there are only few implemented automated CDM generators taking UML AD as the basis. Besides our recent *ADBdesign* prototype [7] and ATL[10]-based implementations [5, 6] with modest achievements in automated generation of associations, as far as we know, there is only one (QVT[11]-based) implementation [31]. Due to its ability only for automated generation of classes corresponding to business process participants and business objects, we are unable to compare our generator with others in automated CDM design based on the same referent business models.

On the other hand, it is possible to compare an automatically generated CDM with a manually designed CDM for the same business system. However, following the well-known Conway's law [13], independent work of several database designers will result in the creation of several different CDMs for the same system, and according to Date [14], the problem of finding the logical design that is incontestably the right one is still a rather intractable problem. Consequently, the objectivity of comparison of the given automatically generated CDM with some manually designed CDM for the same system is questionable since the degree of compliance will differ from one case to another. Thus, in order to obtain some reliable evaluation of the automatically generated CDM, we will evaluate its usability from the database designer's point of view, i.e. whether the automatically generated concepts are suitable and correctly generated to be retained in the design of the target CDM for the given business system. A similar case study-based qualitative evaluation was also used for the evaluation of NLP-based approach by Chen [10] and later by some other authors.

Evaluation of automatically generated classes. All classes corresponding to the participants and business objects are suitable and could be retained without any change (existence of both classes `CatalogItem` and `StockItem` is also acceptable, for example in case of different series of the same product type, etc.). Even activation class `StockItem_Collecting`, despite its "synthetic" name, is also acceptable to be retained but with a changed name since it actually represents the delivery details, i.e. delivered items. Hence, all automatically generated classes could be retained in the target CDM.

The previous considerations and preliminary experimental results of the generator's application to several different DADs in different business domains as well, imply that: (i) implemented generator doesn't "overgenerate" classes, i.e. there are no classes that could be considered as surplus, and (ii) synthetic naming of activation classes, which usually differs from the naming in manual CDM design, doesn't have a substantial importance for some generated classes to be considered as unacceptable to be retained in the target CDM.

[10] ATLAS Transformation Language [18]
[11] Query/View/Transformation [26]

Preliminary experimental results also imply that the identified semantic capacity of the DAD and formally specified transformation rules cover the automated generation of the majority of classes of the target CDM. Other classes, such as subclasses, unions, components and other advanced (EER) concepts, are the subject of future work.

Evaluation of automatically generated associations. All four associations corresponding to the creations of generated objects (Rule 3), as well as both associations of the activation class with the classes corresponding to the given participant and existing object (Rule 5), are completely appropriate and could be retained in the target CDM without any change.

Some of the *participant-object* associations that correspond to the usage of generated objects (Rule 4) may be considered as redundant associations in the automatically generated CDM. For example, the `HeaderCreation` association between classes `COMMERCIAL` and `Request` is redundant, since these two classes are also associated via the `OrderHeader` class and corresponding associations that are more significant. Similarly, the `Specification` association between classes `CUSTOMER` and `OrderHeader` is redundant since these classes are also associated via the `OrderDetail` class, as well as the `Collecting` association between classes `STOCKMAN` and `OrderDetail` since they are also associated via the `StockItem_Collecting` class. These three redundant associations, although generated with correct cardinalities, may be considered as *overgenerated*, i.e. surplus. Although they may constitute a surplus, it is better that the generator automatically generates them since it is easier for the designer to remove some surplus concept from the automatically generated model that is not incorrectly generated, than to add some new concept (database designers sometimes introduce redundant associations to achieve better performances in data retrieval, etc.).

Two of the six remaining associations corresponding to the usage of generated objects (Rule 4) are generated with partly incorrect cardinalities. Both (`Cancelation` and `Acceptance`) have the source end multiplicity equal to "1". Although that could be corrected by changing the incorrect source end multiplicities to "0..1" in both associations (e.g. some order may be canceled, but not necessarily), it is better to consider one of them as surplus and remove it (both associations are related to the fact of cancelation/acceptance and only one of them is sufficient). In this way, the remaining association will be completely appropriate and could be retained in the target CDM without any change. Hence, one of these two associations is incorrect and constitutes a real surplus in the target CDM. This flaw is related to some control patterns (e.g. decision/merge and fork/join) that are presently not covered and will be part of future work.

The other four associations corresponding to the usage of generated objects are completely appropriate and could be retained in the target CDM without any change.

Previous considerations related to the associations corresponding to the usage of generated objects are also relevant to the associations corresponding to the usage of activated existing objects (Rule 6). There is only one such associ-

ation in the automatically generated sample CDM. Since it is redundant, it may be also considered as *overgenerated*.

Both *object-object* associations corresponding to the SISO actions in the sample DAD (Rule 7), as well as all four associations corresponding to the MISO actions (Rule 8), are completely appropriate and could be retained in the target CDM without any change (even "1:*" cardinality of the `Collecting` association is acceptable, for example in case when many delivered items with their own serial numbers correspond to the same order detail). Hence, all six *object-object* associations are correct and appropriate to be retained in the target CDM.

The preliminary experimental results of the generator's application to several different DADs in different business domains as well, imply that: (i) implemented generator "overgenerates" only small number of *participant-object* associations corresponding to the usage of generated and activated existing objects (these associations truly exist, but they are redundant and could be removed manually or retained for some other reason), and (ii) implemented generator generates (but not necessarily) some partly incorrect surplus associations in case of some control patterns (e.g. decision/merge and fork/join).

The preliminary results also imply that the identified semantic capacity of the DAD and formally specified transformation rules cover the automated generation of the majority of associations of the target CDM. Some of them (e.g. generalizations) presently may be considered as the subject of further transformations of the automatically generated initial CDM and they will be part of future work.

It is possible that some *object-object* associations cannot be automatically generated based on one single DAD, but they should exist in the target CDM. However, bearing in mind that the future generator will process the business model of the whole business system (business model of an entire business system contains several DADs representing several different business processes), it is possible that some of the missing *object-object* associations will be automatically generated based on other DADs. For example, in the target CDM for the given business system, classes `CatalogItem` and `StockItem` should be associated with "1:*" cardinality. This association cannot be automatically generated based on sample DAD, but it will be generated based on the DAD representing the production in the given business system.

Quantitative evaluation. All previously implemented CDM generators (as already mentioned, there are only few implementations) are presented without corresponding quantitative evaluation results.

There are several measures adopted for the evaluation of NLP-based CDM generators. We have adopted some of them that were introduced by Harmain & Gaizauskas [16] and Omar *et al.* [29] to perform the quantitative evaluation of the implemented generator based on the sample CDM.

Recall represents the percentage of all concepts in the target CDM that is automatically generated (percentage of all classes that is automatically

generated and percentage of all associations that is automatically generated). According to [29], it may be defined as

$$Recall = \frac{N_{correct}}{N_{correct} + N_{missing}} \cdot 100\%,$$

where $N_{correct}$ represents the number of correctly generated concepts, while $N_{missing}$ represents the number of concepts in the target model that are not automatically generated.

Precision represents the percentage of correctly generated concepts in the automatically generated CDM (percentage of correctly generated classes and percentage of correctly generated associations). According to [16], it may be defined as

$$Precision = \frac{N_{correct}}{N_{correct} + N_{incorrect}} \cdot 100\%,$$

where $N_{incorrect}$ represents the number of incorrectly generated concepts, while $N_{correct}$ is the same as previous.

Basic metrics and calculated measures for the sample CDM are given in Table 3, where $N_{generated}$ represents the total number of automatically generated concepts, while others are same as previous. Number of correctly generated associations is given as 17+4, since all 21 associations are evaluated as correct, but four of them may be considered as surplus. Just one association is evaluated as incorrect and just one association is missing.

The preliminary quantitative evaluation results, after the application of the implemented generator to several different DADs in different business domains, imply that the proposed approach has very high overall *recall* and *precision*, usually 90-100%. An extensive and objectified evaluation of the approach, based on statistically reliable number of models and with statistically reliable number of designers, will be part of future work.

Table 3. Quantitative evaluation based on sample CDM

Concepts	Metrics & Measures					
	$N_{generated}$	$N_{correct}$	$N_{incorrect}$	$N_{missing}$	$Recall$ [%]	$Precision$ [%]
Classes	11	11	0	0	100	100
Associations	22	17+4	1	1	95	95

6. Related Work

Although the idea of CDM design based on the business model is not very new, there are only few papers presenting the implemented automatic generator and providing experimental results, while the others just give the method overview.

The target CDM in our approach is represented by the UML CD. However, the CD is dominantly used for modeling the static structure of software systems at different levels of abstraction with different degrees of implementation

details. In the context of model driven software development, the CD is used from the analysis level, which is computationally independent, to the platform specific level. The *analysis level CD* is usually also called *domain model* and corresponds to the initial conceptual model of the relational database that is the subject of our paper. Consequently, this survey of related work covers more widely the automated generation of the CD based on business models without restriction only to the UML AD that represents the source model in our approach.

Garcia Molina *et al.* were the first to propose an approach to the transition from AD-based business models to the initial conceptual model [15]. They proposed the direct mapping of all information objects (*business objects*) from the AD into the respective classes in the target CD and creation of class associations based on the business rules informally specified in the *supplementary glossary*, which is not a suitable basis for automatic generation.

Besides the mapping of business objects, Rodriguez *et al.* proposed the mapping of all business process participants into the corresponding classes in CDM, and provided the corresponding QVT-based implementation [31]. That was the first automatic CD generator based on AD, although it had the ability only to generate classes. They also proposed some refinement rules for further transformations and creation of composite aggregations of automatically generated classes corresponding to partitions and superpartitions.

Following the previous two approaches, Suarez *et al.* proposed an improvement in creating class associations [37]. They proposed creating associations for actions that have input and output objects by the direct mapping of these actions into respective associations between the classes corresponding to input and output objects. This proposal can be used for the automated generation of associations, but has limitations related to the automated generation of association multiplicity since they didn't propose any explicit rule.

Yet another paper [2] takes the UML AD as the basis for the creation of the CD. Proposing the mapping of the whole AD into the one single class, this paper doesn't belong to the group of all previous papers since such mapping isn't suitable if the AD is used for business process modeling.

The majority of related papers take the AD as the basis for CD design. However, there are also some other papers taking BPMN-based business process models as the basis, but presently they only provide guidelines [32] and ontology-based tool assistance [33] for the extraction of classes only.

The paper by Kamimura *et al.* [19] presents an approach to CDM design based on business model, but with source and target notations that differ from the approach presented in this paper. They propose a detailed algorithm for generating the E-R diagram using the *well-disciplined* IDEF0-based business model, but without an actual implementation.

In our previous papers, we firstly presented an example of the manual use-case-driven approach to CDM design based on the creation of classes corresponding to the extracted participants and business objects from the DAD and the creation of some *participant-object* associations [4]. This approach was the

basis for the *ADBdesign* prototype [7] and its equivalent ATL-based implementation [6]. However, these attempts had flaws regarding the cardinality of the generated associations since all business objects were treated without making a distinction between existing and generated objects. Additionally, in these papers the semantic capacity of UML AD was not sufficiently identified to enable definition of the transformation rules for automated generation of *object-object* associations. Recently, we have been considering the semantic capacity of object flows and actions for automated generation of associations and have performed an analysis related to: (i) the nature of action nodes based on the number of different types of input and output objects, and (ii) the weight of object flows. By introducing the classification of actions into SISO, MISO, SIMO and MIMO actions and making a distinction between existing and generated input objects, we have defined the formal rules for generation of *object-object* associations [5]. However, in that paper we did not consider the activation of existing objects and the subsequent usage of activated existing objects.

This paper includes the results from our previous papers in the field and further develops and more completely and thoroughly presents the entire approach to automated design of the initial CDM based on the business process model represented by the DAD, by covering: (i) the extraction of business process participants and business objects, followed by the automated generation of corresponding classes, (ii) the extraction of action nodes and object flows, followed by the automated generation of *participant-object* and *object-object* associations with respect to the distinction between *generated business objects* (business objects that are generated in the given business process) and *existing business objects* (business objects that are not generated in the given business process and exist independently of the given business process), as well as the *activation of existing objects* and the subsequent usage of *activated existing objects* in the given business process.

7. Conclusion and Future Work

Inspired by some previous papers presenting approaches to CDM design based on business process models, that have mainly resulted in the automated generation of classes and limited set of their associations in the CD representing the CDM, the main objective of our research is the identification of the semantic capacity of the AD and the definition of rules for automated CDM design.

We have identified the part of the UML metamodel that is commonly used for AD-based business process modeling (we use the DAD term), as well as the part commonly used for the CD-based CDM. Based on formal definitions of the source DAD and target CD and some previous proposals, we have firstly defined two formal rules for the automated extraction of participants and business objects and creation of corresponding classes.

By following the recently made distinction between existing and generated business objects, we have considered their typical occurrences in business processes and defined four rules for automated generation of *participant-object*

associations. Two of them cover typical manipulations with generated objects (creation and subsequent usage of created objects), while the other two formal rules cover typical manipulations with existing objects (activation and subsequent usage of activated objects).

Introduction of the *activation* concept enables a similar treatment of activated existing objects and generated objects and enables the application of rules for automated generation of *object-object* associations as well. We have defined four formal rules for automated generation of *object-object* associations covering four possible cases (SISO, MISO, SIMO, MIMO) of actions having input and output objects. Since MISO and SIMO actions represent special cases of MIMO actions, each action with input and output objects is considered as a MIMO action and treated as a set of concurrent SISO actions. Such assumption enables the SISO rule to constitute the basic rule that can be applied to each action with input and output objects.

Based on these formal rules we have implemented an automated CDM generator as a Topcased plug-in named ADBdesign. The implemented generator has been applied to the sample business process model. The results of the qualitative and quantitative analysis of the generator's application to the sample model, as well as the preliminary results of application to some other business process models in different business domains, show that the generator is able to generate a very high percentage of the target CDM (recall usually exceeds 90%) and has a very high precision (over 90% of all automatically generated concepts are usually correct). The performed analysis proves that the UML AD has the semantic capacity for automated generation of the proper structure of the target CDM.

All formal rules for automated CDM design have been defined for UML AD-based business process models. Since all these rules have been derived based on typical business process patterns and typical usages of business objects in business processes, they can also be easily adopted for application to some other business process modeling notations.

The future work will be focused on the extension of the covered UML metamodel for business model representation and further identification of its semantic capacity for automated CDM design. The influence of control patterns, such as alternative (decision/merge) and concurrent control flows (fork/join), which are presently not considered, will also be part of future work.

Presently, the implemented generator is able to process only one single DAD representing the business model of one business process in the business system. The future work will be also focused on extending formal rules to the whole business model (business system typically has more than one business process and the corresponding business model usually contains more than one DAD) and implementing the automated CDM generator that will be able to process whole business model and automatically generate the CDM for the whole business system.

References

1. Alencar, F., Marín, B., Giachetti, G., Pastor, O., Castro, J., Pimentel, J.: From i* requirements models to conceptual models of a model driven development process. In: Proceedings of the PoEM 2009. pp. 99–114. Springer (2009)
2. Barros, J., Gomes, L.: From activity diagrams to class diagrams. In: Workshop Dynamic Behaviour in UML Models: Semantic Questions, In conjunction with Third Int. Conf. on UML. York, UK (2000)
3. Batini, C., Demo, B., Di Leva, A.: A methodology for conceptual schema design of office databases. Information Systems 9(3-4), 251–263 (1984)
4. Brdjanin, D., Maric, S.: An example of use-case-driven conceptual design of relational database. In: Proceedings of the EUROCON 2007. pp. 538–545. IEEE (2007)
5. Brdjanin, D., Maric, S.: On automated generation of associations in conceptual database model. In: De Troyer, O., et al. (eds.) ER Workshops 2011, LNCS, vol. 6999, pp. 292–301. Springer-Verlag, Berlin Heidelberg (2011)
6. Brdjanin, D., Maric, S.: Towards the initial conceptual database model through the UML metamodel transformations. In: Proceedings of the EUROCON 2011. pp. 1–4. IEEE (2011)
7. Brdjanin, D., Maric, S., Gunjic, D.: ADBdesign: An approach to automated initial conceptual database design based on business activity diagrams. In: Catania, B., Ivanovic, M., Thalheim, B. (eds.) ADBIS 2010, LNCS, vol. 6295, pp. 117–131. Springer-Verlag, Berlin Heidelberg (2010)
8. Budinsky, F., Steinberg, D., Merks, E., Ellersick, R., Grose, T.: Eclipse Modeling Framework. Pearson Education, Boston, USA (2003)
9. Chen, P.: The entity-relationship model: Toward a unified view of data. ACM ToDS 1(1), 9–36 (1976)
10. Chen, P.: English sentence structure and entity-relationship diagrams. Information Sciences 29(2-3), 127–149 (1983)
11. Choobineh, J., Mannino, M., Nunamaker, J., Konsynsky, B.: An expert database design system based on analysis of forms. IEEE Transaction on Software Engineering 14(2), 242–253 (1988)
12. Choobineh, J., Mannino, M., Tseng, V.: A form-based approach for database analysis and design. Communications of the ACM 35(2), 108–120 (1992)
13. Conway, M.: How do committees invent? Datamation (1968)
14. Date, C.: An Introduction to Database Systems, 8th edn. Addison-Wesley, Reading, USA (2003)
15. Garcia Molina, J., Jose Ortin, M., Moros, B., Nicolas, J., Troval, A.: Towards use case and conceptual models through business modeling. In: Laender, A., Liddle, S., Storey, V. (eds.) ER 2000, LNCS, vol. 1920, pp. 281–294. Springer-Verlag, Berlin Heidelberg (2000)
16. Harmain, H., Gaizauskas, R.: CM-Builder: A Natural Language-Based CASE Tool for Object-Oriented Analysis. Automated Software Engineering 10(2), 157–181 (2003)
17. Hussey, K.: Getting Started with UML2. IBM Corp, New York, USA (2006)
18. Jouault, F., Allilaire, F., Bezivin, J., Kurtev, I.: ATL: A model transformation tool. Science of Computer Programming 72(1-2), 31–39 (2008)
19. Kamimura, M., Inoue, K., Hasegawa, A., Kawabata, R., Kumagai, S., Itoh, K.: Integrated diagrammatic representations for data design in collaborative processes. Journal of Integrated Design & Process Science 7(4), 35–49 (2003)

20. Ko, R., Lee, S., Lee, E.: Business process management (BPM) standards: A survey. Business Process Management Journal 15(5), 744–791 (2009)
21. Lukovic, I., Mogin, P., Pavicevic, J., Ristic, S.: An approach to developing complex database schemas using form types. Software: Practice & Experience 37(15), 1621–1656 (2007)
22. Martin, J.: Information Engineering. Prentice Hall, Englewood Cliffs, USA (1990)
23. Naiburg, E., Maksimchuk, R.: UML for Database Design. Addison-Wesley, Reading, USA (2001)
24. National Institute of Standards and Technology (NIST): FIPSP 183 - Integration Definition for Function Modeling (IDEF0). NIST, Gaithersburg (1993)
25. Object Management Group (OMG): MOF 2 XMI Mapping Specification, v2.4.1. OMG (2011)
26. Object Management Group (OMG): MOF 2.0 Query/View/Transformation Specification, v1.1. OMG (2011)
27. Object Management Group (OMG): Unified Modeling Language: Infrastructure, v2.4.1. OMG (2011)
28. Object Management Group (OMG): Unified Modeling Language: Superstructure, v2.4.1. OMG (2011)
29. Omar, N., Hanna, P., McKevitt, P.: Heuristics-based entity-relationship modelling through natural language processing. In: Proceedings of the Fifteenth Irish Conference on Artificial Intelligence and Cognitive Science (AICS-04). pp. 302–313 (2004)
30. Reising, W., Muchnick, S., Schnupp, P.: A Primer in Petri Net Design. Springer-Verlag, New York, USA (1992)
31. Rodriguez, A., Fernandez-Medina, E., Piattini, M.: Towards obtaining analysis-level class and use case diagrams from business process models. In: Song, I.Y., et al. (eds.) ER Workshops 2008, LNCS, vol. 5232, pp. 103–112. Springer-Verlag, Berlin Heidelberg (2008)
32. Rungworawut, W., Senivongse, T.: From business world to software world: Deriving class diagrams from business process models. In: Proceedings of the 5th WSEAS Int. Conf. on Aplied Informatics and Communications. pp. 233–238. WSEAS (2005)
33. Rungworawut, W., Senivongse, T.: Using ontology search in the design of class diagram from business process model. PWASET 12, 165–170 (2006)
34. Russell, N., van der Aalst, W., ter Hofstede, A., Wohed, P.: On the Suitability of UML 2.0 Activity Diagrams for Business Process Modeling. In: Proceedings of the 3rd Asia-Pacific conference on Conceptual modeling. pp. 95–104. Australian Computer Society, Darlinghurst (2006)
35. Scheer, A.: Business Process Engineering: Reference Models for Industrial Enterprises, 2nd edn. Springer-Verlag, New York, USA (1994)
36. Shu, N., Wong, H., Lum, V.: Forms approach to requirements specification for database design. ACM SIGMOD Record 13(4), 161–172 (1983)
37. Suarez, E., Delgado, M., Vidal, E.: Transformation of a process business model to domain model. In: Proceedings of the WCE 2008 - World Congress on Engineering. pp. 165–169. IAENG (2008)
38. TOPCASED Project: Toolkit in OPen-source for Critical Application & SystEms Development, v5.1.0. http://www.topcased.org
39. White, S., Miers, D.: BPMN Modeling and Reference Guide. Future Strategies, Lighthouse Point, USA (2008)

Drazen Brdjanin is currently a senior teaching assistant and PhD student at University of Banja Luka, Bosnia and Herzegovina. He received his Diploma degree (2000) in Computing and Automatic Control and MSc degree (2006) in Computing and Informatics, all from the Faculty of Electrical Engineering, University of Banja Luka. He was participating in several R&D projects at national and international level, and authoring several papers in the field of business modeling and automated conceptual design based on business model.

Slavko Maric is currently an associate professor at the Faculty of Electrical Engineering, University of Banja Luka, Bosnia and Herzegovina. He received his Diploma degree (1974) in Electrical Engineering from the Faculty of Electrical Engineering, University of Banja Luka; MSc degree (1979) in Computing and Informatics, from the Faculty of Electrical Engineering at University of Zagreb, Croatia; and PhD degree (2000) in Computer Science from the Faculty of Electrical Engineering, University of Banja Luka. His current fields of interest are: databases, information systems design, eGovernment systems, service oriented architecture, parallel processing and information technology standardization.

Received: March 18, 2011; Accepted: December 19, 2011.

DOI 10.2298/CSIS110106043C

Usage of Technology Enhanced Learning Tools and Organizational Change Perception

Mladen Čudanov[1], Gheorghe Savoiu[2], and Ondrej Jaško[1]

1 University of Belgrade, Faculty of Organizational Sciences, Jove Ilića 154, 11000 Belgrade, Serbia
{mladenc, jasko} @fon.rs
2 University of Pitesti, Faculty of Economics Sciences, Str. Targu din Vale nr. 1, 110040 Pitesti, Arges, Romania
gsavoiu@yahoo.com

Abstract. This paper presents an analysis of organizational changes perceived by the employees in the organizations where Technology Enhanced Learning was facilitated by tools such as wiki, (we)blog, Internet forum and social network, practice often considered as E-learning 2.0. Our research focuses on the technologically advanced organizations, leaders in the ICT and IS adoption. We specifically observe the perception of influence on the organizational structure, organizational culture and the knowledge management processes in the organization. Our findings are that the TEL tools are perceived to have a noteworthy impact on the organizational change in the three mentioned areas, and that the perception of change significantly differs depending on whether the employees are regular or are not regular users for organizational structure and knowledge management processes.

Keywords: Organizational change, TEL, E-learning 2.0, Web 2.0, wiki, blog, Internet forum, social network, organizational structure, organizational culture, knowledge management.

1. Introduction

Contemporary organizations are influenced by an exponentially growing volume of technological changes. Outstanding among those technological changes is the change in information and communication technologies (ICT), sometimes suitably called a revolution. In an overall scope of ICT applied in the organizations we can find the technologies that support learning processes, which is logical, since knowledge is based on information. The technological support to any learning activity is in its widest terms dubbed Technology Enhanced Learning (hereinafter: TEL). The research in this field covers mostly technological and academic application aspects, but the literature analyzing the impact on the organizations is scarce. The concept of TEL is specific in its wider scope, limited to neither academic nor other educational applications. Therefore the term has been chosen instead of E-

learning 2.0 although, from the technological aspect, E-learning 2.0 is more appropriate due to social network usage and Web 2.0 technologies. In this paper we will analyze the impact that technologies commonly used for TEL have on organizational structure, culture and knowledge management processes, describing TEL in business environment. In a host of other technologies, TEL is facilitated today by Wikis, forums, blogs [1], social networks and other interactive technologies that allow the teacher, the learner and the administration functions [2], but is not limited to the mentioned technologies used as example in the case of this research.

Different aspects of organization have been connected to technology since the beginnings of the scientific approach to management, to mention e.g., Frederick Winslow Taylor, but the groundbreaking work was that of Joan Woodward [3], in which relations were established between the success of the organization and the conformity of the organizational structure, on one side, and technology, on the other. Some criticism also emerged as Aston group estimated that the impact of technology on the organization is more limited than it was stated in Woodward's earlier research [4]. The research that followed only reconfirmed the links of organizational structures with technology, and introduced new factors, in addition to purely technologic artifacts, namely, the recurrent social practices in technology implementation and the community of users [5]. The contemporary research finds the influence of organizational structures and technology to be complementary [6] and reconfirms the existence of the relationship between technology and the governing structure [7,8].

In accordance with the previous analysis, we found theoretical background to study the impact that specific technologies and tools, common in TEL, have on organizations. Therefore, the following hypotheses are proposed:

H1: The perception of TEL tools usage impact on the organizational structure significantly differs between the employee groups pursuing regular and non-regular TEL usage

H2: The perception of TEL tools usage impact on the organizational culture is significantly different between employee groups with regular and non-regular TEL usage

H3: The perception of TEL tools usage impact on knowledge management practices in organization is significantly different between employee groups with regular and non-regular TEL usage

2. Methods

In order to empirically check whether there is any perceived influence that the implementation of TEL tools has on organizational changes, we have conducted a survey among the employees in knowledge based industries

(e.g. software engineering, business analysis, consulting, high education, design) that are supposed to use the TEL tools widely. Since the term TEL is not well known to the population, tools like wiki, forums, blog and social networking were explicitly presented to respondents as Web 2.0 concepts that can be used as TEL tools. The organizations to serve as samples were carefully chosen in order to represent the population of technology advanced organizations. Overall, we have gathered 55 responses online and 45 responses in paper as a response to more than 250 invitations. There were 6 questionnaires with missing values. This survey was part of a larger research effort that also included a dissemination of Web 2.0 and TEL tools and loyalty shifts from the company as an entity to industry as a group of connected individuals somewhat resembling a medieval guild (without restrictions), which in turn might be the topic of future papers.

The survey consisted of a descriptive part, where respondents stated their industry, working experience, company size, average number of forum, wiki, blog and social networks used for the sake of job tasks and open comments. A large number of interviewed personnel in our sample was working in software engineering/development (26), education (15), marketing related professions (14) or high education (12). The total working experience ranged between six months and 30 years, the mean being 6.18 years and the standard deviation amounting to 4.70. The respondents evaluated their perception of influence on the structure, the culture and the knowledge management practices on a scale from 1-10, where 1 was described as an extremely low and 10 as an extremely high influence.

Important terms were described for the convenience of respondents, and definitions were proposed on questionnaire in order to ensure the understanding among parties. The organizational structure was defined as a formal system of hierarchy, coordination, communication and control in the company. The organizational culture was defined as a system of beliefs, values, customs, behaviours and traditions shared by employees that defines the ways in which they interact with each other and with other stakeholders. The knowledge management was defined as a range of strategies and practices used in an organization to identify, create, represent, distribute, and enable the adoption of knowledge.

Finally, a definition need to be provided for both the regular and a non-regular implementation of TEL tools mentioned in the introduction and in the hypotheses. Since literature does not provide an explicit range of chosen sets of tools, we have suggested a border in the light of the frequency analysis described in the results section and the common sense at 5 uses of TEL tools per week. That point was chosen as a border in the light of the analysis of six user groups and on the assumption that a significant usage should in average considered to be at least once per day, i.e. at least five times per week. Also, it is in accordance with our sample median of 6, and therefore such division gives us roughly the same group size, needed for a further statistical analysis.

The original survey with explanations in electronic form is available at http://www.surveymonkey.com/s/JBC969J. First, the chosen variables were

analyzed using the descriptive statistics, and the frequency analysis. The gathered data did not yield any conclusive directions towards the model as in the case of using a linear or other regressions with several experimented models, and the results suggest that the included factors do a good job in explaining part of the dependent variable variance, but that more factors should be incorporated in the model. This will be one of the main guidelines in our future research. To further examine our hypothesis a t-test has been used to determine a statistically significant difference since this is the method recommended in the cases when samples are relatively small [9] as stated by Krishnawamy et al, p352.

3. Results

Regarding the use of the selected TEL tools, the respondents offered 94 valid responses, the resultant mean of which was 22.77 and the median was 6 uses per week, with the standard deviation of 36.27. The descriptive statistics is presented in Table 1, and bar chart of average uses per week on the x-axis with the percentage of users on the y-axis is presented in the following figure.

Table 1. How often do respondents use Web 2.0 features weekly

N	Minimum	Maximum	Mean	Std. Deviation
94	.00	150.00	22.7713	36.26935

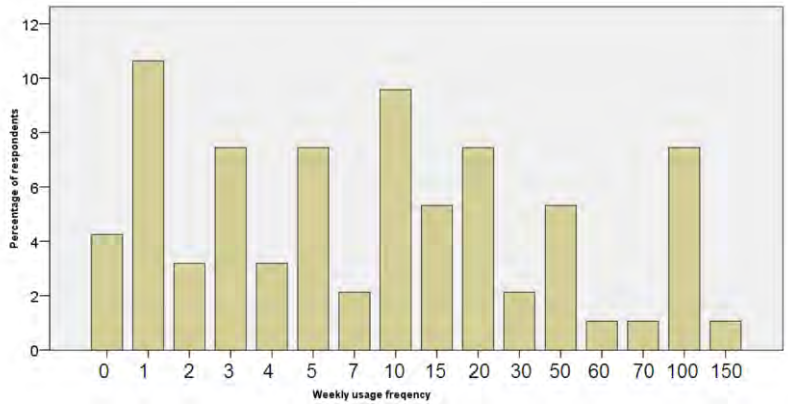


Fig. 1. Bar chart for the frequency of use per week by the percentage of users

When we analyze influence on the structure, the culture and the knowledge management in the organization, results differ. On a 1 to 10 scale, where 1 means the least influence, and 10 means the most influence, the average for the entire sample was 3.87, with a standard deviation of 2.42. This is quite a low value, and can be explained as the perception of the interviewed employees in technologically advanced industries that mentioned concepts

have a below-moderate influence upon the organizational structure. Electronic ties of all sorts, including collaborative networks are found to be loosening the constraints of the organizational structure [10]. The influence of computer networks on the organizational structure is already elaborated, and it is concluded that computer networks (including collaborative networks) enable the emergence of new organizational forms [11]. The perceptions of the employees about the influence of the mentioned concepts on culture are higher; the respondents offered an average grade of 4.67, with a standard deviation of 2.56. This could be interpreted as the perception that the mentioned Web 2.0 concepts have a moderate influence on the organizational culture. Major influence, as was expected, was perceived to be that on the knowledge management, where the average grade is 5.52, with a standard deviation of 2.75. The processed data are presented in Table 2.

Table 2. Perception of partial influences descriptive statistics

		Perceived influence on structure	Perceived influence on culture	Perceived influence on knowledge management
N	Valid	100	99	100
	Missing	0	1	0
Mean		3.87	4.67	5.52
Median		3.00	4.00	5.50
Std. Deviation		2.423	2.556	2.747

If we want to differentiate the results on the use of the mentioned TEL tools in detail, we can select six logically grouped clusters that are similar in size, and given in the following table with selected comments specific to the respective groups. If we analyze the perception of the influence on those groups, we will find that there is some correlation between the usage and the perception of influence on structure, culture and knowledge management. The overall perception of influence on structure also grows as usage grows, with the exception of the sixth group. The culture shows mixed results, while the influence of knowledge management suggests a strong growing trend, with the exception of the third group. The disruption of steady growth that is especially evident in the case of the perceived influence upon the culture could have a twofold explanation. It might indicate that employees perceive changes in the organizational culture differently from the changes the structure and knowledge in the context of an increased Web 2.0 usage, or it might be the consequence of other, non-measured features in this specific sample. In order to recheck the actual reason a new study can be conducted, focused on organizational culture, and possibly on a larger sample. The overall results presented in the "Perceived total influence" column of table 4 are the sum of the perceived influences for the culture, the structure and knowledge management (in each of the six groups). Such transformation of several Likert-based scales by the summation displays an acceptable internal

consistency according to George and Mallery [12] with Cronbach's Alpha value of .785. The goal was to present one aggregate measurement indicating the overall changes in organization. The results are presented in the table 4.

Table 3. Qualitative description of user groups

Weekly use (user groups)	Group	Example comments on usage
(0-2] (1)	Non-users	"I prefer other sources" "I use only basic corporate intranet"
(2-4] (2)	Occasi-onal users	"I use it when I have a problem" " I consult other people about their experience in working with some equipment and I use the information I get in sales"
(4-10] (3)	Moderate users	"More often when I have a problem" "At least once a day"
(10-16] (4)	Regular users	"Every time I have a problem, and for contacts" "I have to search for a solution on corporate blogs because of the lack of knowledge among colleagues in my organization" "A few times a day for information purposes"
(16-60] (5)	Frequent users	"…we have in-house wiki knowledge base, when I have problem, I visit forums..." "I follow many blogs, I don't visit them only when I have a problem"
(60-) (6)	Intense users	"I use the wiki, forums and social networks to look for some answers, and to keep informed about business, etc." "More than 15 times a day I look f wiki, forums, keep informed about business over social networks"

Table 4. Analysis of perceived influence in six groups of users

Weekly usage (user groups)	No. of surveyed users	Perceived influence on structure	Perceived influence on culture	Perceived influence on knowledge management	Perceived total influence
(0-2] (1)	16	3.19	3.56	4.19	10.94
(2-4] (2)	21	3.33	5.30	5.57	14.20
(4-10] (3)	16	3.88	3.31	4.69	11.88
(10-16] (4)	14	4.50	4.86	6.14	15.50
(16-60] (5)	14	4.93	6.07	6.43	17.43
(60-) (6)	13	4.54	4.92	7.00	16.46

A further analysis is aimed to be conducted on two groups, since the presented six groups are not a sufficient sample for significant statistical results to be obtained. In the light of the previous analysis, the dividing point

was chosen to be at 5 uses of TEL tools per week. The first group covers the respondents who use the TEL tools less than five times per week (the group with a lower usage frequency), while the second covers the employees that use those concepts five or more times per week (the group with a higher usage frequency). The conclusions are similar to those of the previous analysis. The first group has lower mean levels as to the influence on the structure, the culture and the knowledge management. The difference between the mean values for the perceived influence in all three groups can be observed in the following two tables.

Table 5. Mean and median analysis on segment with lower usage of TEL tools

			Perceived influence on structure	Perceived influence on culture	Perceived influence on knowledge management	Frequency of weekly usage
Group1 (less than five uses per week)	N	Valid	40	39	40	40
		Missing	0	1	0	0
	Mean		3.20	4.36	4.88	2.01
	Median		3.00	4.00	5.00	2.50
	Std. Deviation		2.09	2.66	2.89	1.18
Group 2 (five or more uses per week)	N	Valid	54	54	54	54
		Missing	0	0	0	0
	Mean		4.56	4.87	6.13	38.15
	Median		4.00	4.50	7.00	17.50
	Std. Deviation		2.55	2.53	2.59	41.73

Table 5 gives us an insight into the perception that employees who use TEL tools less than five times per day have about the influence of those tools on the organizational structure, culture and knowledge management. We can see that the mean of the perceived influence on the structure is rather low, 3.20 out of 10. The perception of influence on culture the employees report is somewhat increased, and in case of the wikis, blogs, forums and social networks the mean of the influence amounts to 4.36 out of 10. Knowledge management is perceived as most connected, and the mean of the influence in this group is 4.88. The median values follow a similar trend, and the standard deviation is rather low. The employees in this group use the mentioned tools 2.01 times per week on average, and the standard deviation of that usage is again relatively low for this group. If we compare these results with the group 2 we can see that there is almost a linear increase in the perceived influence on all three variables, and still the order of the perceived influence is the same, however, this time the values are 4.56 out of 10 for the structure, 4.87 out of 10 for the culture, and 6.13 for the knowledge management. The average frequency of usage is much higher, it amounts to 38.15 times per week, while the standard deviation is relatively higher in

comparison with the first group. In the light of these findings we have performed a t-test to see if differences between those two groups, the one with the average usage frequency of the TEL tools like wikis, blogs, forums and social networks below five times per week, and the other, with more than five uses per week show any statistical significance. The following table presents the results for the structure, the culture and the knowledge management practices, respectively.

Table 6. T-test for perceived influence on organizational structure, culture and knowledge management

T-test for perceived influence on structure, culture and knowledge management with 5 as the cutting point	Levene's Test for Equality of Variances		t-test for Equality of Means						
	F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
								Lower	Upper
Perceived influence on structure	3.178	.078	2.744	92	.007	1.356	.494	.375	2.337
Perceived influence on culture	.583	.447	.942	91	.349	.511	.543	-.567	1.590
Perceived influence on knowledge management	.971	.327	2.211	92	.029	1.255	.567	.128	2.381

Table 7. T-test for perceived influence on culture with 10 as the cutting point

T-test for perceived influence on culture with 10 as the cutting point	Levene's Test for Equality of Variances		t-test for Equality of Means						
	F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
								Lower	Upper
Perceived influence on culture	.038	.847	2.154	91	.034	1.139	.529	.089	2.189

In the table 7 results are presented for the culture, separately analyzed with the cutting point of ten uses per week, in order to propose explanations for non-significant differences presented in the previous table. Descriptive statistics show that group with ten or more uses per week (41 respondent) have mean of variable perceived influence on culture of 5.29 and standard deviation of 2.571, while the group with less than ten TEL tools uses per week (52 respondents) has mean 4.15 and standard deviation of 2.5 for the same variable.

4. Discussion

The results presented in the above tables describe the employees' perceptions but we also presume that the described results are rather accurate in the description of the real influence, on the ground of the assumption that the employee perception is often the best way to measure the less tangible organizational phenomena. In performing their daily tasks the employees get a direct insight into the organization, with estimations that are often correct. Even if not so, the employee perception can materialize as "self-fulfilling" prophecy as explained by Merton [13] and if some trend is perceived as not to have any influence on their organization, it most probably is so. Also, Wanous et al. [14] maintain that a cynical and negative perception of organizational change by employees contributes to its failure, so if employees think that those tools are insignificant and have no influence, it will probably prove to be so in the organizational changes in reality.

This paper is however concerned with three hypotheses presented in first part of the paper, connected to the results given in Table 5 as descriptive guidelines, and also to the results in Table 6 for the statistical significance of the mean difference for groups with a lower and a higher usage of the TEL tools. As described in the methods section, the t-test was used for statistical analysis. The results are relevant in verifying Hypothesis 1 (related to the structure) and Hypothesis 3 (the knowledge management), but do not support Hypothesis 2. This is all on the basis of our definition of the regular and the non-regular usages. A further discussion on results is presented in the joint description of the results, the theoretical review of the organizational impact in general and then in the review of the three parts together.

4.1. Description of results

If we interpret Levene's test for equality of variances, we can confirm the assumption that two groups in Table 6 have equal variances in all three cases and therefore follow and present the related results. The T statistics in the case of the perceived influence on the structure has the value of 2.744, with 92 degrees of freedom and two tailed significance of 0.007. Therefore we can conclude ($p<0,01$) that there is a statistically significant difference in the

means of the variable perceived influence on the structure. The second hypothesis states that the perception of the TEL tools usage impact on the organizational culture significantly differs between employee groups with the regular and the non-regular TEL usages. However, the results presented in Table 6 are not supportive to that hypothesis. The T-statistics has a relatively small value of -0.942, with 91 degrees of freedom and two-tailed significance of 0.349. Also, a 95% confidence interval of the difference is quite indicative, because values include zero. Therefore, we cannot confirm the second hypothesis. That is quite in contrast with the theoretical outline and with our expectations, and the explanation for this is given further on in the analysis. The third hypothesis states that the perception of TEL tools usage impact on the knowledge management practices in the organization is significantly different between the employee groups with the regular and the non-regular TEL usages. The results with t-statistics value of 2.211, 92 degrees of freedom and two-tailed significance of 0.029 indicate that there is a statistically significant difference between the two group means. Hence we can conclude ($p<0,05$) that the perception of influence the TEL tools have on the knowledge management practices significantly differs between the group that has a regular and the group that has a non-regular usage of the TEL tools.

Our empirical findings concur with the theory in the structure and knowledge management processes, but not in the culture. Several reasons for this have been proposed: the cutting point of five uses per week is not representative, and the function of the perceived influence is not linear, or our sample has not been large enough, or the respondents who filled in the e-survey might have had some problems in understanding the organizational culture. In order to check those assumptions, we have performed an analysis with a cutting point of 10, defining the non-regular usage as that which occurs less than 10 times per week, and the regular usage as one that occurs more than 10 times per week. The results, presented in Table 7 again point towards the assumption of equal variances, and this time the t-statistics with the value of 2.154, with 91 degrees of freedom and two tailed significance of 0.034 leads to the conclusion ($p<0,05$) that there is a statistically significant difference in the means of variable perceived influence on culture.

4.2. Theoretical review of impact of Forums, Blogs, Wikis, Social networks and other Web 2.0 based TEL tools on organization

All Web 2.0 tools, including those selected for this paper as favorable for TEL, enable companies to explore new ways to cultivate and exploit knowledge sharing with customers, suppliers and partners [15]. According to McKinsey [16], companies use tools like Wikis, Blogs or Social Networks, because they are important in supporting their market position as well as in addressing customers' demands. These Web 2.0 tools can improve the organizational and the individual performance, however, they also encounter several problems, mainly of organizational nature. Knowledge intensive industries,

such as software development companies, are quickly developing, involving many people working in different phases and activities. Knowledge in such companies is diverse and steadily growing. Organizations have problems identifying the content, location, and usage of the knowledge [17]. The employees with assigned tasks in such organizations have to communicate, collaborate, and coordinate internally in working groups, but also externally with other groups, even outside organizational borders.

As regards the enhancement of learning in business, the benefits of collaborative tools have been elaborately studied [18,19,20] even during the early years of the Internet application [21]. Collaboration as part of learning which can be significantly enhanced by ICT can be described as sharing of common business goals by employees. It helps the organization get over any physical boundaries imposed on it by departments, functions, and levels of hierarchy [22]. The development of an accessible and economically efficient technology for dealing with vast amounts of explicit and partly tacit knowledge was one of the directions of the influence the Internet has imposed on organizational changes. Tools for advanced practices of knowledge creation and distribution are often available at little or even no direct cost as open source solutions and are not reserved any more for top tier companies. The perception of changes that the Internet related technologies imply on organizations, business and other aspects of our lives has been compared several times to changes caused by the printing press [23,24] or, more modestly, by the telegraph [25]. The improvements in the nature of the web continued to influence not only the knowledge in the organizations, but also the culture and the structure.

4.3. Influence on organizational structure - theoretical and empirical outline

The discussed difference in the perception of organizational change is in accord with the current view on the organizational structure. Nadler and Tuchman [26], viewing the organization as an information processing entity, find several important relations between the organizational structure and the technology used for information processing in the organization. Relations between the ICT and the formalization of the organization and its structure have also been observed [27,28,29,30,31]. New forms of learning highlight knowledge as one of the most important resources in the organization influencing authority, power, hierarchy, coordination and collaboration mechanisms, and hence organizational structure. A case study on Infostud, the company that owns several leading Serbian web portals, shows the relation between knowledge sharing and the TEL and the organizational structure. The problems caused by a fast-pace growth of the company were solved by technological enhancement of learning process in the organization. To support knowledge creation and sharing, this company has developed a system represented by the internal web site, where the employees interact, present ideas and can read the procedures and work instructions, the

knowledge database with advanced search options, and the internal messaging system. In order to develop an appropriate culture, they have changed the reward system towards supporting the desired values. Consequently, they have changed the reporting system and communication, influencing the organizational structure.

4.4. Influence on organizational culture - theoretical and empirical outline

The previously discussed results do not entirely fit into the existing theoretical framework regarding the organizational culture. The organizational culture, as a soft organizational concept should be more influenced by the change in the interaction among the employees. The mentioned tools allow for a much better interaction, and form something that Camarinha-Matos et al. [32] define as a collaborative network - a network consisting of a variety of entities (e.g. organizations and people) that are largely autonomous, geographically distributed, and heterogeneous in terms of their operating environment, culture, social capital and goals, but collaborate to better achieve common or compatible goals, thus jointly generating value. This concept has a large intersection with organizational culture. The relations between general ICT usage and the soft, cultural aspects of the organization, such as the management styles [33], the team interaction [34], the management orientation [35], learning in organization [36] or team collaboration [37] have been widely elaborated on in the literature.

A case study on one of fastest growing Serbian companies, Mozzartbet, is not in accordance with the results presented in Table 6, since it has found out that wiki, among other features, has a significant impact upon the organizational culture in that company [38] via employee behaviour. It is viewed as one of contributing factors to a relaxed and academic organizational culture in the IT department. The collaboration in problem solving is observed in different processes in the organizations, not only in case of wiki, and the conflicts on the technical issues among employees using wiki are mostly functional. In such an environment, the IT department of Mozzartbet, one of the major users of wiki, is more creative, more responsive to clients' needs, results in higher job satisfaction among the employees, a usual case in the organizations where a functional conflict is part of the culture [39]. Other features of culture in the IT department fit in a relaxed, academic culture frame, such as a low power distance, a flexible approach to working time, a relaxed general attitude, and a constant development of interpersonal relations among employees, not characteristic for other departments in the same company where technology enhanced learning and collaboration is not common practice. A reciprocal influence of the organizational culture on the TEL tools application was also observed in that company. The application of wiki as a tool for knowledge sharing was affected by cultural expectations determining which knowledge should be shared within the organization and which should be hoarded by individuals [40], and by features indirectly related

to the organizational culture, such as the reward system and the communication with senior management.

4.5. Influence on knowledge management practices - theoretical and empirical outline

The results regarding the influence on the knowledge management adhere to the current theories and practices. Wiki allows for a large number of autonomous, heterogeneous and geographically distributed individuals and organizations to collaborate and jointly generate or record knowledge. On the Internet forum, the same entities can collaborate, but with much more interaction to find a solution to a problem, and generate, combine or share knowledge. Knowledge interactions among employees can be facilitated by the Internet forum, as a communication and collaboration tool for the development of a knowledge support system for dynamic manufacturing networks [41]. Interactive blogs are rather similar to forums, but in the context of TEL they are characterized by a more individual tone and more control. A collaborative blog allows for the multiple users to share knowledge, thus providing a collaborative network. The social networks, those business oriented in particular, can bond interested parties, creating a network of resources that can be used to solve complex business problems. Social networks connect virtual organizations [42,43], and the process is alleviated technically by the Internet social networks. Therefore our results regarding the knowledge management practices comply with the existing work in the field. Further, the published case study concerning the Mozzartbet company is in accordance with conclusions regarding the knowledge management. Wiki is, among other tools, used in that company in order to enhance the learning process, as a tool which allows the users not only to access its contents but also to change the contents online [44]. The knowledge regarding the system development is gathered in more than 1100 articles using an JSPWiki open source solution.

Simultaneously with the quantitative data gathering, the Internet forum "ITsecurity.org" was used to start discussions on the mentioned topics and gather a qualitative input. This forum is one of the major tools in collaboration and knowledge sharing on high technologies in Southeast Europe, and has a community of more than 275.000 registered members. We conducted a discussion and information exchange while the research was in progress rather than after its completion, as described by Pastore [45]. The discussion was partly aimed at the topics defined for this research. Several respondents indicated that their status in the enterprise can be affected by their being active in the usage of the mentioned tools, confirmed by examples where activity was aimed at solving practical problems in the company. New cultural norms were also described, e.g. during the job interview, the managers inquired into the applicants' acquaintance with the forum, blog or other virtual community activities. Such behaviour was accepted as appropriate and beneficial to the company, providing more information about the candidate for

the reviewer and resulting in a better selection process. Also, several responses indicated that the structure, culture and other organizational features reciprocally influence the application of wikis, blogs, forums and social networks to the benefit of the company, e.g. the organizational aversion toward those tools or cultural norms that regard it as leisure activities.

5. Conclusion

It is not often noticed that the learning process is undergoing a change in magnitude unparalleled since the dawn of the human civilization. In paradigm, the process of learning among humans has undergone few changes since the Paleolithic hunters taught younger tribe members. The concept was that distinguished individuals with knowledge presented their knowledge using the theory and practical examples to a group of individuals that presumably lacked that knowledge. The development of writing was another way to partly store that knowledge and communicate it without an interaction with the knowledge-bearer, and effective printing technologies made it applicable to wider social groups. Today, TEL allows for the forming of collaborative networks and generates knowledge in interaction among participants with much more efficiency due to technology enablers, creating an effective and efficient knowledge system that is more than a mere sum of its components. It brings another paradigm change that could lead to even more important advances than the previous ones. Therefore, we can expect that such change will also have an impact upon the organizational phenomena.

We can conclude that the application of modern learning technologies, predominantly of the ICT domain is a novel and promising field. In addition to the issues concerned with core principles and technologies, a large number of lateral questions arise. The impact on the organization is one of those questions, and we hope that this paper will shed some light on them. Based on this research, we conclude that the application of the TEL tools selected as representative for business organizations is perceived as a noteworthy impact on the change of the organizational structure, the culture and the knowledge management processes in the organization. Also, we can see the differences in the perception reported by the groups as related to the usage frequency of those tools. The differences are evident in mean values, but are of a statistical significance only for the structure and the knowledge management processes. If we however choose a different cutting point in 10 uses per week, the difference in the perception of influence on the culture change also becomes statistically significant.

We suggest that further research on this matter should be aimed at description of changes that are observed as influenced by the TEL tools in this paper. Specific aspects of the structure, like centralization, job division, coordination, control, departmentalization or formalization, to name just a few, could be analyzed under the influence of those tools. Also, the change in the

knowledge management processes and its specific parts, like knowledge mapping, creation or sharing could be documented. Further research should especially be aimed at clearing out the perception of influence on the organizational culture change. There is also a large number of other characteristics in the organization that could be changed by the impact of the TEL tools application, that can form more elaborated model in future research.

Acknowledgements. The authors would like to thank to the anonymous reviewers whose constructive comments helped to substantially improve this article and to all representatives of the organizations that have participated in this research. This research has been funded by the Republic of Serbia Ministry of Science and Technological Development, Program of Basic Research 2011-2014 project 47003 - Infrastructure for electronically supported learning in Serbia.

References

1. Ebner, M., Holzinger, A., Maurer, H.: Web 2.0 Technology: Future Interfaces for Technology Enhanced Learning. In C. Stephanidis (Ed.): Universal Access in HCI, Part III, 559–568, Berlin, Germany: Springer-Verlag. (2007)
2. Shimic, G.: Technology enhanced learning tools. In M. Lytras, D. Gasevic, P. Pablos, W. Huang (Eds.), *Technology enhanced learning: Best practices*, 1-27. Hershey, NY: IGI Publishing. (2008)
3. Woodward, J.: Industrial Organization: Theory and Practice. London, UK: Oxford University Press. (1965)
4. Pugh, D.S., Hickson, D.J., Hinings, C.R., MacDonald, K.M., Lupson, T.: A conceptual scheme for organizational analysis. Administrative Science Quarterly, 8(3), 289-315. (1963)
5. Orlikowsky, W.J.: Using technology and constituting structures: A practice lens for studying technology in organizations. Organization science, 11(4), 404-428. (2000)
6. Tiwana, A., Konsynski, B.: Complementarities Between Organizational IT Architecture and Governance Structure. Information Systems Research, 21(2), 288-304. (2011)
7. Karlsson, C., Taylor, M., Taylor, A.: Integrating new technology in established organizations: A mapping of integration mechanisms, International Journal of Operations & Production Management, 30(7), 672 – 699. (2010)
8. Alavi, S.H.A, Matin, H.Z., Jandaghi, G., Saeedi S.R.R.: Pathology of Structure and Organization (Administrator Organization) of Cadastre Plan from Technological Perspective. European Journal of Economics, Finance and Administrative Sciences, 18, 85-98. (2010)
9. Krishnaswamy, K.N., Sivakumar A.I., Mathirajan M.: Management Research Methodology: Integration of Methods and Techniques. Delhi, India: Dorling Kindersley, p. 352. (2006)
10. Faraj, S., Wasko M.M.: The Web of Knowledge: An Investigation of Knowledge Exchange in Networks of Practice, working paper. (2001)
11. Alavi, M. Leidner, D.: Knowledge management systems: Issues, challenges and benefits. *Communication of the Association for Information Systems*, 1, 1-28. (1999)
12. George, D., Mallery, P.: SPSS for Windows step by step: A simple guide and reference. 11.0 update (4th ed.). Boston, USA: Allyn & Bacon. (2003)

13. Merton, R.K.: The self-fulfilling prophecy. *Antioch Review,* 8(2), 193-210. (1948)
14. Wanous, J., Reichers, A. & Austin, J.: Understanding and Managing Cynicism about Organizational Change. *The Academy of Management Executive,* 11(1), 48-59. (1997)
15. Mentzas, G., Kafentzis, K., Georgolios, P.: Knowledge services on the semantic web. *Communications of the ACM*, 50(10), 53-58. (2007)
16. McKinsey: How businesses are using Web 2.0: A McKinsey Global Survey. (2007)
17. Rus, I., Lindvall, M.: Knowledge Management in Software Engineering, *IEEE Software*, 19(3), 26-38. (2002)
18. Aissi S., Malu P., Srinivasan K.: E-Business Process Modeling: The Next Big Step. *Computer*, Vol. 35, No. 5, 55-62. (2002).
19. Tredinnick, L.: Web 2.0 and Business: A pointer to the intranets of the future? *Business Information Review*, 23, 226-234. (2006)
20. Chen, M., Zhang, D., Zhou, L.: Empowering collaborative commerce with Web services enabled business process management systems, *Decision Support Systems,* 43, 530– 546. (2007)
21. Cutkosky, M.R., Tenenbaum, J.M., & Glicksman, J.: Madefast: collaborative engineering over the Internet. *Communications of the ACM.* 39(9), 78-87. (1996)
22. Malik, K., Goyal, D.P.: Organizational environment and information systems. *Vikalpa*, 28(1), 61-75. (2003)
23. Builder C.: Is it a Transition or a Revolution?, *Futures*, 25(3), 155-168. (1993).
24. Bawden D., Robinson, L.: A distant mirror?; the Internet and the printing press. *Aslib Proceedings* 52(2), 51 – 57. (2000).
25. Standage T.: The Victorian Internet. New York, USA: Berkley Books. (1998)
26. Tushman M.L. Nadler D.A.: Information Processing as an Integrating Concept in Organizational Design. *The Academy of Management Review*, 3(3), 613-624. (1978).
27. Bailey, A., Nielsen E.H.: Creating a Bureau-Adhocracy: Integrating Standardized and Innovative Services in a Professional Work Group. *Human Relations*, 45(7), 687-703. (1992)
28. Bovens, M., Zouridis, S.: From Street Level to System Level Bureaucracies: How ICT is Transforming Administrative Discretion and Constitutional Control. *Public Administration Review*, 62(2), 174-184. (2002)
29. Sinkovics, R.R., Bell, J., Deans, K.R.: Using information communication technology to develop international entrepreneurship competencies. *Journal of International Entrepreneurship*, 2(1-2), 125-137. (2004)
30. Tonchia, S., Tramontano, A.:Process Management for the Extended Enterprise Organizational and ICT Networks. Heidelberg, Germany: Springer. (2004)
31. Jean, R.B.: The ambiguous relationship of ICT and organizational performance: a literature review. *Critical Perspectives on International Business*, 3(4), 306-321. (2007)
32. Camarinha-Matos, L.M., Afsarmanesh, H., Galeano N., Molina, A.: Collaborative networked organizations – Concepts and practice in manufacturing enterprises. *Computers & Industrial Engineering*, 57(1), 46-60. (2009)
33. Čudanov, M., Jaško, O., Jevtić, M.: Influence of Information and Communication Technologies on Decentralization of Organizational Structure. *Computer Science and Information Systems Journal – COMSIS*, 6 (1), 93-108. (2009)
34. Chen, C.: Information Technology, Organizational Structure, and New Product Development—The Mediating Effect of Cross-Functional Team Interaction. *IEEE Transactions on Engineering Management*, 54 (1), 287-698. (2007)
35. Čudanov, M., Jaško, O.: Adoption of information and communication technologies and dominant management orientation in organisations. *Behaviour & Information*

Technology. Online first: http://www.informaworld.com/10.1080/0144929X.2010.499520. [Accessed 12th May 2011]. (2011)
36. Fertalj, K., Hoić-Božić, N., Jerković, H: The Integration of Learning Object Repositories and Learning Management Systems. *Computer Science and Information Systems Journal – COMSIS*, 7 (3), 387-407. (2010)
37. Roşu, M.S., Drăgoi G.: VPN solutions and network monitoring to support virtual teams work in virtual enterprises. Computer Science and Information Systems Journal – COMSIS, 8(1), 1-26. (2011)
38. Cudanov, M., Kirchner, K.: Knowledge Management in High-Growth Companies: A Case Study in Serbia. In Al-Shammari, M (ed.) Knowledge Management in Emerging Economies: Social, Organizational and Cultural Implementation, 227-248. Hershey, NY: IGI Publishing. (2011)
39. Jordan, P. J., Troth, A. J.: Emotional Intelligence and Conflict Resolution: Implications for Human Resource Development. *Advances in Developing Human Resources*, 4, 62-79. (2002)
40. Zheng, W., Yang B., McLean G. N.: Linking organizational culture, structure, strategy, and organizational effectiveness: Mediating role of knowledge management. *Journal of Business Research,* article in press - http://dx.doi.org/10.1016/j.jbusres.2009.06.005 (2009).
41. Hemila, J. Development of a Knowledge Support System for Dynamic Manufacturing Networks. Proceedings of the Second International Conference on Information, Process, and Knowledge Management, IEEE, Washington, USA, 106-109 (2010).
42. Pedersen, J. D.: A social network perspective on virtual organisations: social structure as enabler and barrier. *International Journal of Networking and Virtual Organisations,* 4(4), 431 – 445. (2007)
43. Hardy, B.: Growing significance of communities and collaboration in discovery and development. Future Medicinal Chemistry, 1(3), 435-449. (2009)
44. Raitman, R., Augar, N., Zhou, W.: Employing wikis for online collaboration in the e-learning environment: *Case study. Proceedings—3rd International Conference on Information Technology and Applications.* 142–146. Washington, DC, USA: IEEE Computer Society. (2005)
45. Pastore, S.: Social networks, collaboration and groupware software for the scientific research process in the Web 2.0 world. *Proceedings of the 7th WSEAS International Conference on Artificial intelligence, knowledge engineering and data bases*, 403–408, Stevens Point, USA: World Scientific and Engineering Academy and Society (WSEAS). (2008).

Mladen Čudanov works as assistant professor at the Faculty of Organizational Sciences, University of Belgrade. He has been visiting for one semester as an assistant professor in joint programs of iVWA from Germany and Jiangsu College of Information Technology from Wuxi and Zhuhai City Polytechnics from Zhuhai in China. His major research interests are ICT and organizational design, restructuring of business systems and organizational change. He has published more than 50 articles at scientific journals and conferences, and works as reviewer in several scientific journals, some on Thomson-Reuters SCI.

Prof. Gheorghe Săvoiu is a senior lecturer at University of Pitesti, Romania. He has served as a dean of Faculty of Finance – Accountancy. His main research interests are multidisciplinary application of economics, statsistics and business related disciplines. He has published more than 100 papers in scientific journals or conferences, and large number of books and monographic editions. He currently works as editor-in-chief for Econophysics, Sociophysics & Other Multidisciplinary Sciences Journal (ESOMSJ) , and as an editor/reviewer for several journals, some on Thomson-Reuters SCI.

Ondrej Jaško works as a full professor at Faculty of Organizational Sciences, Organization of Business Systems department, University of Belgrade has graduated at the Faculty of Organizational Sciences, University of Belgrade, in 1989. He got his master degree at the same Faculty in 1995 and PhD degree at the same Faculty in 2000. He was in position of vice dean, and in editing boards of our leading Journals in Management. His major research interests are organization theory and design, business systems, restructuring of business systems and organizational change. He has published more than 100 articles at scientific journals and conferences, and has coauthored several books and monographic editions.

Received: January 6, 2011; Accepted: July 5, 2011.

DOI 10.2298/CSIS110404070B

Effective Hierarchical Vector-based News Representation for Personalized Recommendation

Mária Bieliková, Michal Kompan, and Dušan Zeleník

Faculty of Informatics and Information Technologies, Slovak University of Technology, Ilkovičova 3,
842 16 Bratislava, Slovakia
{bielik, kompan, zelenik}@fiit.stuba.sk

Abstract. With amount of information on the web, users often require functionality able to filter the content according to their preferences. To solve the problem of overwhelmed users we propose a content-based recommender. Our method for the personalized recommendation is dedicated to the domain of news on the Web. We propose an effective representation of news and a user model which are used to recommend dynamically changing large number of text documents. We work with the vector representation of the news and hierarchical representation of similarities among items. Our representation is designed with aim to effectively estimate user needs and generate personalized list of items in information space. This approach is unique thanks its low complexity and ability to work in real-time with no visible delay for the user. To evaluate our approach we experimented with real information space of largest Slovak newspaper and simulated recommending.

Keywords: news, recommendation, hierarchical similarity, vector-based content representation, user model.

1. Introduction

Web has become perhaps the most important source of knowledge since day by day more and more information is available in the Web. But within the amounts of webpages, documents, pictures, videos or music we often miss what we really need. The amount of news which is published every hour becomes increasingly overwhelming. Furthermore, not only in the case of news, we often find duplicity or several information sources covering the same topics. The users are sensitive to the recency of information and their interests are also changing over time along with the content of the Web.

The logical consequence of such insufficiency is personalization of the information provided on the Web. One basic technique for personalization is the sorting information in the Web or in specific domains, to somehow improve user experience. This requires understanding of user needs and preferences and thus user modeling, which enables us to process information,

documents and even multimedia content on the web in new ways and with greater accuracy.

Moreover, there is another motivation which leads more and more companies to adapt to user needs. Overwhelmed customers are less likely to buy what they really want when they are not able to find it. On the other hand, everyone who takes interest in offers which are directly adapted to their needs is a potential customer. Suggesting products, videos, music or news has become important for web users but also for web service providers. Recommender systems have been designed and deployed on the Web to improve user comfort and increase profits. But there are still many areas where to improve recommender systems [1].

The problem with recommending items on the Web lies in searching for a combination of users and items. We often discover user interests by monitoring user behavior, i.e. what users search, what they comment on or what they have already purchased. Item recommendation includes two groups of approaches - collaborative recommendation and content based recommendation.

The idea of collaborative recommendation is in discovering similarities among users and subsequently recommending items according to similar user preferences. This approach uses similar users to search for items which have been displayed and could be interesting for the current user. The content based approach, which we focus on, discovers similarity between items based on their description and recommends items similar to those which have been already accepted by a given user (e.g., bought or viewed).

Both approaches introduce new challenges that must be addressed in practical applications, where a combination of both approaches is most often used to address particular problems of single approaches.

In this article, we present a content-based approach for recommending news, which is a very interesting topic since there are many readers who look for interesting information and want to read news comfortably. In every moment there are approximately 20 thousand active users. In combination with around 250 new articles each day we work with nontrivial real data. It makes it effectively impossible for every reader to read all new articles or search for interesting ones. We address this challenge by designing a recommender system in the news articles domain which reduces reader effort during search for articles covering specific topics.

We work with text documents and employ automatic extraction of keywords, named entities and other important terms. Text processing and similarity calculation are discussed in section 3. Our method is a combination of two separate approaches which were experimentally evaluated on the same dataset. First, we focus on similarity between text documents and combine it with another approach where we focus on the representation of document similarity and a user model. We describe our method for content-based recommendation in section 4 and present the evaluation of our work in section 5.

2. Related Work

Several approaches for recommendation and filtering have been proposed since early nineties. Two basic concepts are often mixed together to bring better results [5], [20]. Collaborative recommendation exploits social elements, when users are grouped into clusters based on their previous activity (preferences, habits, etc.). Recommendation is based on the assumption that items liked by other similar users are also potentially interesting for the current user (Fig. 1). This is also the main drawback of collaborative recommendation, as items not rated by a critical amount of users cannot be recommended. This is critical for high dynamic domains such as news recommending as with frequent changes of the information space recommended items can be obsolete before it achieves necessary popularity to be recommended. Despite this drawback, collaborative recommendation is probably the most used approach today [6,26].

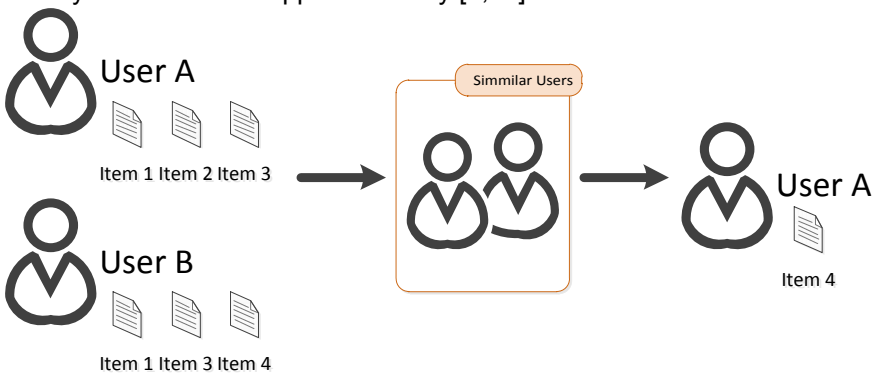


Fig. 1. Collaborative recommendation approach [26]. Users are grouped based on their similarity (visited sites, age, hobbies etc.). We recommend Item 4 to the *User A* because it is the only one item liked by other similar users and not visited by *User A*.

The main goal of content-based based personalization is to identify similar items – to create "clusters" of items instead of users based on associated features (attributes) of the processed items. Recommendation returns similar items to the items positively ranked before (Fig. 2). This type of recommendation is successful in well-structured domains like movies, news etc. [2]. One drawback of content-based personalization is the need for effective and sufficiently expressive item representation as effective similarity computation plays a crucial role (often also being highly domain dependent).
News recommendation and filtering is presently a current research topic with focus on two types of word similarity algorithms: statistical measures based on corpus and semantic distance based on hierarchical organization [30]. These can be further extended by paraphrase identification, vector approximation [31]. Standard methods and their extensions are widely used including n-grams, longest common subsequence, measuring shared syntax or text "fingerprints" [19], [23].

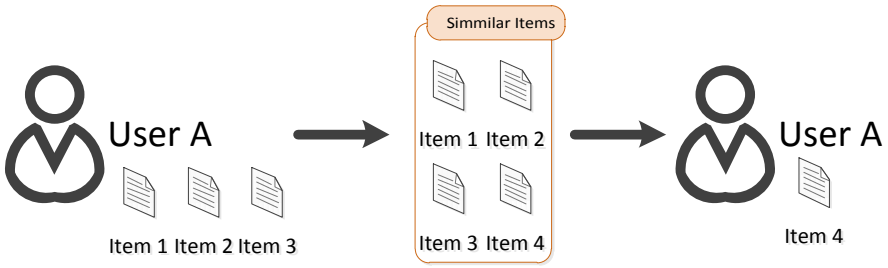


Fig. 2. Content-based recommendation approach. Items are grouped base the content similarity. Item 4 is recommended to the User *A* because of it is the only one item not visited in the group of liked similar items.

A lot of approaches use semantic nearness of documents [22], often based on the WordNet dictionary [32], what is a significant problem when non-English content is processed and fast computation is needed. Various metrics for similarity computation such as Cosine similarity, Euclidean distance, Jaccard Index and many others have been proposed [4].

Text similarity computation in news article domain is however not a developed area. There are various projects in the field of text summarization [9], text classification [24] or categorization [21], latent semantic analysis or SOM [32]. Vector representation of text is widely used in many approaches and by several systems [15], where it is not used for hierarchical or weighting purposes.

The definition of similarity in news recommendation systems has so far been somewhat unclear. For example, similarity can be defined based on news content (similar to plagiarism), or based on the “topic” of a news article, and its definition is extremely important when recommendation lists are created [36].

There are several content-based news recommendation systems. The OTS system [34] provides content-based and collaborative personalization based on association rules and a user interest table. The system works offline because of the large amount of processed data, and users have to choose interesting articles manually.

PURE [35] is designed to recommend medicine articles from a repository to which about 1000 new articles are added daily. Users have to create their own profile by defining interesting articles and the system then recommends new articles based on a classification taxonomy and Expectation-Maximization algorithm once per day.

NewsMe [33] is an adaptive recommendation system based on an open user model, which monitors 81 RSS channels from 21 sources. The core of recommendation method is based on the Nearest Neighbor algorithm. Brusilovsky has shown that a manually and explicitly populated open user model usually results in worse recommendation results in the news domain. Other systems also work with user location to provide location based recommendation [14].

GoogleNews handles thousands of users and articles per day. Three basic approaches are used for recommendation generation: MinHash clustering, Probabilistic latent semantic indexing and co-visitation counts [8], which have all been adapted to the Map-Reduce architecture employed by Google.

Daily Learner [2] is designed to provide adaptive news access based on implicit and explicit user feedback via standard a web interface and also via mobile devices. The user model consists of long-term and short-term user interests and uses k-NN and Naïve Bayes for recommendation.

Content-based approaches suffer from computation complexity, which can be addressed by adding more computational power or finding ways to reduce it. This is important with respect to article information value, which often decreases rapidly. Overspecialization is usually omitted. Recommendation methods should consider returning different numbers of articles for topics, when for example only football is recommended, and also work towards recommendation variability (i.e., prevent too homogenous recommendations) as users likely need not read 10 articles on a topic.

3. Representing News

When recommending news, based on their content, it is necessary to focus on the article content representation effectiveness, because of its high complexity. Besides effective article content representation, effective similarity representation and computation is also needed.

Two aspects must be considered - we try to maximize the useful information extracted from article content, and also need to minimize the amount of processed data (article words). These needs are combined in order to compute fast and accurate content similarity, which is used in personalized recommendation.

3.1. Construction of the Representative Article Vector

The main limiting factor for fast similarity estimation is compact and high precision article vector representation. We propose a vector, which consists of 5 basic parts described in Table 1.

Table 1. Vector representation of a news article.

Title	TF of title words in the content	Keywords	Category	Names/ Places

Title

Article vector comprises lemmatized words from the article's title. It consists of approximately 5 words (based on a 150,000 Slovak article dataset). We

estimate that the title should be a good describing attribute in most occurrences.

Term frequency of title words in the content
We use term frequency to estimate the confidence. If the title is abstract and does not correspond to article content (misleading titles etc.), we easily discover this situation. Term frequency is computed as:

$$tf_i = \frac{n_i}{\sum_k n_k} \quad (1)$$

where tf_i is the term frequency for term *i* (a term from the title) and n_i is the number of occurrences of term *i* in the document (article content) and $\sum_k n_k$ is the sum of occurrences of all terms in the document.

Keywords
Keywords consist of the 10 most relevant keywords for an article. Although many news portals store a list of keywords for every article, these are unfortunately usually at different abstraction level in different portals. This disadvantage can be solved by introducing a custom keywords list, which can be obtained via TF-IDF list calculated over the dataset (100,000 Slovak news articles from the news portal SME.SK). We can reduce the high dimensionality by removing any words except nouns and names. The keywords extraction approach can be easily replaced by any keywords extraction service, while the time consumption issue should be considered.

Category
We include a "tree-based" category vector with weights. This vector is constructed based on a portal specific category hierarchy (optional). The category is important for similarity search, when articles from one category are evaluated as more similar. The weight for every category is estimated as:

```
n=1
For i=|Category| downto 0 do
  weighti=1/n
  n=n*2
end
```

For example, let us consider three articles *A*, *B*, *C* and four categories *C1*, *C2*, *C3*, and *C4*, where the article *A* and *B* correspond to categories *C1*, *C2*, *C3* and article *C* to categories *C1*, *C2*, *C4*. Then the vector for every article is represented in Table 2 and Fig. 3. As we can see, articles *A* and *B* would be more similar then the article *C*. Values in the Table 2 are computed using proposed algorithm.

Table 2. Article Category Vector Representation.

	C1	C2	C3	C4
A	1/4	1/2	1	-
B	1/4	1/2	1	-
C	1/4	1/2	-	1

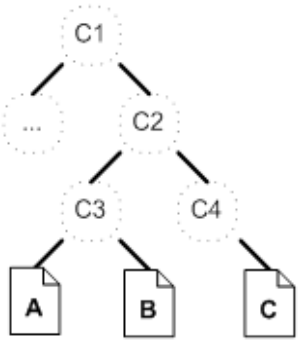


Fig. 3. Example hierarchy of news portal categories.

Names/Places
We include names and places extracted from article content. We extract names or place during the preprocessing stage since our method does not remove full-stops or uppercase letters. We identify names or places as words starting with an upper-case letter without a full-stop before it in the text stream (precision=0.934, recall=0.863). Alternatively, other name extractor systems for English language can be used [10].

3.2. Similarity Computation

For similarity calculation we employ cosine similarity and Jaccard index computation, which is widely used in information retrieval tasks [29]. The similarity of two articles *A*, *B* is computed as:

$$Jaccard\ index = \frac{|A \cap B|}{|A \cup B|} \quad (2)$$

Jaccard index was chosen based on our experiments. We used two datasets. The first were articles from news portal (1000 articles), which had assigned (by author) at last one similar article. The second was manually annotated dataset (100 articles). The expert assigned the similarity level (1-5) for each pair.

We computed list of similar articles for every article in the dataset and compared these to the similarity assigned by authors or experts. We calculated precision, recall and F-Score for every dataset and proposed

content representation. Results were compared [18] to standard text mining method TF-IDF as shown in Table 3.

Table 3. The Similarity metrics comparison.

Dataset	SME.SK		Manually annotated dataset		
Method	Proposed method	TF-IDF	Proposed method		TF-IDF
			Cosine similarity	Jaccard index	
Precision	0.165	0.091	0.700	0.843	0.511
Recall	0.202	0.117	0.816	0.818	0.587
F-Measure	0.182	0.102	0.753	0.870	0.546

As we can see there is a huge increase of precision and recall using proposed content representation. Poor results for the real "similar" dataset obtained directly from news portal can be explained by not accurate and incomplete information on the news portal. An author of a new article chooses none, one or two similar articles intuitively nowadays, and our method found probably more similar articles. Based on these results we decided to use Jaccard index because of the better results – computation complexity and F-Measure.

Every part of the article vector has its own global weight. By changing this weight we adjust the calculated similarity and its precision. Since we calculate article similarity online in a fast way, we can dynamically change these weights to obtain new results. For example if a user reads all recommended articles, we can change the weight of the category part to zero and recalculate similarity to obtain a new set of similar articles from different categories.

4. Personalized News Recommendation

Based on the proposed approach to article representation, we compute similarity between articles. To personalize news we work with large numbers of articles which are possibly interesting to individual users. We have devised a method to efficiently model user preferences and recommend interesting articles via hierarchical representation based on the aforementioned vector representation of articles and similarity computation. We designed this representation to support incremental addition of new articles and enable real time computation of article similarity.

4.1. Tree Representation to Store News

The main aim of our representation is to group articles using their similarity. A typical usage scenario is the retrieval of a list of similar articles to a selected article. Using a similarity matrix is unfeasible due to its high memory complexity and has difficulty of real-time recommendation with the large

number of articles in our dataset (we have more than 250 new news articles daily).

We use binary tree to represent similarity relations. Our motivation to use tree structure is low time complexity of storing and retrieving items and its ability to be created incrementally. In our approach we follow the simple concept where real articles act as leaf nodes. The hierarchy itself is generated over these articles and is used for representation of metadata to access similar articles (see Fig. 4).

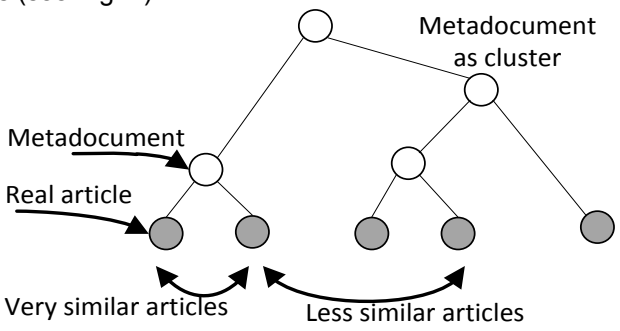


Fig. 4. The hierarchy is built over real articles. Connections are made based on similarity between articles. Closer articles are more similar than articles which are further apart. Metadocuments acts as clusters created by aggregating their child nodes.

Our relations hierarchy is incrementally built similarly to the hierarchy presented by Sahoo [25]. In our hierarchy, we rely on a repository which contains current articles and also assume that they are properly organized. A set of words extracted from articles and normalized is used as features to compute similarity between articles. Each node in the tree is labeled by a set of features, while edges in the tree represent the hierarchy which keeps similar articles nearby. We designed our representation as a hierarchy where:

- real articles are placed at the lowest level of the tree as leaf nodes,
- features are spread to upper levels of the hierarchy structure,
- similarity is stored in the hierarchy itself.

The hierarchy is built incrementally, when an article is downloaded as soon as it was published and added to the hierarchy structure. We use special nodes (metadocuments) to aggregate information (union of features) stored in the children documents. Metadocument itself is also represented as proposed vector of features used for real articles (title words, keywords, category, names and places). This is important especially for applications where real-time retrieval is needed (e.g., recommenders). With every new article we modify the tree to satisfy the rule that more similar articles are closer in its structure. Fetching similar articles is then only an issue of locating nearby nodes. The built hierarchy is large, but on the other hand it is compensated by the fast storage and retrieval processes.

When a new article is processed and ready to be added to the structure we:

- locate a place in the tree where to add the article,
- add the article at the correct place and adjust edges in the hierarchy,
- modify the rest of the structure which is affected by the new article.

Searching for the best position for a new article starts at the root of the tree. We proceed step by step through each node were every node has assigned vector representation which describe real articles occurring in the corresponding subtree. The decision on placing the article is made using the Jaccard's similarity (higher similarity wins). Tree traversal ends once a leaf node is reached or the calculated similarity is almost equal for each branch.

To insert a new article we split the branch at a designated place, where the article should be inserted, and append a new node for the new article. We also modify the parents of the new node by spreading information about the new article and aggregating features as needed (i.e., unify features as shown in Fig. 5).

We use all of the features extracted from news articles. To prevent too many of the features at the top of the tree (root), we discard selected features (features with low TF-IDF) when they are spread to the root. We presume that every feature is relevant, but features which are rare do not affect the decision process at the beginning of search. We reduce these rare features because their weight is low. We calculate the weight as the ratio of feature occurrence and occurrence of other features. Rare features are spread only if they reach a weight comparable to other features which is more likely to happen deeper in the tree.

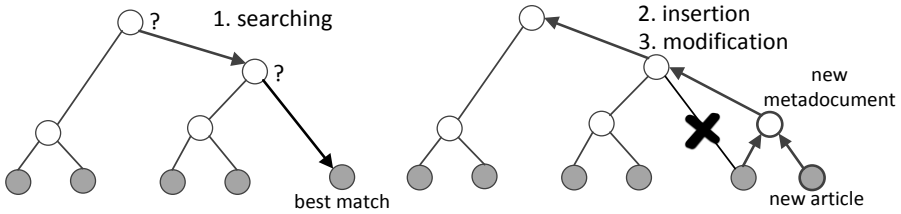


Fig. 5. In the first step we find the best place to insert a new article (left picture), which is added at the correct position in the second step. Metadata are created for the new node (right picture) and features are propagated to the root.

4.2. Modeling User Preferences Using Tree Representation

We discover interests of individual users by monitoring the activity of each reader. Similarly to the work by Carvalho et al. [7] we analyze records of users' activity. Articles that readers view are located in the hierarchical structure we described above, which contains relations between similar articles. We discover user interests by using the records of user activity and the hierarchy. During recommendation we constrain the number of recommended articles to a limited number in order to prevent information overload. We also seek equilibrium between recency and relevancy to maximize recommendation precision.

A prerequisite for our method is that each individual has some interests, what can be easily verified using the history of article views for particular readers and evaluating their interest in certain categories or sections of the news portal.

Similarly, there are identifiable fields of interests for each reader, for who it makes sense to explore other interests based on the calculated similarity between articles. We substitute this metadata (categories) created by editors with the hierarchy of similarity relations which provides our own metadata.

There are several options how to calculate similarity based on the content itself. There are also sophisticated methods, which are able to determine semantic similarity [11]. However, simple text similarity is often used in news recommendation with good results [17]. We use Jaccard's similarity to calculate article similarity with the aforementioned customization of vector-based article representation.

Figure 6 shows our method for discovering of reader interests. Here we present our understanding of user profile [13] in the hierarchical form. We use the tree structure created using the similarity of articles and records of user activity. We have a hierarchy of nodes which effectively represents similarity of real articles even without actual similarity calculation between particular pairs. Thick edges are paths from the displayed article to the root of the tree. Nodes where thick edges merge denote fields of interests.

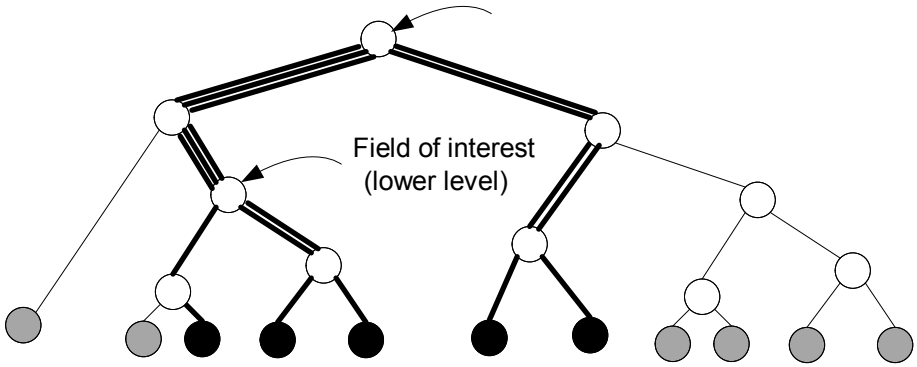


Fig. 6. User interests originate from black nodes representing already displayed articles. Bold lines connect displayed articles with the root node. Multiple lines represents the intensity of user interest in specific cluster of articles.

In this way, we discover interests for each user using their reading history. One user interest corresponds to one node in the tree which is used to define a hierarchical set of articles belonging to this interest (articles in the subtree).

When considering a reader's fields of interests we compare granularity of the interests depending on the depth where the node in the tree is located. Fields of interest that are closer to the leaves of the tree are more focused on a particular topic (e.g., articles about hockey). Fields of interest that are closer to the root correspond to more general topics (e.g., articles about sport).

4.3. Recommendation Generation

Recommending specific articles using our proposed hierarchical structure is a matter of selecting articles based on several reader interests. We first find the relevant interests for a particular reader, and calculate the relevance of the interest as the ratio of articles displayed from this interest and all articles belonging to the interest. Thus, we are able to sort interests and prepare for the selection of appropriate articles. Figure 7 shows the selection of interesting articles for a specific reader. Highlighted articles, which the user has read, are used to determine the relevance of his interest, while other articles are potentially interesting for the reader.

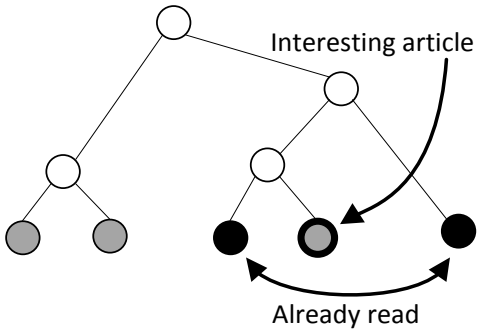


Fig. 7. Selecting interesting articles. Displayed articles are on the same branch as a candidate for recommendation.

In order to avoid overspecialization, we penalize the recommendation of overly similar interests, since otherwise the recommended articles would likely be very similar to those already viewed.

Because readers often have problems with long lists of recommendations [3], we choose articles that cover all relevant reader interests but are from distinct fields.

We also integrated time as an attribute in the computation of the list of recommendations, as it is a very important attribute, which indicates whether the interest that we discovered is outdated or current. Specifically in the news domain, time and recency are crucial. We store this additional information in our hierarchical structure and thus can find the latest article which was added for each branch of the tree. The time attribute is propagated as the maximum of two sub-branches. This way we efficiently identify the most recent article and the time when it was published. The interest is then as relevant as the last added article. We thus combine time relevancy and the content relevancy to create a combined list of recommended articles. Articles are not eliminated based on time recency, however aged articles are penalized. The tree contains all published articles (potentially interesting for some readers). We decide not to eliminate old articles because these could be still used for recommendation. We designed our method to work with huge amount of

articles with low complexity. Even higher number of articles stored in the tree does not affect the time needed to recommend.

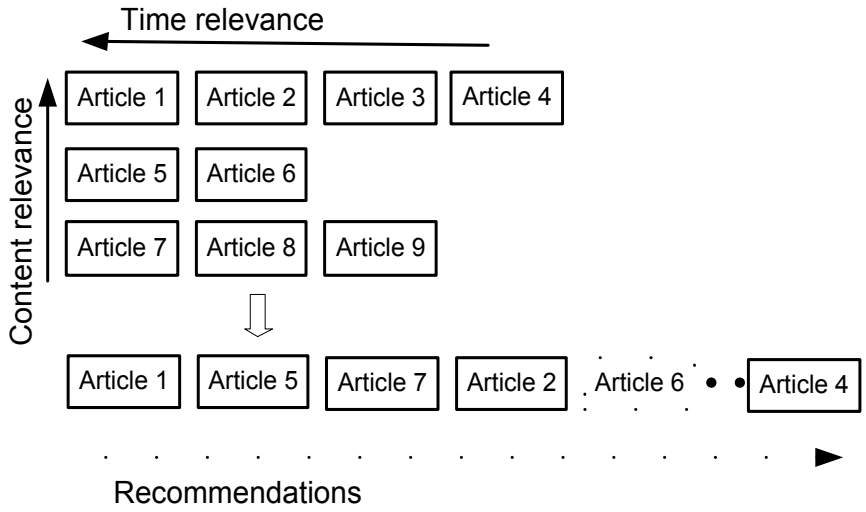


Fig. 8. Compiling the list of recommendations includes selection of most relevant articles based on content relevance and based on time. Articles in rows belong to the same field of interest; the most relevant interest is at the top. The most recent articles for each interest are on the left.

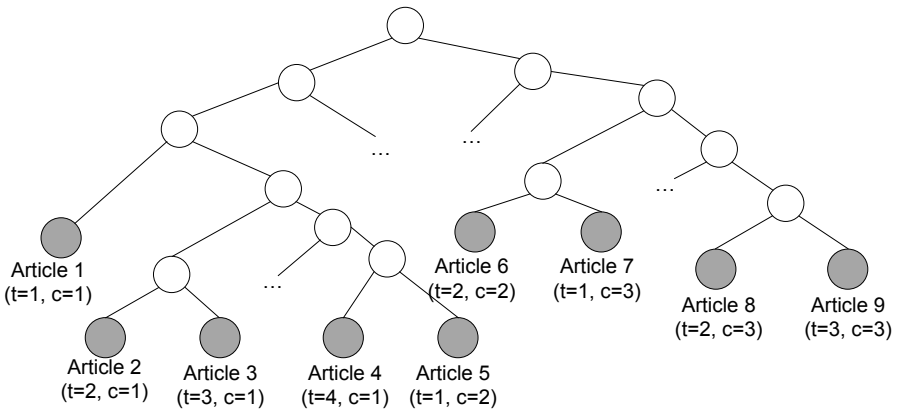


Fig. 9. Corresponding tree to the example of the recommendation compilation (Figure 8). Values *t* and *c* represent time and content relevancy for the specific user.

Figure 8 shows our method described in these steps:

- Selection of articles viewed by a reader
- Discovery of areas of interests in the tree
- Selection of unread articles for each interest
- Sorting articles by time for a particular interest

- Creation of an interest matrix
- Compiling the columns of the interest matrix into recommendations (Figure 8). Corresponding binary tree is shown in Figure 9.

The list of recommendations covers user interests but is designed to recommend at most 10 items per request to avoid choice overload [3] with articles covering several topics prioritizing recent articles.

We create the list of recommendations in real-time thanks to our similarity hierarchy which we use to represent relations between articles. In practice, we need to change the whole matrix only in cases where a user initiates a new session, or exhausts the recommendation list.

5. Evaluation

We proposed a method for effective content-based news recommendation. We designed this method to help readers to find interesting news and worked with more alternatives which we experimentally compared. We devised a data structure which we used to facilitate the sorting of text documents based on their similarity. Since the most interesting parameters for similarity computation are the used metrics, we experimented with different approaches to examine the effect on the quality of recommendation.

For our experiment, we used an interval of 14 days of user activity on the news portal SME.SK. We monitored articles which were viewed by users and have recorded over million unique users who had displayed over 12 million articles. We also added articles which were not viewed by users but were published during the interval. These additional articles belong to those which were not found by users but could be interesting. During preprocessing, we filtered out users who were not active and read only few articles during these 14 days, and also users who were too active (e.g., possibly the crawlers of search engines) thus reducing the noise caused by outlier users.

Our experiment was conducted using these records which were split into two subsets – one used for training our approach, and another for testing its accuracy. We used the training set to predict articles for every user who had been active in the test set. The predicted activity and real activity was then compared in via standard precision / recall metrics [12].

We started with simple bag of words which was used to represent and compare text documents. Later we worked with more sophisticated text preprocessing techniques, which have an important role in similarity search, because they can significantly reduce the search space. This part of the process is highly language dependent. Our experiments are performed in the Slovak language, which is one of the most complicated languages (declension of nouns, verbs etc.). The architecture of the system is variable, so preprocessing for Slovak language can be easily replaced by other languages and their methods (e.g., the Porter algorithm). Since maximum reduction of the article words dimensions and performance optimization are paramount, effective preprocessing played a critical role.

We first removed stop-words via a static list of words which are frequently used using TF-IDF under threshold output [28]. As the main part of the pre-processing of Slovak language articles we used text lemmatization, which cannot be algorithmically solved and needs to use a dictionary of lemmas. The result we received is a lemmatized (basic form) bag of words for every article.

Preprocessing of text is followed by the generation of the aforementioned vector representation for each article. In comparison to the whole article text which we firstly used to represent articles, we experimented with the vector components (metadata) and their weights (title words – 0.2, keywords – 0.3, category – 0.1, names and places – 0.4). We used different combinations of components to find the most appropriate components and how they affected the precision of our method during news recommendation. We placed these values as weight by experimentation. We used evolutionary algorithms as the approach for obtaining appropriate values. We placed fitness function as the similarity accuracy for articles.

We rebuilt the tree representation of article similarity and performed evaluation of our method for several combinations. We focused on the simple bag of words approach and then our vector representation where we observed the impact of names and places which were extracted from text on recommendation (see Table 4 for results).

We compared predicted user interest and articles viewed in the test set using a combination of category and section metadata, i.e. we did not compare exact articles but fields of interest which were covered by the combination of category and section where an article belonged. We chose this approach, because there were many articles on the same topic and users had no chance to read them all even if they were actually interested in the topic since one article on the same topic was often sufficient. To cover a topic we only compared the section and category of suggested articles and viewed articles. There were 420 valid options to choose this combination; therefore we guessed 1 combination out of 420 when predicting a topic.

Table 4. Comparison of the simple bag of words approach and vector based article representation during news recommendation.

Test interval [hours]		1 h	4 h	10 h	24 h	48 h
Bag of words similarity calculation	Precision	0.40	0.49	0.56	0.58	0.59
	Recall	0.71	0.60	0.44	0.32	0.25
	F1-Mesure	0.51	0.54	0.49	0.41	0.35
Vector based similarity calculation	Precision	0.35	0.42	0.52	0.54	0.66
	Recall	0.91	0.80	0.70	0.63	0.47
	F1-Measure	0.50	0.55	0.60	0.58	0.54

Our experiments indicate little to no improvement for shorter intervals. Predicting interests for 1 hour means to score only few articles. The average number of articles viewed by one user (from the set of active users) is 5 per

hour. Precision in such cases is very low, hence recall is high. We recommend 10 articles what means that there is greater chance to cover all interests for small intervals.

We observed more interesting improvement for longer intervals. When recommending articles to cover topics in 48 consecutive hours the precision of our method is relatively high. Besides the reduction of article representation complexity, we also achieved improvement in precision and recall for these longer intervals. In Figure 10 we present distribution of precision (number of correct recommendations).

We implemented proposed approaches in Ruby 1.9.2 language on Rails 2 platform. Experiments were executed on Dual Core 2.1 GHz, 2GB RAM. Data structure was simulated in MySQL database using native indexing. The average adding time of article into the proposed tree representation was approximately 2s with logarithmical tendency of time increment. By this achievement we fulfilled our aim to keep the time complexity low, even if the number of articles is high. Time complexity is important especially in this domain of news recommendation where recency of articles which are recommended counts.

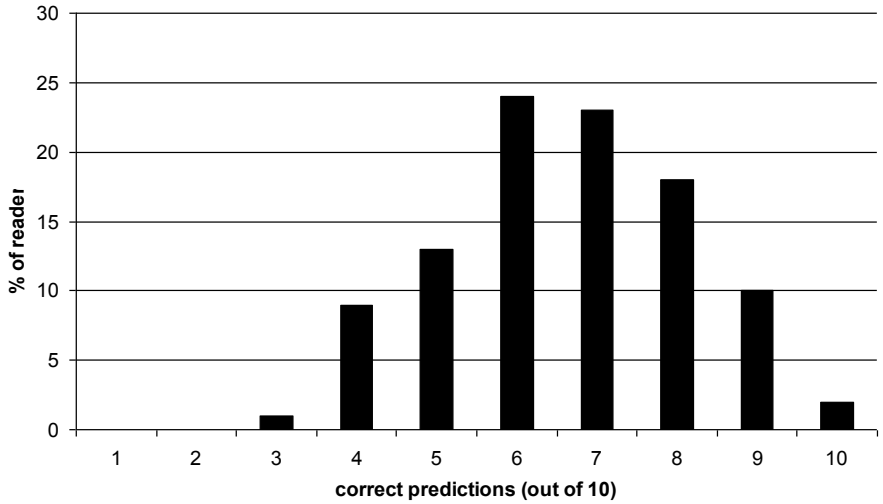


Fig. 10. Distribution of the readers by precision of the recommender. Shown values are for a 48 hour recommendation window over the test set.

6. Conclusion and Future Work

When recommending news several aspects have to be considered, such as the domain of news portals which is highly dynamic often requiring the use of content based methods. We compute article similarity at publication time instead of waiting for user ratings. Furthermore, the dynamic nature of the

news domain is also visible in the number of articles (i.e., huge amounts of data) and the need for freshness (i.e., quick response to new articles). Recommender systems often use tags and meta-tags assigned to articles by authors or editors to provide an extra source of information, which can improve similarity computation. However, the so called “tunnel - vision” should be avoided and recommenders should consider various article topics in order to avoid overspecialization.

Our contribution is the effective and expressive article content representation, and binary tree representation of article similarity and user interests, which in combination enables real-time content-based recommendation in highly dynamic domains. The proposed approach ensures that overspecialization and very similar articles recommendation are avoided.

Every article vector consists of five sub-vectors based on the article part used for their construction. Every part has its own weight, which can be dynamically changed to rearrange similar articles list to enable fast personalization. Based on this computed similarity the binary-tree is constructed incrementally. Our tree representation allows us to mine user interests with various granularities depending on tree depth, and also overcome “tunnel-vision” and consider the freshness of articles in relevance computation.

We performed several experiments with real news portal data and have shown that the proposed method brings improvement especially for longer window intervals. There was no or only little improvement in the one and four hour test window, which can be explained by the lack of user activity for such a short time period. Beyond this, the proposed approach reduces the complexity of the similarity computation process significantly as we can compute recommendations in real time.

Limitations of proposed approach include well known cold start problem, since we cannot perform recommendation without sufficient user preference data. A combination with recommendation based on behavior of large number of readers (e.g. news popularity) is viable alternative to overcome this limitation. We currently work on mixed collaborative content-based recommendation as a merge of content-based approach described in this paper and collaborative approach which employs several strategies for recommending news based on implicit positive and negative feedback while reading [27]. A combination of several approaches including multiple-interests in the group recommendation approach is natural step in such research. We also plan to include in recommendation new sources of information about user context [16] such as time, mood, and location.

Acknowledgement. This work was supported by grants No. VG1/0675/11, APVV 0208-10 and it is a partial result of the Research and Development Operational Program for the projects Support of Center of Excellence for Smart Technologies, Systems and Services, ITMS 26240120005 and Research and Development. Operational Program for the projects Support of Center of Excellence for Smart Technologies, Systems and Services II, ITMS 26240120029, co-funded by ERDF.

References

1. Adomavicius, G., Tuzhilin, A. 2005. Toward the Next Generation of Recommender Systems. IEEE Trans. on Knowl. and Data Eng. vol. 17, no. 6, 734-749.
2. Billsus, D., Pazzani, M. 2000. User Modeling for Adaptive News Access. User Modeling and User-Adapted Interaction, vol. 10, nos. 2-3, (Feb. 2000),147-180.
3. Bollen, D., Knijnenburg, B. P., Graus, M. 2010. Understanding Choice Overload in Recommender Systems Categories and Subject Descriptors. In Proceedings of 4th ACM Conf. on Recommender Systems. Barcelona, Spain (Sept. 2010), 63-70.
4. Borges, H., Lorena, A. 2010. A Survey on Recommender Systems for News Data. Smart Information and Knowledge Management, 129–151.
5. Bouras, C., Tsogkas, V. 2009. Personalization Mechanism for Delivering News Articles on the User's Desktop. In Proc. of the 4th int. Conf. on Internet and Web Applications and Services (May 2009). ICIW. IEEE Computer Society, Washington, DC, 157-162.
6. Burke, R. 2002. Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction 12, 4 (Nov. 2002), 331-370.
7. Carvalho, C., Jorge, A. M., Soares, C. 2006. Personalization of E-newsletters Based on Web Log Analysis and Clustering. In Proc. of the IEEE/WIC/ACM Int. Conf. on Web Intelligence. IEEE Computer Society, WDC, 724-727.
8. Das, A., Datar, M., Garg, A., Rajaram, S. 2007. Google news personalization: scalable online collaborative filtering. In Proc. of the 16th Int. Conf. on World Wide Web, ACM Press, New York, 271–280.
9. Díaz, A., Gervás, P. 2005. Personalization in news delivery systems: Item summarization and multitier item selection using relevance feedback. Web Intelli. and Agent Sys. 3, 3 (Jul. 2005), 135-154.
10. Finkel, J.,R., Grenager, T., Manning, Ch. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In Proc. of the 43nd Annual Meeting of the Assoc. for Comp. Ling. (ACL), 363-370.
11. Gabrilovich, E., Markovitch, S. 2007. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In Proc. of the 20th int. Joint Conf. on Artificial Intelligence, Hyder., India, 1606-1611.
12. Ge, M., Delgado-battenfeld, C. 2010. Beyond Accuracy: Evaluating Recommender Systems by Coverage and Serendipity. In Proc. of 4th ACM Conf. on Rec. Systems. (Barcelona, Spain, Sept. 2010), 257-260.
13. Grcar, M., Mladenic, D., and Grobelnik, M. User profiling for the web. Computer Science and Information Systems 3, 2 (2006), 1-29.
14. Chen, Ch., Hong, Ch., Chen, S. 2009. Intelligent Location-Based Mobile News Service System with Automatic News Summarization. Int. Conf. on Environmental Science and Information Application Technology, 527-530.
15. Ingvaldsen, J. E., Gulla, J. A., Laegreid, T., Sandal, P. C. 2006. Financial News Mining: Monitoring Continuous Streams of Text. In Proc. of IEEE/WIC/ACM Int. Conf. on Web intelligence. Washington, DC, 321-324.
16. Jancsary, J., Neubarth, F., Trost, H. 2010. Towards Context-Aware Personalization and a Broad Perspective on the Semantics of News Articles. In Proc. of 4th ACM Conf. on Rec. Sys. Spain, 289-292.
17. Kroha, P.,Baeza-Yates, R., 2005. News classification based on term frequency. In Proceedings of the 16th Conf. on Database and Expert Sys. Apps, 428–432.
18. Kompan, M., Bieliková, M., 2010. Content-Based News Recommendation. In Proc. of the 11th Conf. EC-WEB, Springer, 61-72.
19. Lukashenko, R., Graudina, V., Grundspenkis, J. 2007. Computer-based plagiarism detection methods and tools: an overview. In Proc. of Int. Conf. on Computer

Systems and Technologies (Bulgaria, June 2007). CompSysTech '07, vol. 285. ACM, New York, NY, 1-6.
20. Melville, P., Mooney, R., Nagarajan, J. 2002. Content-boosted collaborative filtering for improved recommendations. In Proc. of 18th National Conf. on Artificial intelligence (Edmonton, Alberta, Canada). AAAI, Menlo Park, CA, 187-192.
21. Mooney, R. J. and Roy, L. 2000. Content-based book recommending using learning for text categorization. In Proc. of the 5th Conf. on Dig Libraries, TX, USA.
22. Pandya, A., Bhattacharyya, P. 2005. Text Similarity Measurement Using Concept Representation of Texts. LNCS 2776, 678-683.
23. Parapar, J., Barreiro, Á. 2009. Evaluation of Text Clustering Algorithms with N-Gram-Based Document Fingerprints. In Proc. of the 31st European Conf. on IR Research on Advances in information Retrieval (Toulouse, France, April 2009). LNCS, vol. 5478. Springer-Verlag, Berlin, Heidelberg, 645-653.
24. Ramos, J. 2000. Using TF-IDF to Determine Word Relevance in Document Queries. Tech. rep., Departament of Computer science. Rutgers University.
25. Sahoo, N., Callan, J., Krishnan, R., Duncan, G., Padman, R. 2006. Incremental hierarchical clustering of text documents. In Proc. of the 15th ACM int. Conf. on Information and knowledge management, NY, USA.
26. Su, X. and Khoshgoftaar, T. M. 2009. A survey of collaborative filtering techniques. Adv. in AI. 2009.
27. Suchal, J., Navrat, P. 2010. Full text search engine as scalable k-nearest neighbor recommendation system. In Proc. of the AI in Theory and Practice 2010. WCC. IFIP AICT 331, Springer, Boston, 165-173.
28. Tata, S., Patel, J.M. 2007. Estimating the Selectivity of TF-IDF based Cosine Similarity Predicates. SIGMOD Record, Vol. 36, 7-12.
29. Tintarev, N., Masthoff, J. 2006. Similarity for news recommender systems. In Proc. of the AH'06 Workshop on Rec. Systems and Intelligent User Interfaces.
30. Wang, X., Ju, S., and Wu, S. 2008. Challenges in Chinese Text Similarity Research. In Proc. of the 2008 Int. Symposiums on Information Processing (May 23 - 25, 2008). ISIP. IEEE Computer Society, Washington, DC, 297-302.
31. Wang, Q., You, S. 2006. Fast Similarity Search for High-Dimensional Dataset. In Proc. of the Eighth IEEE int. Symposium on Multimedia (Dec., 2006). ISM. IEEE Computer Society, Washington, DC, 799-804.
32. Wermter, S., Hung, C. 2002. Selforganizing classification on the Reuters news corpus. In Proc. of the 19th Int. Conf. on Computer Ling. - Vol 1 (Taipei), 1-7.
33. Wongchokprasitti, C., Brusilovsky, P. 2007. NewsMe: A case study for adaptive news systems with open user model. In: Proc. of Int. Conf. on Autonomic and Autonomus Systems, 2007. ICAS07, 69.
34. Wu, Y., Chen, Y., Chen, A. L. 2001. Enabling Personalized Recommendation on the Web Based on User Interests and Behaviors. In Proc. of the 11th Int. Workshop on Research Issues in Data Engineering. RIDE. IEEE Computer Society, Washington, DC, 17.
35. Yoneya, T., Mamitsuka, H., 2007. Pure: a pubmed article recommendation system based on content-based filtering. In Proc. of Int. Conf. on Genome Inf. 18, 267-276.
36. Ziegler, C., McNee, S. M., Konstan, J. A., Lausen, G. 2005. Improving recommendation lists through topic diversification. In Proc. of the 14th Int. Conf. on World Wide Web (Chiba, Japan, May, 2005). ACM, New York, NY, 22-32.

Maria,Bielikova received her Master degree (with summa cum laude) in 1989 and her PhD. degree in 1995, both from the Slovak University of Technology in Bratislava. Since 2005, she has been a full professor, presently at the Institute of Informatics and Software Engineering, Slovak University of Technology. Her research interests are in the areas of software web-based information systems, especially personalized context-aware web-based systems including user modeling and social networks. She co-authored over 60 papers in international scientific journals and she is editor of more than 30 proceedings, six of them published in Lecture Notes in Computer Science series of Springer. She is a member of IEEE Computer Society, ACM and International Society for Web Engineering.

Michal Kompan is currently a doctoral student at the Institute of Informatics and Software Engineering, Slovak University of Technology in Bratislava. He received master degree in 2010 from the same university. His research interests are in the areas of personalized recommendation systems for single or group of users, user modeling and social networks.

Dušan Zeleník is a doctoral student in Software Engineering study program at the Slovak University of Technology in Bratislava. He holds a master's degree in Software Engineering (2010) from the same university. His research interests are in areas of personalized systems, recommendation systems, user modeling and context. He is a member of the PeWe group which joins researchers and their interests at aforementioned university.

Received: April 4, 2011; Accepted: October 10, 2011.

DOI 10.2298/CSIS101124020C

Concept Extraction and Clustering for Search Result Organization and Virtual Community Construction

Shihn-Yuarn Chen[1], Chia-Ning Chang[2], Yi-Hsiang Nien[3], and Hao-Ren Ke[4]

[1] PhD Candidate, Dept. of Computer Science, National Chiao Tung University
Hsinchu, 300, Taiwan
sychen@cs.nctu.edu.tw
[2] Master, Dept. of Computer Science, National Chiao Tung University
Hsinchu, 300, Taiwan
chianing.chang@gmail.com
[3] Master, Institute of Information Management, National Chiao Tung University
Hsinchu, 300, Taiwan
hugo3318@gmail.com
[4] Corresponding Author, Professor and Deputy Library Director, Graduate Institute of Library and Information Studies, National Taiwan Normal University
Taipei, 106, Taiwan
clavenke@ntnu.edu.tw

Abstract. This study proposes a concept extraction and clustering method, which improves Topic Keyword Clustering by using Log Likelihood Ratio for semantic correlation and Bisection K-Means for document clustering. Two value-added services are proposed to show how this approach can benefit information retrieval (IR) systems. The first service focuses on the organization and visual presentation of search results by clustering and bibliographic coupling. The second one aims at constructing virtual research communities and recommending significant papers to researchers. In addition to the two services, this study conducts quantitative and qualitative evaluations to show the feasibility of the proposed method; moreover, comparison with the previous approach is also performed. The experimental results show that the accuracy of the proposed method for search result organization reaches 80%, outperforming Topic Keyword Clustering. Both the precision and recall of virtual community construction are higher than 70%, and the accuracy of paper recommendation is almost 90%.

Keywords: information retrieval, concept extraction, document clustering, virtual community, social network analysis, bibliographic coupling.

1. Introduction

In the digital library era, information retrieval (IR) systems become essential for researchers to discover literature on particular subjects. Two major concerns when a researcher uses IR systems are how to filter out irrelevant documents and how to discover latest or significant documents.

IR systems have been used in digital libraries for decades, and various approaches, such as query reformulation/expansion [1], have been applied to improve the search quality. However, it is not easy for IR systems to correctly identify information need of every individual within a few queries. Personalized search is one solution, but analyzing collected personal information (e.g. search log, click-through history) has to be done in advance. Moreover, one well-known problem of IR systems is the representation of search results, which are usually in an item-by-item list view. Even for a veteran, such kind of representation takes time to filter out irrelevant documents. This study focuses on refining the organization of results by clustering. Search results are clustered by concepts; in addition, clusters and relationships between clusters are represented in a visual form. Users can have an overview of the search result concepts, easily identifying which groups are closer to their interests, and then look into the group for literatures.

Besides better representation of search results, researchers also use IR systems to find out latest or important documents in their research areas. These documents are either highly cited or written by significant authors in particular areas, and provide the latest research trends or basic knowledge of a research domain. Citation databases [2] such as SCI/SSCI/A&HCI, Scopus, and Google Scholar can provide information on highly cited documents. This study tackles this issue by constructing a social network of authors from their co-authoring relationship so as to construct virtual research communities and find out representative authors in a research area. Then the system will recommend documents to the authors themselves or those who are interested in an author's documents.

This study extracts concepts from textual information of documents. A concept comprises a group of terms and is constructed on the basis of word co-occurrence statistics. On the basis of the extracted concepts, two value-added IR services that exploit clustering, citation relations, and social information among authors are developed for solving the two preceding issues. The rest of this paper is organized as follows. Section 2 describes related works. Section 3 provides an overview of the value-added IR services proposed by this study, and elaborates the core mechanism for concept extraction and clustering. Section 4 proposes the service for the organization and visual representation of search results. Section 5 presents the service for virtual research community construction and paper recommendation. Section 6 describes the evaluation, and Section 7 draws a brief conclusion and future work.

2. Related Works

This section reviews three topics related to this study, including document clustering, social network analysis, and different representations of search results.

2.1. Document Clustering

Among the various types of clustering techniques [3], hierarchical and partitional clustering are the most common. Hierarchical clustering often provides better results, but the time complexity is the vital issue. In contrast, partitional clustering methods, such as k-means [4], are more efficient than hierarchical methods, but the proper number of clusters is hard to define in advance.

Though lots of efforts had been devoted to document clustering research, there are still various issues discussed in these years, including clustering techniques [5], similarity metrics [6], online Web document clustering [7], local optimal problem [8], knowledge aided document clustering [9], etc. One of the most common issues of document clustering is that there are too many words and phrases in the corpus, and a high-dimensional clustering is usually time-consuming [5]. Many approaches are proposed to improve traditional document clustering. Some try to select representative features to represent a document, and perform document clustering based on these features. For example, Iliopoulos et al. proposed TEXTQUEST[1] for Medline documents clustering [11]. TEXTQUEST is based on statistical treatment of words, and TF-IDF is used to extract important words from abstracts. Each Medline abstract is represented as a vector, which is used as input for the document clustering.

Some approaches use word clusters to improve the quality of document clustering. For example, Slonim and Tishby [12] use word clusters to capture the mutual information [13] about a set of documents. The experiments on a corpus comprising 20 newsgroups show that word clusters are helpful in document clustering. Another example is Topic Keyword Clustering [14]. Firstly, Topic Keyword Clustering retrieves keywords from documents and clusters them to obtain concept clusters. Secondly, it computes the similarity of each keyword cluster and each document, and finally assigns each document to the most similar keyword cluster to achieve document clustering. This study modifies Topic Keyword Clustering to organize search results and find out virtual research communities.

[1] http://www.textquest.de/

2.2. Social Network Analysis

Social Network Analysis (SNA) is a method based on sociometry for studying into social organizations, people relationships and interactions. Moreno quantizes people relationships and interactions, and represents the interactions and distances between people in a social network graph [15]. The vertices and edges in the social network graph represent social actors and the relationships of social actors, respectively. The properties of a social network graph can be described by global graph metrics and individual actor properties [16]. Global graph metrics describes the characteristics of a social network as a whole, e.g. average node distance, the number of components (fully connected subgraphs), clusters, etc. Individual actor properties cover actor distance, position in a cluster, etc. This graph representation allows applying graph theory [17] to analyze the tangled interaction of a society. Graph theory models objects and their relationships in mathematical structure for analyzing. In graph theory, objects are modeled as vertices, and the relationships of pairwise objects are modeled as directed or undirected edges, and the weight of edges represents the relationship degree. Graph theory is used in various applications, such as discovering implicit people relationships and exploring diseases propagation. Besides, with various metrics, researchers can know more about a social network and the significance of a member in the network.

Tyler et al. [18] use email to construct a social network in graph representation, and employ Betweenness Centrality to analyze communities in the network. Mika [19] proposes a system named Flink, which analyzes Web pages, emails, research papers and FOAF (Friend of a Friend) data to find out the social network of researchers, and visualizes the network and the ontology of a research area. Liu et al. [16] uses ACM, IEEE and JCDL (ACM/IEEE Joint Conference on Digital Libraries) publications as corpus for analyzing the co-author relationships; Liu et al. construct the social network of researchers and propose AuthorRank to compare with PageRank [20]. POLYPHONET uses participants of Japan Society of Artificial Intelligence conferences and their publications to explore their research interests and compute the context similarity for constructing social network relationships [21].

2.3. Different Representations of Search Results

The most common scenario that a user utilizes an IR system is – after submitting a query, the system returns a list of results to the user, and the user has to filter out unnecessary results in the list. Many systems pay a lot of efforts to provide better representation of results for users. Facet analysis is an example, which organizes results according to various types of criteria, such as publication date, topics, author, and publisher. Fig. 1 shows the

interface of a federated search system called MetaLib[2], which not only returns a list of search results, but also displays the facet criteria (including clustering) at the right side. Users can look into an interesting group by clicking any facet hyperlink.

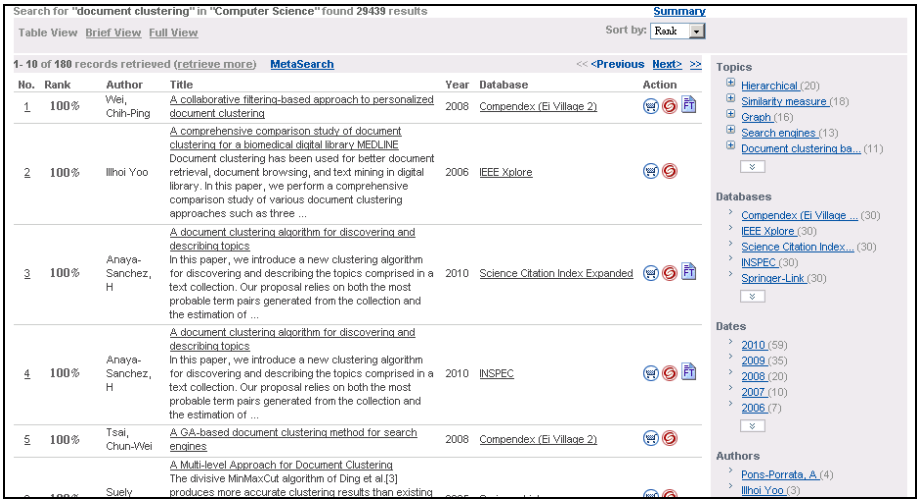


Fig. 1. Facet analysis in MetaLib.

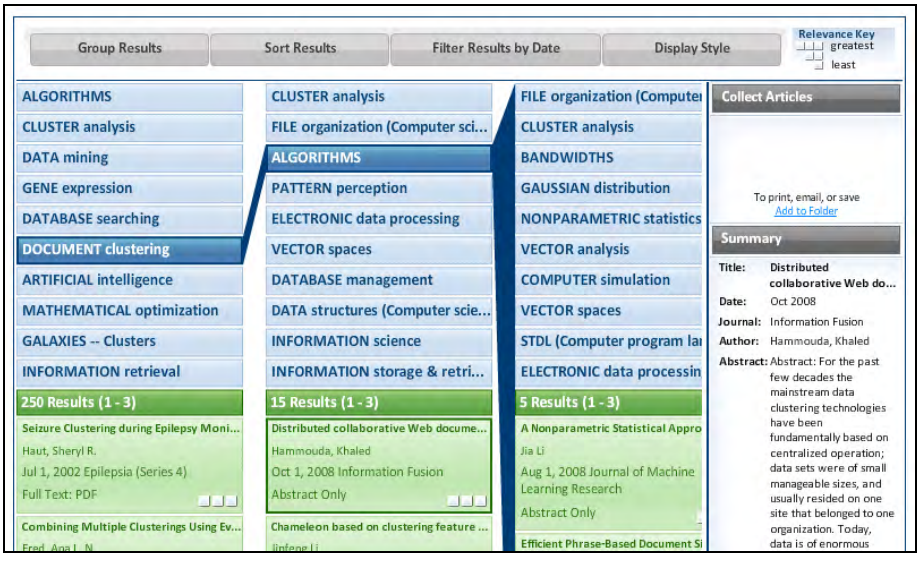


Fig. 2. Visual representation of search results in EBSCOhost.

[2] http://www.exlibrisgroup.com/category/MetaLibOverview

EBSCOhost[3] provides a visual representation of search results, as shown in Fig. 2. Users can click a result group to expand its sub-groups and result documents.

Google.com also provides "wonder wheel" view of search results (Fig. 3), which is similar to what EBSCOhost provides. Users can click a topic group on a wheel border to expand more detail results.

Grokker.com presents the results in a graphical view, as depicted in Fig. 4. Grokker.com organizes search results from Yahoo!, Wikipedia, and Amazon Books in a graphical interface. Each group is labeled with a keyword which helps users to figure out the concept of the group and to judge whether the group is relevant to their information need.

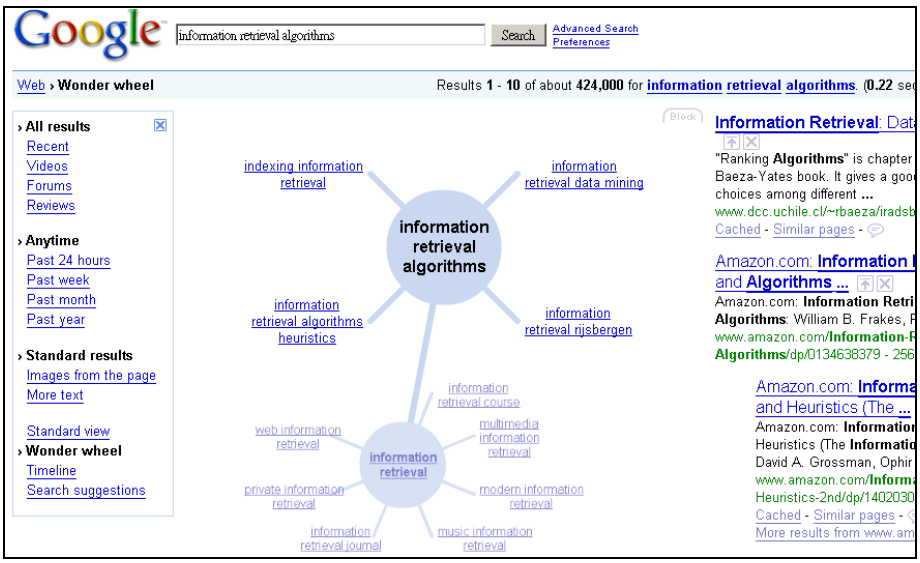


Fig. 3. Google "wonder wheel" view of search results.

These systems provide alternative organizations of search results to users, and save users' time on filtering out irrelevant results. However, these systems do not provide their users with sufficient relationship information among topic groups or virtual research communities.

[3] http://search.ebscohost.com/

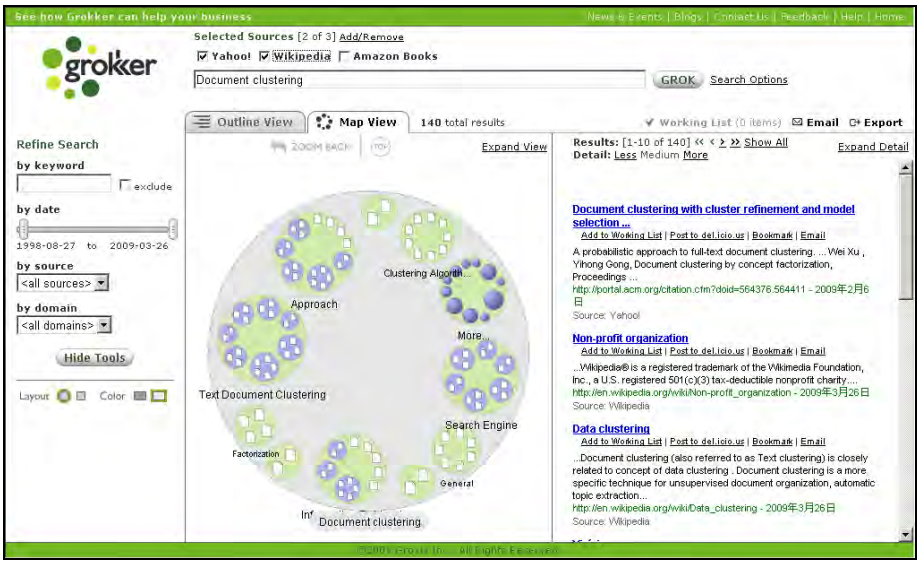


Fig. 4. Visual representation of search results of Grokker.com.

3. Value-Added IR Services Based on Concept Extraction and Clustering

First, this section describes the system architecture of the IR system with the two value-added services proposed in Section 4 and Section 5. Then, the core mechanism for the two value-added services, concept extraction and clustering, will be elaborated.

3.1. System Architecture

As shown in Fig. 5, an IR system stores document items. Each item has an internal representation and is indexed for efficient retrieval. For each new item, preprocessing tasks such as tokenization, lowercasing, stop words removing, part of speech (POS) tagging, and stemming [1] are conducted for extracting important features for the item; and these features, usually called index terms or keywords[4], and their weights (e.g. TF-IDF) associated with the item are inserted into the index. When a user submits a query, the system analyzes the user's information need and translates the information need into a formal query. The IR system then exploits the index to find relevant items, which are traditionally returned in an item-by-item list view.

[4] Hereafter, features, keywords, index terms, and terms are used interchangeably.

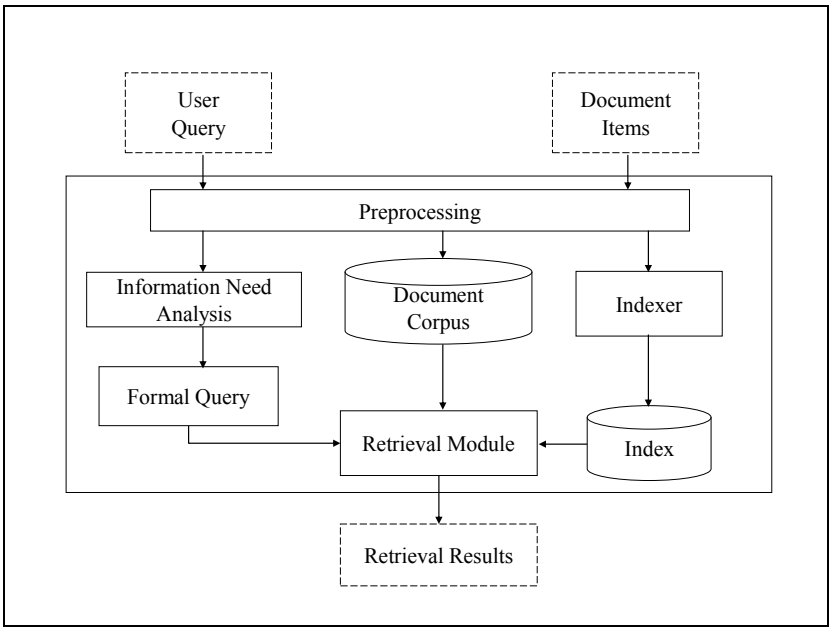


Fig. 5. System architecture of an IR system.

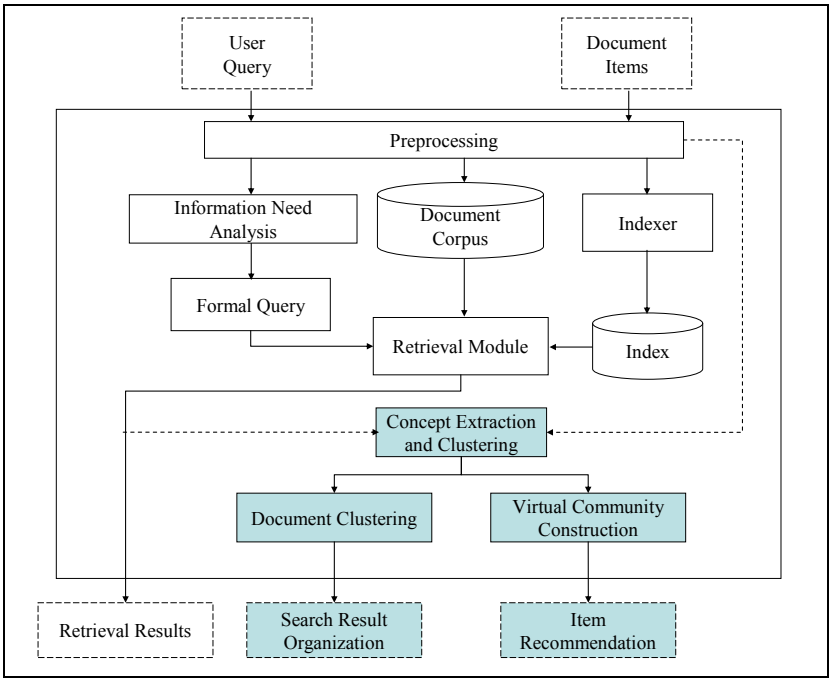


Fig. 6. System architecture of an IR system with proposed value added services.

This study aims at utilizing concept extraction and clustering to provide value-added services in an information retrieval system. These value-added services cover two aspects. The first reorganizes listed retrieval results into a graphical representation for intuitive browsing, and the second analyzes the relationships between authors of document items to generate virtual communities for item recommendation. The architecture of these proposed value-added services of an IR system is illustrated in Fig. 6. The following subsection will describe the core technique of these value-added services, concept-based extraction and clustering. The two value-added services will be explicated in Section 4 and Section 5, respectively.

3.2. Concept Extraction and Clustering

Concept extraction and clustering is the core technique of the proposed value-added IR services. A concept comprises a group of terms and is constructed on the basis of word co-occurrence statistics. The underlying assumption is that words co-occurring in a passage (such as document or paragraph) are likely to be in some sense similar or related in semantics [1]. The following subsections explicate how to conduct concept clustering, which is based on Topic Keyword Clustering [14].

3.2.1. Computing Semantic Relationship

The semantic relationship of two terms is based on their co-occurrences in a certain passage, for example, co-occurrences in the same paragraph. Different applications may choose different types of passages to judge word co-occurrences; furthermore, different applications may choose different methods to compute the semantic relationship of two terms. This study considers the co-occurrences in the same sentence and a few bibliographic fields like *title* and *keyword* fields. Instead of mutual information (MI), which is used in Topic Keyword Clustering, this study exploits Log Likelihood Ratio (LLR) [22][23] for computing the relationship between terms *i* and *j*, r_{ij} and keeps the confidence of r_{ij} larger than 99.9%. The main reason this study uses LLR is that MI is more proper for judging the independent degree rather than relative degree of two terms [34].

3.2.2. Concept Extraction and Clustering

After semantic relationships of terms (features) are computed, the concept extraction process is applied to construct concept clusters. The detailed processes are described below.

3.2.2.1. Feature Nethwork Construction

1. Constructing draft feature network

In the draft feature network, each vertex represents a feature, and the weighted edge represents the relationship of two features, which is computed in 0. Fig. 7 shows the idea of future network construction, and Fig. 7 (a) represents the draft feature network.

2. Removing insignificant edges and vertices

Insignificant edges and vertices may cause noises during concept extraction. To remove insignificant edges, edges with weight less than the threshold, which is defined as the average weight of edges in the network, are removed. For experimental comparison of Topic Keyword Clustering and our approach, this study follows the parameter and threshold settings of Topic Keyword Clustering.

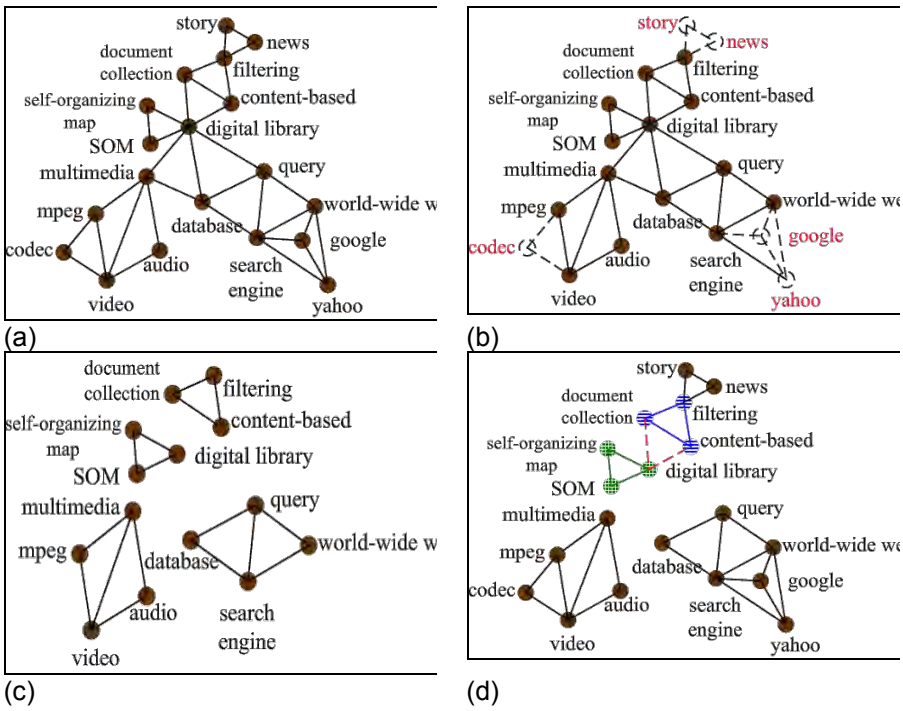


Fig. 7. Example of concept extraction and clustering [14]

The weight of each vertex is CW_i, where r_{ij} is the relationship of two term vertices V_i and V_j computed in 3.2.1, and m is the number of vertices connected to V_i,. For example, in Fig. 7 (a), the vertex "story" is connected by "news" and "filtering", so the weight of "story", CW_{story}, is the average weight

of the two connected edges (i.e. the average of relationships of story-news and story-filtering).

$$CW_i = \frac{\sum_{j=1}^{m} r_{ij}}{m} \quad (1)$$

To remove insignificant vertices, the threshold is the average weight of all vertices. Fig. 7 (b) shows the network after removing insignificant edges and vertices.

3.2.2.2. Concept Extraction and Concept Cluster Generation

1. *k*-Nearest neighbor (knn) grouping [24]

Each vertex and its *k* nearest neighbors can be grouped as a connected graph, named candidate feature unit. Different values of *k* affect the number of candidate feature units. A larger k results in fewer units, and each unit contains more features. If there are too many features in a candidate feature unit, its concept may not be well expressed, and the significance represented by each feature is also reduced. This study follows the setting of Topic Keyword Clustering and lets *k*=2. Fig. 7 (c) shows an example containing four candidate feature units.

2. Candidate Feature Subgroups Generation

Except the edges that cross two candidate feature units, out-linked edges from each candidate feature unit are recovered to form candidate feature subgroups. The weight of a subgroup is the sum of edge weights W_{G_m}, where G_m represents a candidate feature subgroup *m*, and r_{ij} is the weight of an edge.

$$W_{G_m} = \sum_{r_{ij} \in G_m} r_{ij} \quad (2)$$

3. Candidate Feature Subgroups Merging

The goal of subgroups merging is to reduce the number of subgroups by a greedy algorithm. The two most inter-connected subgroups are merged iteratively till the inter-connected degrees of all subgroup pairs are less than the threshold (empirically chooses 0.5 in this study). The degree of inter-connected subgroups, $RI(G_i, G_j)$,is computed as follows.

$$RI(G_i, G_j) = \frac{\left| W_{E(G_i, G_j)} \right|}{\left| W_{G_i} \right| + \left| W_{Gj} \right|} \quad (3)$$

$W_{E(Gi, Gj)}$ is the sum of the weights of the inter-connected edges of the two candidate feature subgroups, G_i and G_j. If *RI* is larger than the threshold, it indicates that the two subgroups are significantly related and should be merged. For example, in Fig. 7 (d), the two subgroups {digital library, SOM, self-organizing map} and {document collection, filtering, content-based} are merged.

$$AS(G) = \frac{\sum_{r_{ij} \in E(G)} r_{ij}}{|E(G)|}$$

$$CD(G) = \frac{(E(G))}{|V(G)| \times |V(G)-1| / 2} \tag{4}$$

$$CW = CD(G) \times AS(G)$$

4. Concept Clusters Generation

Subgroups with more features usually cover more documents, and a different amount of documents may affect the accuracy of clustering. To reduce the number of different documents in each subgroup, the concept weight of each subgroup (*CW*) should be considered.

CW is computed by multiplying the average edge weight (*AS*) and the connected density (*CD*). If the concept weight of a subgroup is less than the average or the number of features is larger than the average (as the threshold setting in Topic Keyword Clustering), the most non-related feature is removed. And, the final results are the concept clusters.

4. Search Result Organization

The first value-added IR service utilizing the concept extraction and clustering method in Section 3 lies in presenting search results in a graphical form. To demonstrate this service, a corpus containing about 500 "Information Retrieval" related documents from CiteSeer[5] is used. Three bibliographic fields, *title*, *abstract* and *citations*, are collected for each document in this corpus. The average length of abstracts is about 1000 words.

4.1. Concept Extraction

Before concept extraction, pre-processing is applied to *title* and *abstract* fields, and terms appearing in more than 8% of documents or less than three

[5] http://citeseer.ist.psu.edu/

times are filtered out. The concept extraction algorithm mentioned in Section 3.2 is then applied to the remaining terms for constructing concept groups.

This value-added service only considers the co-occurrences in the same sentence of the *title* and *abstract* fields. As mentioned previously, Log Likelihood Ratio (LLR) is used for computing the relationship between terms *i* and *j*, r_{ij}; in addition, to emphasize the importance of the *title* filed, terms co-occur in the *title* field are doubly weighted.

4.2. Document Clustering and Post-processing

After concept groups are generated, this study represents each document as a vector. Each element in the vector represents the semantic similarity of a document and a concept group. Documents can then be grouped by clustering these semantic similarity vectors. The detail of document clustering and the post processing are described in the following subsections.

4.2.1. Semantic Similarity Vector and Document Clustering

According to Section 3.2, *M* concept clusters can be generated and there are *n* features in total. A document D_j and a concept cluster C_m can be represented as vectors by these *n* features.

$$D_j = (w_1, w_2, w_3, ..., w_n) \text{ where } j = 1, 2, 3, \ldots, N$$

w_i: the weights of the feature *i* (estimated as TF-IDF)

N: # of documents in corpus

$$C_k = (t_{k1}, t_{k2}, t_{k3}, ..., t_{kn}) \text{ where } k = 1, 2, 3, \ldots, M \quad (5)$$

$$t_{ki} = \begin{cases} 1, & \text{If term } i \in C_k \\ 0 & \text{otherwise} \end{cases}$$

This study uses cosine similarity to compute the similarity between a document and a concept cluster. In this manner, each document D_j can also be represented as a vector of semantic similarities with all concepts, SD_j. SD_j also represents the degree of how well a document covers a concept.

$$SD_j = \left(sim(D_j, C_1), sim(D_j, C_2), sim(D_j, C_3), ..., sim(D_j, C_M)\right) \quad (6)$$

Bisection *k*-Means algorithm [25] is then applied to perform document clustering. After document clustering, post-processing, including labeling, cluster relationship construction, and visualization, are applied.

4.2.2. Labeling

Visualization of document clustering usually applies labeling to help users understand the concepts and meanings of each cluster. In this study, the following steps are applied to choose proper labels:

- Select the top 10 weighted features from each cluster;
- In the 10 features, nouns and noun phrases are selected;
- Choose the top 2 weighted candidates as the final labels.

4.2.3. Cluster Relation Construction

Besides labeling, the relation of documents is also helpful to understand relationships of clusters. As CiteSeer preserves the references made by each document, this study exploits the idea of bibliographic coupling [26] to establish the relation between clusters. The similarity of documents cited by two documents can be computed, and then the similarity can be used to construct the citation relation of the two documents. Furthermore, the citation relations of clusters can be computed and used in visualization. The similarity of cited documents for two documents is computed as follows:

1. Hyperlinks (URL) of cited documents

URLs of cited documents are considered in higher priority. In CiteSeer, most documents cited by a document have hyperlinks, and can be accessed through these links. A hyperlink URL can be treated as the unique id of the document. Then the citation similarity of two documents, $linksim_{i,j}$, can be computed, where d_i and d_j are two documents, and $link(d_i)$ and $link(d_j)$ are links to their cited documents, respectively.

$$linksim_{i,j} = \frac{\left| link\left(d_i\right) \cap link\left(d_j\right) \right|}{\min\left(\left| link\left(d_i\right) \right|, \left| link\left(d_j\right) \right| \right)} \tag{7}$$

2. Titles of cited documents

If the URL of a cited document is lost or there are multiple different hyperlinks of a cited document, the abovementioned citation similarity of two documents is affected. To fix the error, titles of cited documents should be considered. After pre-processing and vectorization, the similarity of titles, *textsim*, can be computed by cosine similarity.

The preceding two similarities, *linksim* and *textsim*, are linearly combined to obtain the citation similarity *DR(i,j)* of two documents d_i and d_j, and this study choose α = 0.5 empirically.

$$DR(i,j) = \alpha \cdot \left(linksim_{i,j}\right) + (1-\alpha) \cdot \left(textsim_{i,j}\right) \quad (8)$$

Citation relation of clusters is computed based on the citation similarity of documents. If C_m contains d_i, C_n contains d_i, and the citation similarity of d_i and d_i, *DR(i,j)*, is larger than threshold, θ (0.5, empirically chosen in this work), the citation count of C_m and C_n is increased by 1. Then, the cluster similarity of C_m and C_n, *CR(m,n)*, can be computed.

$$CR(m,n) = \frac{\#\text{ of document pairs } \left(d_i, d_j\right) \text{ that } DR(i,j) > \theta}{\max\left(|C_m|, |C_n|\right)} \quad (9)$$

4.2.4. Cluster Visualization

Two visualization methods are exploited to represent the clustering result, and Table 1 shows the comparison of the proposed approach and other existed approaches mentioned in Section 2.3. The first method is used for inner document cluster representation, and Fig. 8 is an example. Each circle in Fig. 8 represents a cluster, and the more documents a cluster contains, the larger its radius is.

Table 1. Visual representation comparison of the proposed approach and others.

Concept label	Inner Cluster	Inter Cluster
Our Method	Radius of circle for cluster size. Different color for inner-cluster similarity	Radar graph to show cluster relations. Cluster relation degrees are labeled.
MetaLib	Numeric label for cluster size.	No relations are shown
EBSCOhost	Topic grouping only	Hierarchy-like structure for cluster relations
Google wonder wheel	Topic grouping only	Network-like graph for cluster relations
Grokker.com	Radius of circle for cluster size.	Hierarchy-like structure for cluster relations

Furthermore, the color of a pie represents the similarity / homogeneity of documents in a cluster, the darker the more similar / homogeneous they are.

Besides, the text above a circle is the generated labels of a cluster. In Fig. 8, the document similarity of the cluster “collaborative filtering” is the highest, followed by “speech Recognition”, “self-organizing map”, and “decision tree” is the lowest.

All the approach mentioned in Section 2.3 and the proposed approach provide document clustering/grouping according to the topics, but only MetaLib, Grokker.com and the proposed approach provides the amount information of each clusters/groups. MetaLib uses numeric label; Grokker.com and our proposed approach uses graphical view (the size of circles) to represent the cluster/group size. Besides, only our proposed approach uses different colors to represents the inner-cluster similarity.

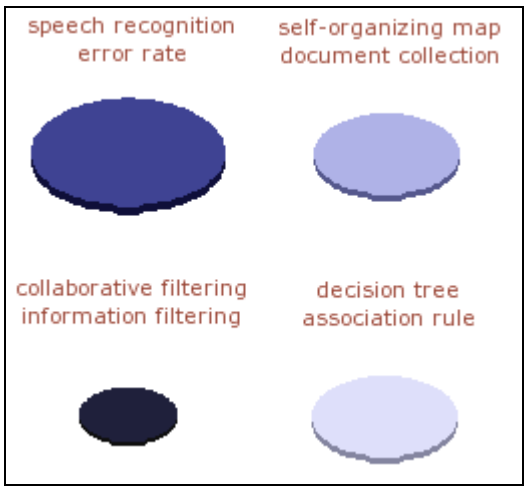


Fig. 8. Chart for cluster visualization.

The second method is a radar map for representing the relationships between clusters. In Fig. 9, the labels, “self-organizing map” and “document collection”, represent the concept of a cluster. Each apex represents a related cluster. The relation degree of two clusters can be realized easily by the radar map; in this example, the “text categorization and relevance feedback” cluster has the highest degree relation with the cluster labeled with “self-organizing map” and “document collection”.

As shown in Table 1, except MetaLib, all the approaches in Section 2.3 and the proposed approach provide the information of topic cluster/group relationship. EBSCOhost and Grokker.com model this information in a hierarchy-like representation, and Google wonder wheel models in a network-like representation. However, only our approach provides the relationship degree to users in a radar graph representation.

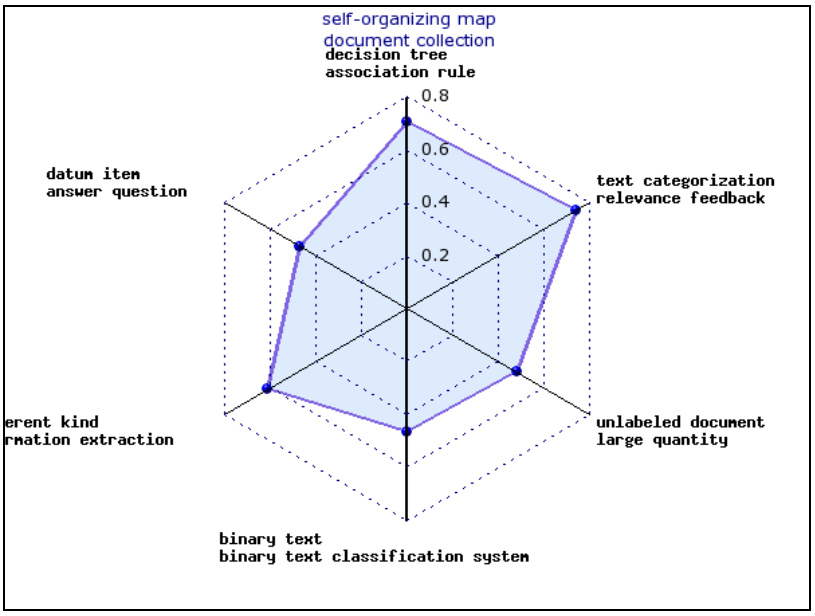


Fig. 9. Radar map for cluster relation representation.

5. Virtual Community Generation and Paper Recommendation

The second value-added IR service concerns the generation of virtual research communities and the recommendation of papers based on the virtual communities. To demonstrate this service, a corpus comprising 226 research papers from the institutional repository[6] of National Chiao Tung University (NCTUIR) is used, and four bibliographic fields, *title*, *abstract*, *keyword* and *author*, are collected for each paper in this corpus.

5.1. Author Model

Each term used in a paper is assumed to relate to the paper's concept, and also has a relationship to each author. The author model uses the relationship of terms and authors to model the research interests of authors. The relationship of terms and authors can be computed by TF-IAF (Term Frequency-Inverse Author Frequency). [27]

$$tf_{ij} = freq_i, \text{frequency of term } i \text{ associated with author} \quad (10)$$

[6] http://ir.lib.nctu.edu.tw/

j

$iaf_i = \log_2 \frac{N}{n_i}$, where *N*: # of total authors,

n_i: # of authors who use term *i*

$$w_{ij} = \begin{cases} tf_{ij} \times ia & \text{, if term } i \text{ is associated with author } j \\ 0 & \text{, otherwise} \end{cases}$$

$U_j = (w_{1j}, w_{2j}, ..., w_{mj})$, where m is the number of terms

After computing TF-IAF, the research interests of an author *j* can be represented as a vector, U_j. The author model is the collection of authors, and can be represented as a matrix, *U*.

$$U = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ \vdots \\ U_N \end{bmatrix}, \text{where } N \text{ is the total number of authors} \quad (11)$$

5.2. Concept Clustering and Labeling

This value-added service only considers the co-occurrences in the same sentence of the *title* and *abstract* fields. As mentioned previously, Log Likelihood Ratio (LLR) is used for computing the relationship between terms *i* and *j*, r_{ij}; in addition, to emphasize the importance of the *title* filed, terms co-occur in the *title* field are doubly weighted.

The method introduced in Section 3.2 is used to gain the concept clusters. This value-added service considers the co-occurrences in the same sentence, and in the same *keyword* and/or *title* fields of a research paper. As mentioned previously, Log Likelihood Ratio (LLR) is used for computing the relationship between terms *i* and *j*, r_{ij}; in addition, to emphasize the importance of the *title* and *keyword* fields, the semantic relationship of terms co-occurring in the *title* and/or *keyword* fields equals to 1.0.

After concept clustering, each research cluster C_k can be represented as a vector.

$C_k = (t_{k1}, t_{k2}, t_{k3}, ..., t_{kn})$, where n is the number of terms, (12)

and

$$t_{ki} = \begin{cases} 1, & \text{if term } i \in C_k \\ 0, & \text{Otherwise} \end{cases}$$

Similar to Section 4.2.2, labeling is also applied for helping users to realize the concept of each research cluster. Table 2 shows the labels, keywords, and keyword weights of a few example research cluster.

Table 2. Concept clusters and their keywords and relative weights.

Concept label	Keywords	Keyword weight
Mobile Computing	MANET	351.1895
	mobile ad hoc network	152.5828
	route	140.446
	mobile computing	126.5852
	wireless network	95.5504
Genetic Algorithm	genetic algorithm	97.3025
	dynamic linkage discovery	66.415
	particle swarm optimization	53.332
	GA	41.3458
	economic dispatch	27.0827
Neural Network	neural network	72.7728
	optimization problem	49.7963
	constraint	43.1446
	energy function	38.9312

5.3. Virtual Community Construction

5.3.1. Author Social Network

Author social network is constructed from the co-author relationship, and the relative degree of two authors is calculated by Jaccard coefficient [28]. First, the author-paper relationship can be represented as a matrix, W_{ij}. As illustrated in Fig. 10, there are three authors, $u_1 \sim u_3$, four papers, $d_1 \sim d_4$, and each edge connecting an author and a paper represents the author-paper relationship.

$$W_{ij} = \begin{cases} 1, & \text{if user } i \text{ is one of the authors in paper } j \\ 0, & \text{Otherwise} \end{cases} \quad (13)$$

The co-author relationship, S, can be computed by $S = W \times W^T$, and each element S_{ij} represents the number of papers co-authored by i and j. Fig. 11 is the co-author relationship of Fig. 10.

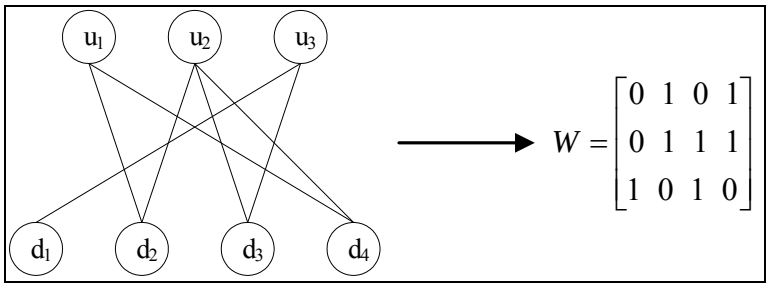


Fig. 10. The relationship graph and matrix of authors and papers.

$$S = W \times W^T = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 2 & 0 \\ 2 & 3 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

Fig. 11. Co-author matrix of Fig. 10.

This study applies Jaccard coefficient to measure the similarity between two authors, and it is defined as the ratio of the number of co-author papers over the number of the union of each author's publication.

$$J(U_i, U_j) = \frac{S_{ij}}{|S_i| + |S_j| - S_{ij}} \tag{14}$$

Where $|S_i|$ and $|S_j|$ represents the amount of papers authored by author i and j, respectively. Applying Jaccard coefficient can also normalize the main diagonal of S to 1. Fig. 12 is the S after applying Jaccard coefficient, named S'.

As the co-authoring relationship may affect each author too much, a parameter α ($0 \le \alpha \le 1$) can be added to adjust the co-author relationship matrix, S'. If α is closer to zero, the co-authoring relationship exerts less effect to each author. The adjusted co-author relationship matrix is R, and each element R_{ij} can be represented as follows.

$$R_{ij} = \begin{cases} 1 & \text{, if } i = j \\ \alpha \times S_{ij} & \text{, otherwise} \end{cases} \tag{15}$$

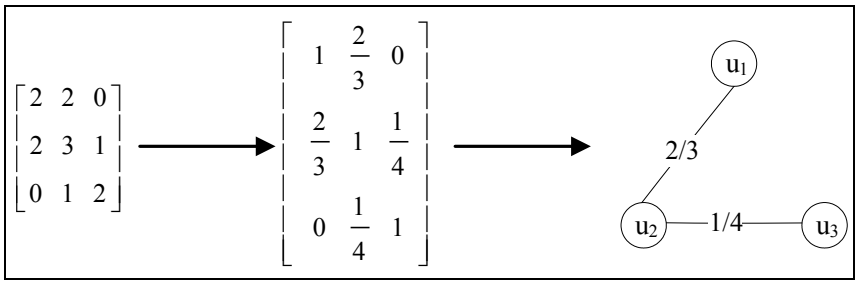


Fig. 12. Co-author matrix normalized by Jaccard coefficient and relative social network graph.

5.3.2. Author Clustering and Paper Recommendation

This study models authors and terms using the author model. The author model *U* is represented as an *N*×*m* matrix, where *N* is the number of authors and *m* is the number of terms. Each row in *U* shows how an author uses terms in his/her research papers. For taking the co-author factor into account, the updated author model *U'* is the production of the adjusted co-author relationship matrix (*R*) and *U*. Each row in *U'* represents the relationship of an author and all terms when the co-author relationship is considered.

$$U = \begin{bmatrix} w_{11} & \cdots & w_{1m} \\ \vdots & \ddots & \vdots \\ w_{N1} & \cdots & w_{Nm} \end{bmatrix}$$

$$U' = R \times U = \begin{bmatrix} R_{11} & \cdots & R_{1N} \\ \vdots & \ddots & \vdots \\ R_{N1} & \cdots & R_{NN} \end{bmatrix} \times \begin{bmatrix} w_{11} & \cdots & w_{1m} \\ \vdots & \ddots & \vdots \\ w_{N1} & \cdots & w_{Nm} \end{bmatrix} = \begin{bmatrix} w'_{11} & \cdots & w'_{1m} \\ \vdots & \ddots & \vdots \\ w'_{N1} & \cdots & w'_{Nm} \end{bmatrix} \quad (16)$$

The similarity of author *j* and each concept, SU_{jk}, can be calculated by cosine similarity, and it also represents author *j*'s preference for concept *k*. The set of SU_{jk} is SU_j, the concept preference vector of author *j*.

$$SU_j = \{SU_{jk} \mid SU_{jk} = sim(U'_j, C_k), k = 1,2,\ldots, p\}, \quad (17)$$

where $j = 1, 2, \ldots, N$; p is the total number of clusters

U'_j is a row of *U'*, representing the relationship of author *j* and all keywords. C_k, introduced in Section 4.3, is the vector representation of cluster *k*.

If SU_{jk} is larger than a pre-defined threshold, it means that author *j* has an apparent preference for concept *k*, and author *j* should be assigned to

concept *k*. Each author cluster represents a virtual research community with a specific concept. The threshold used in this work is the average value of the similarity of an author and a cluster.

Virtual research communities and co-author relationship can be used to assist users in knowing the distribution of virtual research communities and identify significant researchers of each community. Furthermore, with the graph representation of virtual communities, SNA metrics mentioned in Section 2.2 can be computed[7].

While virtual research communities are constructed, authors in the same virtual research community have similar research interests. Researchers can be recommended according to the relationship of each community and their information need. In this work, two approaches are proposed for recommending papers to authors in the NCTUIR, and the two approaches can be easily extended to any users who are interested in the papers collected in the NCTUIR.

1. Recommendation by personal interests

If an author wants to find out some papers related to his/her research interests, the system can recommend him/her the top n (n=5 in this work) similar papers from the author's virtual communities, except the author's previous works. Cosine similarity of the author model and papers in the author's communities is used to find out the top n papers for recommendation.

2. Recommendation by community

If an author wants to find out some representative papers of a specific virtual research community, the system can recommend him/her the top n (n=5 in this work) relevant papers. The relevant degree of a paper and a community is calculated by the cosine similarity of the paper and the community.

6. Evaluation

This section describes experiment setting and evaluation measurements for the two services presented in Section 4 and Section 5. In addition, some discussions about the proposed approach and legacy methods are also elaborated.

[7] Due to space limitation, the description of SNA metrics for the authors in the NCTUIR is omitted.

6.1. Evaluation for Graphical Representation of Search Results

The corpus containing 541 "Information Retrieval"-related documents is employed to compare our proposed approach with Topic Keyword Clustering. This study applies three evaluations. Firstly, domain experts classify documents into groups in advance, and then the clustering results of the proposed approach and Topic Keyword Clustering are evaluated with these paper classes by entropy, purity, recall and f-measure [29]. The second evaluation compares the compactness and separation of clustering results. In the last evaluation, the similarity values of papers calculated by the proposed approach and Topic Keyword Clustering are compared with the labelling decided by domain experts.

6.1.1. Entropy, Purity, Recall and F-measure

Domain experts classify all papers into 35 groups in advance, and four common evaluation methods – purity, recall, entropy and f-measure, are applied to evaluate the clustering results. The evaluation results are shown in Table 3, and our method is better than Topic Keyword Clustering (less is better in Entropy estimation).

Table 3. Evaluation of clustering results by proposed method and Topic Keyword Clustering.

# of clusters	Topic Keyword Clustering			Our Method		
Estimation	50	75	100	50	75	100
Purity	0.3008	0.4084	0.5259	0.5359	0.6096	0.6295
Recall	0.3239	0.2569	0.2138	0.4709	0.3811	0.3298
Entropy	2.6289	2.0581	1.4362	1.7427	1.348	1.1622
F-measure	0.2472	0.2971	0.3078	0.4339	0.4643	0.4586

6.1.2. Compactness and Separation Degrees

Table 4 shows the compactness (*Cmp*), separation (*Sep*), and overall cluster quality (*Ocq*) of clustering results [30].

$$Ocq(\beta) = \beta \cdot Cmp + (1-\beta) \cdot Sep \tag{18}$$

This study treats compactness and separation with the same weight, and chooses β = 0.5, and smaller *Ocq* value is better. Although the separation of 75 and 100 clusters using Topic Keyword Clustering is better, our method still leads in the overall cluster quality.

Table 4. Compactness, separation degree, and overall cluster quality of clustering results.

	Topic Keyword Clustering			Our Method		
# of clusters	50	75	100	50	75	100
Compactness	0.9217	0.8520	0.7841	0.4914	0.4350	0.3806
Separation	0.6086	0.4820	0.4531	0.5031	0.4991	0.4911
Overall cluster quality	0.7652	0.6670	0.6186	0.4973	0.4671	0.4358

6.1.3. Document Pair Similarity

In this evaluation, two domain experts were asked to label the similarity of 200 random document pairs, and the results is depicted in Table 5. Kappa statistics [31] is applied to evaluate the agreement, and the kappa value is 0.8584 (confidence level: 95%, confidence interval: 0.7893~0.9305), which means high agreement [32]. Consequently, the 186 documents (97+89) agreed by both experts are used for similarity evaluation. Topic Keyword Clustering and our method are evaluated and compared with domain experts' judgement.

Table 6 shows the evaluation results, and according to this table, sensitivity, specificity and accuracy can be computed [33], as illustrated in Table 7. It is easy to find that our method is better than Topic Keyword Clustering.

Table 5. Similarity agreement results of domain experts.

		Expert A		
		No	**Yes**	**Total**
Expert B	**No**	97 (90.65%)	10 (9.35%)	107 (53.5%)
	Yes	4 (4.3%)	89 (95.7%)	93 (46.5%)
Total		101 (51.5%)	99 (49.5%)	200

Table 6. Similarity agreement evaluation.

Similarity Evaluation		Domain Experts	
		Yes	No
Topic Keyword Clustering	Yes	20	5
	No	69	92
Our Method	Yes	57	3
	No	32	94

Table 7. Accuracy of clustering results and domain experts' labeling.

	Sensitivity	Specificity	Accuracy
Topic Keyword Clustering	0.2247	0.9485	0.6022
Our Method	0.6405	0.9691	0.8118

6.1.4. Discussion

According to 6.1.1 – 6.1.3, our method is better than Topic Keyword Clustering. The differences of the two methods are as shown in Table 8, and the major differences are semantic correlation and the document clustering algorithm.

Table 8. Difference between proposed approach and Topic Keyword Clustering.

	Topic Keyword Clustering	**Proposed Approach**
Semantic Correlation	Mutual Information	Log likelihood ratio
Keyword Clustering	*k*-nearest neighbor graph approach	
Document Clustering	Cosine similarity measurement	Bisection *k*-Means

Topic Keyword Clustering uses mutual information (MI) to compute the semantic correlation, and our method uses LLR. The main reason is that MI is proper for judging the independent degree of two terms, but not proper for the relative (dependent) degree of two terms [34]. Table 9 shows an instance of concept subgroups computed by MI and LLR. The concept subgroup computed by MI contains more features, but the concepts of features are less identical, such as "part-of-speech" and "markov model". On the other hand, by LLR, these two feature terms, "part-of-speech" and "markov model", belongs to different concept subgroups, and the concept of each concept subgroup are more consistent.

Table 9. Concept subgroups computed by MI and LLR.

mutual information	**Log likelihood ratio**
part-of-speech tagging, tagger, institute, markov, text segmentation, markov model, estimation, disambiguation, information retrieval ir, workshop, resolution, tagging, rule-based, probability, series	part-of-speech tagger, part-of-speech tagging, text segmentation, story, tagger
	markov, markov model, threshold, relevance feedback

When a document contains multiple concepts, the clustering algorithm used in our method (Bisection k-Means) has better results than Topic Keyword Clustering (cosine similarity). The main reason is that Topic

Keyword Clustering only maps a document to the most similar cluster, but our method maps a document to a similarity vector of every concept subgroup. Thus, if two documents contain common or similar concepts, these two documents can be considered as similar documents. Table 10 is a real example, documents A and B are two search results using "machine learning" as query. In the experiment, our proposed method clusters documents A and B together, but Topic Keyword Clustering assigns them into different groups.

Table 10. Two searching result documents using “machine learning” as keyword.

A	Title: using reinforcement learning to spider the web efficiently
	consider the task of exploring the web in order to find pages of a particular kind or on a particular topic. this task arises in the construction of search engines and web knowledge bases. this paper argues that the creation of efficient web spiders is best framed and solved by reinforcement learning, a branch of machine learning that concerns itself with optimal sequential decision making. one strength of reinforcement learning is that it provides a formalism for measuring the utility of actions that give benefit only in the future. we present an algorithm for learning a value function that maps hyperlinks to future discounted reward by using naive bayes text classifiers. experiments on two real-world spidering tasks show a three-fold improvement in spidering efficiency over traditional breadth-first search, and up to a two-fold improvement over reinforcement learning with immediate reward only. keywords: reinforcement learning, text classification, world wide web, spidering,
B	Title: A machine learning approach to building domain-specific search engines
	domain-specific search engines are becoming increasingly popular because they offer increased accuracy and extra features not possible with general, web-wide search engines. unfortunately, they are also difficult and time-consuming to maintain. this paper proposes the use of machine learning techniques to greatly automate the creation and maintenance of domain-specific search engines. we describe new research in reinforcement learning, text classification and information extraction that enables efficient spidering, populates topic hierarchies, and identifies informative text segments. using these techniques, we have built a demonstration system: a publicly-available search engine for computer science research papers.

6.2. Evaluation for Virtual Community Generation and Paper Recommendation

The corpus used for virtual research community generation contains 235 authors and 226 papers stored in the NCTUIR, and all the authors and papers are computer science major. The evaluation contains two parts. The first one evaluates the result of author clustering, and the other one evaluates the accuracy of paper recommendation.

Table 11. Research concept clusters and relative labels and representative keywords.

Class label	Cluster label
Network Communication	Mobile Computing Routing Protocol PIM-SM Bandwidth Requests TCP Network Management
Artificial Intelligence	Genetic Algorithm Network Motif Brick Motif Content Analysis Neural Network SPDNN Divide-and-conquer Learning
Computer Graphics	Content-based Image Retrieval Watershed Segmentation Toboggan Approach
Information Retrieval	Semantic Query Content Management
Computer System	Memory Cache Parallel Algorithm
Information Security	End-to-end Security
Graph Theory	Interconnection Network
Software Engineering	Reliability Analysis

6.2.1. Evaluation of Author Clustering

The NCTUIR does not provide classifications of research areas in advance, so this study asks two domain experts to partition the formed concept clusters into classes. Domain experts create a total of 8 classes, and Table 11 lists the class labels and the labels of the contained clusters for all the 8 classes.

After concept classes are created, three domain experts then assign all the 235 authors into these classes, and the result is shown in Table 12. If there's an author not similar to any research area class, he/she would be assigned to "others" class.

Table 12. Author classification by domain experts.

Class label	# of authors
Network Communication	111
Artificial Intelligence	28
Information Retrieval	7
Computer System	6
Computer Graphics	23
Information Security	10
Graph Theory	29
Software Engineering	4
Others	17
Total	235

The correctness of author clustering is evaluated by precision and recall [35]. The relative degree of two authors is calculated by Jaccard coefficient, and is adjusted by a parameter, α, whose range is from 0 to 1. If α is 0, it means that the author clustering process does not take social relationship into consideration.

Table 13. Recall and precision of author clustering with different *α* value.

***α* value**	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Precision	0.7071	0.6917	0.6981	0.7107	0.7172	0.7209	0.7209	0.7209	0.7209	0.7209	0.7209
Recall	0.6271	0.7606	0.7785	0.7839	0.7817	0.7828	0.7828	0.7828	0.7828	0.7828	0.7828

Table 13 and Fig. 13 show the recall and precision while adjusting *α* from 0 to 1 in steps of 0.1. According to the results, precision is about 0.7 and

various α values do not significantly affect precision. Recall is about 0.78 while α is larger than 0.3. These show our approach is able to cluster authors to proper research communities. However, when α is small (<0.3), the recall is apparently worse, so 0.3 is chosen as the parameter of Jaccard coefficient for the following evaluation.

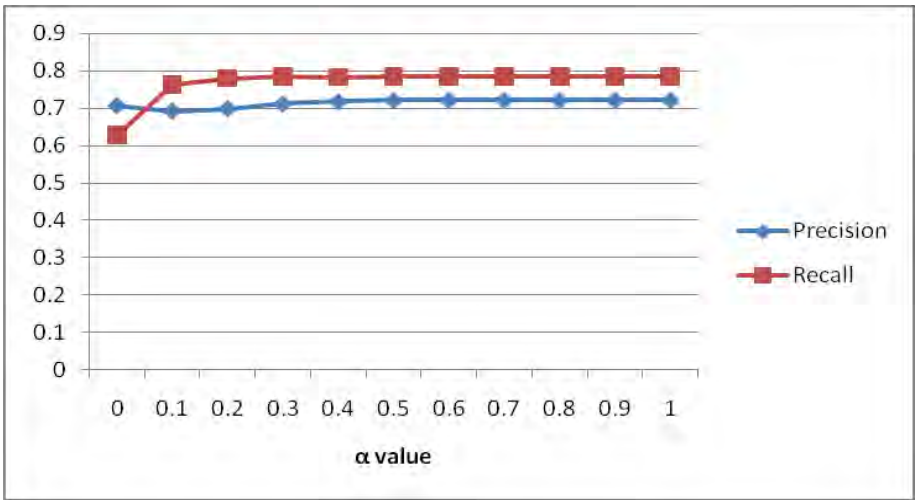


Fig. 13. Graph of recall and precision of author clustering with different α value.

6.2.2. Accuracy of Paper Recommendation

This evaluation uses different scenarios for paper recommendation. Each scenario represents a single user with identical information need. The recommended papers are evaluated by two domain experts, and Kappa Statistics is used for calculating the accuracy of paper recommendation. There are 219 papers recommended to two domain experts for evaluation, and the relevance judgement results of two domain experts is as Table 14.

Table 14. Relevance judgment results of domain experts.

		Expert A				Total	
		No		Yes			
Expert B	No	21	(9.6%)	9	(4.1%)	30	(13.7%)
	Yes	2	(0.9%)	187	(85.4%)	189	(86.3%)
Total		23	(10.5%)	196	(89.5%)	219	

The kappa value is 0.764, and the strength of agreement is substantial. Besides, for the 208 papers with the same relevance judgements of the two domain experts, the accuracy, 187/208 = 89.9%, is apparently good.

7. Conclusion and Future Work

This study shows the application of concept extraction and clustering in IR systems. The contribution is two fold. Firstly, the modified concept extraction and clustering method outperforms the original Topic Keyword Clustering. Secondly, from the application aspect, the proposed method facilitates the organization and visualization of search results, and the provision of virtual research communities and paper recommendation for researchers.

Concept extraction and clustering is the core mechanism of this study. A concept is defined as a group of terms whose co-occurrence statistics reveal their similarity in semantics. The proposed approach modifies the original Topic Keyword Clustering. It treats terms and the relationships of term pairs as vertices and edges of a graph. Removing insignificant vertices and edges, and merging similar concept sub-groups generate the concept groups. The proposed method has better performance than Topic Keyword Clustering, and the evaluation results provide evidences. The main reason is, instead of MI used in Topic Keyword Clustering, LLR is applied to calculate semantic correlation. Furthermore, Topic Keyword Clustering assigns a document to a single concept group; on the other hand, the proposed method may assign a document to more than one concept groups with high similarity.

To well organize search results, concepts should be extracted from the corpus in advance. Then, documents are clustered according to the similarities of documents and concepts. Citation relations and labelling are also applied for assisting users to realize the concept of each cluster and the intra- and inter-relations among clusters.

For providing virtual research communities and paper recommendation, the author model is created to maintain author-term relationships. Second, the social information, i.e. co-author relationship, is applied to adjust the author model. Then, author clustering is done by assigning authors to concept groups according to the similarity of the updated author model and concept groups. Finally, each author cluster is a virtual research community, and each virtual community can provide users with information on representative researchers and papers. Besides, according to user's research interests, proper papers in a virtual community can be recommended.

The proposed concept extraction and clustering method generates concepts in a flat level and do not consider the hierarchical structure inside a concept group. With a hierarchical structure, users are able to know more details of a research concept when they track down deeply. For this purpose, the future study will track the process of concept group generation and the relationships of subgroups, and use ontology to model the hierarchical structure of groups. Besides, although at least two domain experts are asked

to do qualitative evaluation, we will apply more experts for more rigorous evaluation in the future.

References

1. Manning, C. D., Raghavan, P. and Schütze, H. (2008), *Introduction to Information Retrieval*, Cambridge University Press, pp. 349-375.
2. Garfield, E. (1979). Citation indexing – Its theory and application in science, technology, and humanities. New York, NY: John Wiley & Sons.
3. Gan, G., Ma, C. and Wu, J. (2007), "Data Clustering: Theory, Algorithms, and Applications", SIAM, Society for Industrial and Applied Mathematics.
4. MacQueen, J. (1967), "Some methods for classification and analysis of multivariate observations", in *Proceeding of Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281-297.
5. Killani, R., Rao, K. S., Satapathy, S. C., Pradhan, G. and Chandran, K. R. (2010), "Effective Document Clustering with Particle Swarm Optimization", *Swarm, Evolutionary, and Memetic Computing, Lecture Notes in Computer Science*, Vol. 6466/2010, pp. 623-629
6. Taghva, J. and Veni, R. (2010), "Effects of Similarity Metrics on Document Clustering", in *Seventh International Conference on Information Technology*, pp. 222-226.
7. Oikonomakou, N. and Vazirgiannis, M. (2010), "A Review of Web Document Clustering Approaches", *Data Mining and Knowledge Discovery Handbook*, Part 6, pp. 931-948
8. Mahdavi, M. and Abolhassani, H. (2009), "Harmony K-means algorithm for document clustering", *Data Mining and Knowledge Discovery*, Vol. 18, No. 3, pp. 370-391
9. Zhang, X., Jing, L., Hu, X., Ng, M. and Zhou, X. (2007), "A Comparative Study of Ontology Based Term Similarity Measures on PubMed Document Clustering", *Advances in Databases: Concepts, Systems and Applications, Lecture Notes in Computer Science*, Vol. 4443/2007, pp. 115-126.
10. Kogan, J. (2006), Introduction to Clustering Large and High-Dimensional Data, Cambridge University Press.
11. Iliopoulos, I., Enright, A. J. and Ouzounis, C. A. (2001), "TEXTQUEST: Document Clustering of MEDLINE Abstracts For Concept Discovery In Molecular Biology", in *Proceeding of Pacific Symposium on Biocomputing 2001*, pp. 384-395
12. Slonim, N., Tishby, N. (2000), "Document clustering using word clusters via the information bottleneck method", in Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval in Athens, Greece, ACM, New York, NY, USA, pp. 208-215.
13. Athanasios P. (1991), Probability, Random Variables and Stochastic Processes, McGraw-Hill, New York, NY.
14. Chang, H. C. and Hsu, C. C. (2005), "Using topic keyword clusters for automatic document clustering", *IEICE Transaction on Information System*, Vol. E88-D No. 8, pp. 1852-1860.
15. Moreno, J. L. (1934), *Who Shall Survive?*, Mental Health Resources.

16. Liu, X., Bollen, J., Nelson, M. L. and Van de Sompel, H. (2005), "Co-authorship networks in the digital library research community", *Information Processing and Management*, Vol. 41 No.6, pp. 1462-1480.
17. Wasserman, S. and Faust, K. (1994). “Social network analysis: Methods and applications”, Cambridge: Cambridge University Press.
18. Tyler, J. R., Wilkinson, D. M. and Huberman, B. A. (2003), "Email as spectroscopy: Automated discovery of community structure within organizations", in Huysman, M. H., Wenger, E. and Wulf, V. (Ed.), *Communities and technologies*, Springer, pp. 81-96.
19. Mika, P. (2005), "Flink: Semantic Web technology for the extraction and analysis of social networks", *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 3 No. 2-3, pp. 211-223.
20. Page, L. and Brin, S. (1998), "The anatomy of a large-scale hypertextual Web search engine", *Computer Networks and ISDN Systems*, Vol. 30 No. 1-7, pp. 107-117.
21. Matsuo, Y., Mori, J. and Hamasaki, M. (2007), "POLYPHONET: An advanced social network extraction system from the Web", *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 5 No. 4, pp. 262-278.
22. Lehmann, L. E. (1986), *Testing Statistical Hypotheses*, Wiley.
23. Neyman, J. and Pearson, E. (1967), *Joint statistical papers*, Hodder Arnold.
24. Gowda, K. C. and Krishna, G. (1978), "Agglomerative clustering using the concept of mutual nearest neighbourhood", *Pattern Recognition*, Vol. 10 No. 2, pp. 105-112.
25. Steinbach, M., Karypis, G. and Kumar, V. (2000), "A comparison of document clustering techniques", paper presented at the KDD Workshop on Text Mining, August 20-23, Boston, MA, USA, available at: http://www.cs.cmu.edu/~dunja/KDDpapers/Steinbach_IR.pdf (accessed 12 May 2009).
26. De Bellis, N. (2009), Bibliometrics and citation analysis: from the science citation index to cybermetrics, The Scarecrow Press, pp. 156-166.
27. Chan, S. E., Pon, R. K. and Cárdenas, A. F. (2006), "Visualization and Clustering of Author Social Networks", in International Conference on Distributed Multimedia Systems Workshop on Visual Languages and Computing in Grand Canyon, Arizona, USA, 2007, pp. 30-31.
28. Shah, M. A. (1997), ReferralWeb: A resource location system guided by personal relations, Master's thesis, M.I.T.
29. Rosell, M., Kann, V. and Litton, J. (2004), "Comparing comparisons: Document clustering evaluation using two manual classifications", in *Proceeding of International Conference on Natural Language Processing*, Allied Publishers Pvt. Ltd., pp. 207-216.
30. He, J., Tan, A., Tan, C. L. and Sung, S. Y. (2003), "On quantitative evaluation of clustering systems", in Wu, W., Xiong, H. and Shekhar S. (Ed.), *Clustering and Information Retrieval*, Springer, pp. 105-134.
31. Cohen, J. (1960), "A coefficient of agreement for nominal scales", *Educational and Psychological Measurement*, Vol.20 No.1, pp. 37-46.
32. Landis, J. R. and Koch, G. G. (1977), "The measurement of observer agreement for categorical data", *Biometrics*, Vol. 33, pp. 159-174.
33. Han, J. and Kamber, M. (2000), *Data Mining: Concepts and Techniques*, Morgan Kaufmann, San Fransisco, CA, USA.
34. Manning, C. D. and Schutze, H. (1999), *Foundations of Statistical Natural Language Processing*, MIT Press, Cambridge, MA, USA.

35. Makhoul, J., Kubala, F., Schwartz, R. and Weischedel, R. (1999), "Performance measures for information extraction", in *Broadcast News Workshop '99 Proceedings*, Morgan Kaufmann, San Fransisco, CA, USA, pp. 249-252.

Shihn-Yuarn Chen received his B.S. and M.S. degree in Computer Science and Information Engineering from National Chiao Tung University, Taiwan, R.O.C. Now he is a Ph.D. candidate in Computer Science, National Chiao Tung University. His research interests include digital library, information retrieval, social tagging and Web 2.0.

Chia-Ning Chang received her B.S. and M.S. degree in Computer and Information Science, from National Chiao Tung University, Taiwan, R.O.C. She is currently employed as a RD-DE-PCM engineer in Cyberlink Corp. in Taiwan.

Yi-Hsiang Nien received his M.S. degree in Institute of Information Management, from National Chiao Tung University, Taiwan, R.O.C. He is currently working as an engineer in Taiwan Semiconductor Manufacturing Company Ltd. in Taiwan

Hao-Ren Ke received his B.S. degree in 1989 and Ph.D. degree in 1993, both in Computer and Information Science, from National Chiao Tung University, Taiwan, R.O.C. Now he is a professor of the Graduate Institute of Library and Information Studies and the deputy library director in National Taiwan Normal University. His main research interests are in digital library/archives, library and information system, information retrieval, and Web mining.

Received: November 24, 2010; Accepted: May 17, 2011

DOI 10.2298/CSIS110206044M

A Method for Decision Making with the OWA Operator

José M. Merigó[1] and Anna M. Gil-Lafuente[1]

[1] Department of Business Administration, University of Barcelona
Av. Diagonal 690, 08034 Barcelona, Spain
{jmerigo, amgil}@ub.edu

Abstract. A new method for decision making that uses the ordered weighted averaging (OWA) operator in the aggregation of the information is presented. It is used a concept that it is known in the literature as the index of maximum and minimum level (IMAM). This index is based on distance measures and other techniques that are useful for decision making. By using the OWA operator in the IMAM, we form a new aggregation operator that we call the ordered weighted averaging index of maximum and minimum level (OWAIMAM) operator. The main advantage is that it provides a parameterized family of aggregation operators between the minimum and the maximum and a wide range of special cases. Then, the decision maker may take decisions according to his degree of optimism and considering ideals in the decision process. A further extension of this approach is presented by using hybrid averages and Choquet integrals. We also develop an application of the new approach in a multi-person decision-making problem regarding the selection of strategies.

Keywords: decision making, OWA operator, aggregation operator, index of maximum and minimum level, selection of strategies.

1. Introduction

The index of maximum and minimum (IMAM) level [1] is a very useful technique that provides similar results with the Hamming distance with some differences that makes it more complete. It includes the Hamming distance and the adequacy coefficient [2-6] in the same formulation. Since its appearance, it has been used in a wide range of applications such as fuzzy set theory, business decisions and multicriteria decision making [7-8]. Often, we prefer to use the normalized IMAM (NIMAM) because we want an average result of all the individual comparisons. This type of index is also known as the weighted IMAM (WIMAM) when we prefer to give different degrees of importance to the individual comparisons instead of giving them the same importance.

Sometimes, when calculating the NIMAM, it would be interesting to consider the attitudinal character of the decision maker. A very useful tool for aggregating the information considering the attitudinal character of the decision maker is the ordered weighted averaging (OWA) operator [9]. The OWA operator is an aggregation operator that includes the maximum, the minimum and the average criteria, as special cases. It has been used in a wide range of applications [10-21].

The aim of this paper is to present a new type of IMAM operator that uses the OWA operator in the aggregation process. We call this new aggregation operator, the ordered weighted averaging index of maximum and minimum level (OWAIMAM) operator. The fundamental characteristic of this index is that it normalizes the IMAM with the OWA operator. Therefore, it is possible to develop a more general IMAM that includes the maximum, the minimum and the NIMAM, as special cases. The main advantage of the OWAIMAM is the possibility of over or under estimate the results of an aggregation in order to take a decision according to a certain degree of optimism. Then, in a decision making problem, the decision maker will be able to take decisions according to his degree of optimism. Some of its main properties and different families of OWAIMAM operators are studied.

A further extension of this approach is presented by using the hybrid average [22-27]. The main advantage of this approach is that it uses the weighted average and the OWA operator in the same formulation. Thus, it is possible to consider the subjective probability and the attitudinal character of the decision maker. We call it the hybrid averaging IMAM (HAIMAM) operator. Moreover, we generalize this approach by using Choquet integrals [28-32] obtaining the Choquet integral IMAM aggregation (CIIMAMA). Thus, a more robust and general formulation of the IMAM operator is obtained.

We also develop an application of this new method in a business multi-person decision-making problem. This decision-making model can be summarized in one aggregation operator called the multi-person OWAIMAM (MP-OWAIMAM) operator. We apply it in the selection of strategies because this problem can be considered as a general one that includes a wide range of business situations. Note that other applications could be developed such as in human resource management, supplier selection and product management. For further information on other decision-making methods, refer, e.g., to [33-42].

This paper is organized as follows. In Section 2 some basic concepts such as the OWA operator and the IMAM are described. Section 3 presents the OWAIMAM operator and Section 4 analyzes some of its families. Section 5 presents an extension by using the hybrid average and Section 6 a generalization by using Choquet integrals. In Section 7 a multi-person decision-making model is presented and in Section 8 an application of the new approach in the selection of strategies. Finally, Section 9 summarizes the main findings of the paper.

2. Preliminaries

In this Section, we briefly review some basic concepts to be used throughout the paper such as the IMAM and the OWA operator.

2.1. The Index of Maximum and Minimum Level

The NIMAM [1] is an index used for calculating the differences between two elements, two sets, etc. In decision making, it is very useful for comparing alternatives in different business decision making problems such as financial management, human resource management, product management, etc. In fuzzy set theory, it can be useful, for example, for the calculation of distances between fuzzy sets, interval-valued fuzzy sets, intuitionistic fuzzy sets, etc. It is a very useful technique that provides similar results than the Hamming distance but with some differences that makes it more complete. Basically, it can be defined as a measure that includes the Hamming distance and the adequacy coefficient [2-6] in the same formulation. For two sets P and P_j, it can be defined as follows.

Definition 1. A NIMAM of dimension n is a mapping K: $[0, 1]^n \times [0, 1]^n \rightarrow [0, 1]$ such that:

$$K(P, P_j) = \frac{1}{u+v}\left[\sum_u \left|\mu_i(u) - \mu_i^{(j)}(u)\right| + \sum_v \left(0 \vee (\mu_i(v) - \mu_i^{(j)}(v))\right)\right], \quad (1)$$

where μ_i and $\mu_i^{(j)}$ are the ith arguments of the sets P and P_j respectively, u and v are the number of elements used with the Hamming distance and with the dual adequacy coefficient, respectively, and $u + v = n$.

Sometimes, when normalizing the IMAM it is better to give different weights to each individual element. Then, the index is known as the WIMAM. It can be defined as follows.

Definition 2. A WIMAM of dimension n is a mapping K: $[0, 1]^n \times [0, 1]^n \rightarrow [0, 1]$ that has an associated weighting vector W of dimension n with the following properties:

$\sum_{i=1}^{n} Z_i = 1$ with $\sum_u Z_i(u) + \sum_v Z_i(v) = 1$

$Z_i \in [0, 1]$

and such that:

$$K(P, P_j) = \quad (2)$$

$$\sum_{u} Z_i(u) \times \left| \mu_i(u) - \mu_i^{(j)}(u) \right| + \sum_{v} Z_i(v) \times \left[0 \vee (\mu_i(v) - \mu_i^{(j)}(v)) \right],$$

where μ_i and $\mu_i^{(j)}$ are the ith arguments of the sets P and P_j respectively, u and v are the number of elements used with the Hamming distance and with the dual adequacy coefficient, respectively, and $u + v = n$.

Note that if $u = n$, the WIMAM operator becomes the usual weighted Hamming distance (WHD) that can be defined as follows.

Definition 3. A weighted Hamming distance of dimension n is a mapping WHD: $[0, 1]^n \times [0, 1]^n \rightarrow [0, 1]$ that has an associated weighting vector W of dimension n with $W = \sum_{j=1}^{n} w_j = 1$ and $w_j \in [0, 1]$, such that:

$$WHD\ (A, B) = \left(\sum_{i=1}^{n} w_i \mid a_i - b_i \mid \right), \tag{3}$$

where a_i and b_i are the ith arguments of the sets A and B respectively.

Moreover, the WIMAM operator accomplishes similar properties than the distance measures [29] although it does not always accomplish commutativity, from the perspective of a distance measure, because it uses norms in the aggregation process. In this case we have that a WIMAM aggregation fulfils:

Non-negativity: $K\ (A_1, A_2) \geq 0$.
Reflexivity: $K\ (A_1, A_1) = 0$.
Triangle inequality: $K\ (A_1, A_2) + K\ (A_2, A_3) \geq K\ (A_1, A_3)$.

2.2. The OWA Operator

The OWA operator [9] provides a parameterized family of aggregation operators which have been used in many applications. It can be defined as follows.

Definition 4. An OWA operator of dimension n is a mapping OWA: $R^n \rightarrow R$ that has an associated weighting vector W of dimension n having the properties:

$w_j \in [0, 1]$
$\sum_{j=1}^{n} w_j = 1$

and such that:

$$OWA\ (a_1, a_2, \ldots, a_n) = \sum_{j=1}^{n} w_j b_j\ , \tag{4}$$

where b_j is the jth largest of the a_i.

From a generalized perspective of the reordering step it is possible to distinguish between the descending OWA (DOWA) operator and the ascending OWA (AOWA) operator. Note that the weights of these two operators are related by $w_j = w^*_{n-j+1}$, where w_j is the jth weight of the DOWA and w^*_{n-j+1} the jth weight of the AOWA operator. For further properties and applications on the OWA operator, refer, e.g., to [10,21,43-45].

3. The OWAIMAM Operator

In this Section, the use of the OWA operator in the IMAM operator is introduced. We call it the ordered weighted averaging index of maximum and minimum level (OWAIMAM). It can be defined as follows.

Definition 5. An OWAIMAM operator of dimension n, is a mapping *OWAIMAM*: $[0, 1]^n \times [0, 1]^n \rightarrow [0, 1]$ that has an associated weighting vector W, with $w_j \in [0, 1]$ and $\sum_{j=1}^{n} w_j = 1$, such that:

$$OWAIMAM\ (\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_n, y_n\rangle) = \sum_{j=1}^{n} w_j K_j\ , \tag{5}$$

where K_j represents the jth largest of all the $|x_i - y_i|$ and the $[0 \vee (x_i - y_i)]$.

In the following, a simple numerical example concerning the aggregation process with the OWAIMAM operator is presented.

Example 1. Assume the following arguments in an aggregation process: X = (0.3, 0.4, 0.8, 0.6), Y = (0.5, 0.7, 0.3, 0.7). Assume the following weighting vector W = (0.1, 0.2, 0.3, 0.4). If we calculate the similarity between X and Y using the OWAIMAM operator, we get the following. Assume that the first two arguments have to be treated with the Hamming distance and the other two with the dual adequacy coefficient.

$OWAIMAM\ (X, Y) = 0.1 \times [0 \vee (0.8 - 0.3)] + 0.2 \times |0.4 - 0.7| + 0.3 \times |0.3 - 0.5| + 0.4 \times [0 \vee (0.6 - 0.7)] = 0.17$.

Note that from a generalized perspective of the reordering step it is possible to distinguish between descending and ascending orders. The weights of these operators are related by $w_j = w^*_{n-j+1}$, where w_j is the jth weight of the descending OWAIMAM (DOWAIMAM) and w^*_{n-j+1} the jth weight of the ascending OWAIMAM (AOWAIMAM) operator.

If K is a vector corresponding to the ordered arguments K_j, we shall call this the ordered argument vector, and W^T is the transpose of the weighting vector, then the OWAIMAM can be expressed as:

$$OWAIMAM\ (\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_n, y_n\rangle) = W^T K\ . \tag{6}$$

Note that if the weighting vector is not normalized, i.e., $W = \sum_{j=1}^{n} w_j \neq 1$, then, the OWAIMAM operator can be expressed as:

$$OWAIMAM\ (\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_n, y_n\rangle) = \frac{1}{W} \sum_{j=1}^{n} w_j K_j\ . \tag{7}$$

Analogously to the OWAIMAM operator, we can suggest a removal index that it is a dual of the OWAIMAM because $Q\ (\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_n, y_n\rangle) = 1 - K\ (\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_n, y_n\rangle)$. We refer to it as the ordered weighted averaging dual index of maximum and minimum level (OWADIMAM). Note that it can be seen as a dissimilarity measure. It is defined as follows.

Definition 6. An OWADIMAM operator of dimension n, is a mapping $OWADIMAM$: $[0, 1]^n \times [0, 1]^n \rightarrow [0, 1]$ that has an associated weighting vector W, with $w_j \in [0, 1]$ and the sum of the weights is equal to 1, then:

$$OWADIMAM\ (\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_n, y_n\rangle) = \sum_{j=1}^{n} w_j Q_j\ , \tag{8}$$

where Q_j represents the jth largest of all the $[1 - |x_i - y_i|]$ and the $[1 \wedge (1 - x_i + y_i)]$; with $k = 1,2,\ldots,m$.

The final result will be a number between [0, 1]. Note that in this case the recommendation is to select the lowest value as the best result.

In this case, we can also distinguish between the descending OWADIMAM (DOWADIMAM) and the ascending OWADIMAM (AOWADIMAM) operator.

Note also that the OWAIMAM operator follows the usual methodology of the aggregation operators. Thus, it is commutative, monotonic, bounded, idempotent, nonnegative and reflexive. As we can see, it accomplishes the usual properties excepting commutativity from the perspective of a distance measure because of the use of norms in the aggregation. These properties can be proved with the following theorems.

Theorem 1 (Monotonicity). Assume f is the OWAIMAM operator, if $|x_i - y_i| \geq |u_i - v_i|$, for all i, then:

$$f\ (\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_n, y_n\rangle) \geq f\ (\langle u_1, v_1\rangle, \langle u_2, v_2\rangle, \ldots, \langle u_n, v_n\rangle). \tag{9}$$

Proof. Let

$$f(\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_n, y_n\rangle) = \sum_{j=1}^{n} w_j K_j \,, \tag{10}$$

$$f(\langle u_1, v_1\rangle, \langle u_2, v_2\rangle, \ldots, \langle u_n, v_n\rangle) = \sum_{j=1}^{n} w_j Q_j \,. \tag{11}$$

Since $|x_i - y_i| \geq |u_i - v_i|$, for all i, it follows that, $K_j \geq Q_j$, and then

$$f(\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_n, y_n\rangle) \geq f(\langle u_1, v_1\rangle, \langle u_2, v_2\rangle, \ldots, \langle u_n, v_n\rangle). \quad \blacksquare$$

Theorem 2 (Commutativity). Assume f is the OWAIMAM operator, then:

$$f(\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_n, y_n\rangle) = f(\langle u_1, v_1\rangle, \langle u_2, v_2\rangle, \ldots, \langle u_n, v_n\rangle). \tag{12}$$

where $(\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_n, y_n\rangle)$ is any permutation of the arguments $(\langle u_1, v_1\rangle, \langle u_2, v_2\rangle, \ldots, \langle u_n, v_n\rangle)$.

Proof. Let

$$f(\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_n, y_n\rangle) = \sum_{j=1}^{n} w_j K_j \,, \tag{13}$$

$$f(\langle u_1, v_1\rangle, \langle u_2, v_2\rangle, \ldots, \langle u_n, v_n\rangle) = \sum_{j=1}^{n} w_j Q_j \,. \tag{14}$$

Since $(\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_n, y_n\rangle)$ is a permutation of $(\langle u_1, v_1\rangle, \langle u_2, v_2\rangle, \ldots, \langle u_n, v_n\rangle)$, we have $K_j = Q_j$, for all j, and then

$$f(\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_n, y_n\rangle) = f(\langle u_1, v_1\rangle, \langle u_2, v_2\rangle, \ldots, \langle u_n, v_n\rangle). \quad \blacksquare$$

Theorem 3 (Idempotency). Assume f is the OWAIMAM operator, if $|x_i - y_i| = |x - y|$, for all i, then:

$$f(\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_n, y_n\rangle) = |x - y|. \tag{15}$$

Proof. Since $|x_i - y_i| = |x - y|$, for all i,

$$f(\langle x_1, y_1\rangle, \ldots, \langle x_n, y_n\rangle) = \sum_{j=1}^{n} w_j K_j = \sum_{j=1}^{n} w_j \,|\, x - y \,| \,= |\, x - y \,| \sum_{j=1}^{n} w_j \,. \tag{16}$$

Since $\sum_{j=1}^{n} w_j = 1$,

$$f(\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_n, y_n\rangle) = |x - y|. \quad \blacksquare$$

Theorem 4 (Bounded). Assume f is the OWAIMAM operator, then:

$$\text{Min}\{|x_i - y_i|\} \leq f(\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_n, y_n\rangle) \leq \text{Max}\{|x_i - y_i|\}. \quad (17)$$

Proof. Let $\max\{|x_i - y_i|\} = b$, and $\min\{|x_i - y_i|\} = a$, then

$$f(\langle x_1, y_1\rangle, \ldots, \langle x_n, y_n\rangle) = \sum_{j=1}^{n} w_j K_j \leq \sum_{j=1}^{n} w_j b = b \sum_{j=1}^{n} w_j , \quad (18)$$

$$f(\langle x_1, y_1\rangle, \ldots, \langle x_n, y_n\rangle) = \sum_{j=1}^{n} w_j K_j \geq \sum_{j=1}^{n} w_j a = a \sum_{j=1}^{n} w_j . \quad (19)$$

Since $\sum_{j=1}^{n} w_j = 1$,

$$f(\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_n, y_n\rangle) \leq b. \quad (20)$$

$$f(\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_n, y_n\rangle) \geq a. \quad (21)$$

Therefore,

$$\text{Min}\{|x_i - y_i|\} \leq f(\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_n, y_n\rangle) \leq \text{Max}\{|x_i - y_i|\}. \quad \blacksquare$$

Theorem 5 (Nonnegativity). Assume f is the IOWAD operator, then:

$$f(\langle x_1, y_1\rangle, \ldots, \langle x_n, y_n\rangle) \geq 0. \quad (22)$$

Proof. Let

$$f(\langle x_1, y_1\rangle, \ldots, \langle x_n, y_n\rangle) = \sum_{j=1}^{n} w_j b_j , \quad (23)$$

Since $|x_i - y_i| \geq 0$, for all i, we obtain

$$f(\langle x_1, y_1\rangle, \ldots, \langle x_n, y_n\rangle) \geq 0. \quad \blacksquare$$

Theorem 6 (Reflexivity). Assume f is the IOWAD operator, then:

$$f(\langle x_1, x_1\rangle, \ldots, \langle x_n, x_n\rangle) = 0. \quad (24)$$

Proof. Let

$$f(\langle x_1, x_1\rangle, \ldots, \langle x_n, x_n\rangle) = \sum_{j=1}^{n} w_j b_j , \quad (25)$$

Since $x_i = x_i$, $|x_i - x_i| = 0$, for all i, therefore,

$$f(\langle x_1, x_1\rangle, \ldots, \langle x_n, x_n\rangle) = 0.$$ ∎

A further interesting feature to consider in the OWAIMAM operator is the unification point with distance measures. The unification point between the IMAM and the Hamming distance appears when $x_i \geq y_i$ for all i. In the OWAIMAM operator, we find a similar situation with the difference that now the unification is with the ordered weighted averaging distance (OWAD) operator [6]. Then, we get the following.

Theorem 7. Assume *OWAD* $(\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_n, y_n\rangle)$ is the OWAD operator and *OWADIMAM* $(\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_n, y_n\rangle)$ the OWADIMAM operator. If $x_i \geq y_i$ for all i, then:

$$OWAD(\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_n, y_n\rangle) = OWADIMAM(\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_n, y_n\rangle). \quad (26)$$

Proof. Let

$$OWAD(\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_n, y_n\rangle) = \sum_{j=1}^{n} w_j b_j, \quad (27)$$

$$OWADIMAM(\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_n, y_n\rangle) = \sum_{j=1}^{n} w_j Q_j. \quad (28)$$

Since $x_i \geq y_i$ for all i, $[0 \vee (x_i - y_i)] = (x_i - y_i)$ for all i, then

$$OWADIMAM(\langle x_1, y_1\rangle, \ldots, \langle x_n, y_n\rangle) = \sum_{j=1}^{n} w_j (x_i - y_i) = OWAD(\langle x_1, y_1\rangle, \ldots, \langle x_n, y_n\rangle).$$ ∎

As we can see, the unification with the Hamming distance is found with the dual IMAM. Note that it is possible to distinguish between different types of unifications depending on the problem analyzed such as partial and total unification point. The partial unification point appears if at least one of the alternatives but not all of them is in a situation of unification point and the total unification point appears if all the alternatives accomplish the conditions of the unification point. Note that it is straightforward to prove these unifications by looking to [5,46] and following Theorem 7.

Another interesting issue to analyze is the different measures used to characterize the weighting vector of the OWAIMAM operator. Based on the measures developed for the OWA operator in [9,19], they can be defined as follows. For the attitudinal character, we get the following:

$$\alpha(W) = \sum_{j=1}^{n} w_j \left(\frac{n-j}{n-1} \right). \quad (29)$$

It can be shown that $\alpha \in [0, 1]$. Note that for the optimistic criteria $\alpha(W) = 1$, for the pessimistic criteria $\alpha(W) = 0$, and for the average criteria $\alpha(W) = 0.5$.

The dispersion is a measure that provides the type of information being used. It can be defined as follows.

$$H(W) = -\sum_{j=1}^{n} w_j \ln(w_j). \quad (30)$$

For example, if $w_j = 1$ for some j, then $H(W) = 0$, and the least amount of information is used. If $w_j = 1/n$ for all j, then, the amount of information used is maximum. The divergence can be defined as follows.

$$Div(W) = \sum_{j=1}^{n} w_j \left(\frac{n-j}{n-1} - \alpha(W) \right)^2. \quad (31)$$

Note that the divergence can also be formulated with an ascending order in a similar way as it has been shown in the attitudinal character.

4. Families of OWAIMAM Operators

The OWAIMAM operator provides a parameterized family of aggregation operators. Therefore, it includes a wide range of special cases. In Table 1, some of these families of OWAIMAM operators are presented.

For more information on these and other families of OWAIMAM operators that are based on the OWA methodology, refer, e.g., to [9,15,17,47-49].

In the following, the main features of the families presented in Table 1 are commented.

Table 1. Families of OWAIMAM operators.

Basic families	*Weighting vector W*
• The OWA operator • The Hamming distance • The OWAD operator • The adequacy coefficient • The OWAAC operator • Etc.	• Maximum and Minimum • NIMAM and WIMAM • Olympic-OWAIMAM • Window-OWAIMAM • S-OWAIMAM • Centered-OWAIMAM • BUM function – OWAIMAM • Nonmonotonic-OWAIMAM • Etc.

Remark 1. If the second set is empty, the OWAIMAM operator becomes the OWA operator. If all the individual similarities use the Hamming distance, we get the OWAD operator [6] and if all of them use the adequacy coefficient, we obtain the ordered weighted averaging adequacy coefficient (OWAAC) operator [5].

Remark 2. The maximum is obtained if $w_1 = 1$ and $w_j = 0$, for all $j \neq 1$ and the minimum if $w_n = 1$ and $w_j = 0$, for all $j \neq n$. More generally, if $w_k = 1$ and $w_j = 0$, for all $j \neq k$, we get, *OWAIMAM* $(\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_n, y_n\rangle) = b_k$, where b_k is the kth largest argument a_i. The NIMAM is formed when $w_j = 1/n$, and $\omega_i = 1/n$, for all a_i. The WIMAM is obtained when $w_j = 1/n$, for all a_i. Note that the construction of the WIMAM from the OWAIMAM is artificial in the sense that it considers the importance of the attributes while the OWAIMAM focuses on the degree of optimism of the aggregation.

Remark 3. The olympic-OWAIMAM is found when $w_1 = w_n = 0$, and for all others $w_{j^*} = 1/(n - 2)$. Note that it is possible to present a general form of the olympic-OWAIMAM considering that $w_j = 0$ for $j = 1, 2, \ldots, k, n, n - 1, \ldots, n - k + 1$; and for all others $w_{j^*} = 1/(n - 2k)$, where $k < n/2$. Note that if $k = 1$, then, this general form becomes the usual olympic-OWAIMAM. If $k = (n - 1)/2$, then, it becomes the median-OWAIMAM aggregation. That is, if n is odd we assign $w_{(n+1)/2} = 1$ and $w_{j^*} = 0$ for all others. If n is even we assign for example, $w_{n/2} = w_{(n/2)+1} = 0.5$ and $w_{j^*} = 0$ for all others.

Remark 4. Additionally, it is also possible to form the contrary case of the general olympic-OWAIMAM operator. This case is obtained when $w_j = (1/2k)$ for $j = 1, 2, \ldots, k, n, n - 1, \ldots, n - k + 1$; and $w_j = 0$, for all others, where $k < n/2$. Note that if $k = 1$, then, the contrary case of the median-OWAIMAM is obtained.

Remark 5. Following the ideas of Yager [49], the window-OWAIMAM operator can be formed. It is obtained when $w_{j^*} = 1/m$ for $k \leq j^* \leq k + m - 1$ and $w_{j^*} = 0$ for $j^* > k + m$ and $j^* < k$. Note that k and m must be positive integers such that $k + m - 1 \leq n$. Also note that if $m = k = 1$, the window-OWAIMAM becomes the maximum and if $m = 1$, $k = n$, it becomes the minimum. And if $m = n$ and $k = 1$, it is obtained the NIMAM.

Remark 6. A further interesting family is the S-OWAIMAM operator [49]. It can be classified in three classes: the "orlike", the "andlike" and the generalized S-OWAIMAM operator. The generalized S-OWAIMAM operator is obtained when $w_1 = (1/n)(1 - (\alpha + \beta)) + \alpha$, $w_n = (1/n)(1 - (\alpha + \beta)) + \beta$, and $w_j = (1/n)(1 - (\alpha + \beta))$ for $j = 2$ to $n - 1$ where $\alpha, \beta \in [0, 1]$ and $\alpha + \beta \leq 1$. Note that if $\alpha = 0$, the generalized S-OWAIMAM operator becomes the "andlike" S-OWAIMAM operator and if $\beta = 0$, it becomes the "orlike" S-OWAIMAM operator. Also note that if $\alpha + \beta = 1$, we get the Hurwicz IMAM criteria.

Remark 7. Another family of aggregation operator that could be used is the centered-OWAIMAM operator. An OWAIMAM operator is defined as a centered aggregation operator if it is symmetric, inclusive and strongly decaying. It is symmetric if $w_j = w_{j+n-1}$. It is inclusive if $w_j > 0$. It is strongly decaying when $i < j \leq (n + 1)/2$ then $w_i < w_j$ and when $i > j \geq (n + 1)/2$ then $w_i < w_j$. Note that it is possible to consider a softening of the third condition by using $w_i \leq w_j$ instead of $w_i < w_j$ and it is possible to remove the second condition.

Remark 8. Another interesting method for determining the OWAIMAM weights is the functional method. It can be described as follows. Let f be a function f: [0, 1] $\rightarrow$ [0, 1] such that $f(0) = f(1)$ and $f(x) \geq f(y)$ for $x > y$. We call this function a basic unit interval monotonic function (BUM). Using this BUM function we form the OWAIMAM weights w_j for $j = 1$ to n as

$$w_j = f\left(\frac{j}{n}\right) - f\left(\frac{j-1}{n}\right). \tag{32}$$

It is easy to see that using this method, the w_j satisfy that the sum of the weights is 1 and $w_j \in [0,1]$.

Remark 9. Another interesting family is the nonmonotonic-OWAIMAM operator based on [18]. It is possible to form it when at least one of the weights w_j is lower than 0 and $\sum_{j=1}^{n} w_j = 1$. Note that a key aspect of this operator is that it does not always accomplish the monotonicity property. Then, this property could not be included in this special type of the OWAIMAM operator.

5. Using the Hybrid Average in the IMAM Operator

A further generalization of the OWAIMAM operator can be introduced by using the HA operator [26]. Thus, we can use in the IMAM operator, the weighted average and the OWA operator, considering both the attitudinal character of the decision maker and its subjective probability (or degree of importance). This new approach is called the hybrid averaging IMAM (HAIMAM) operator. Before defining the HAIMAM operator, let us briefly recall the definition of the HA operator.

Definition 7. A HA operator of dimension n is a mapping HA: $R^n \rightarrow R$ that has an associated weighting vector W of dimension n with $\sum_{j=1}^{n} w_j = 1$ and $w_j \in [0, 1]$, such that:

$$HA\ (a_1, a_2, \ldots, a_n) = \sum_{j=1}^{n} w_j b_j\ , \tag{33}$$

where b_j is the jth largest of the $\hat{a}_i$ ($\hat{a}_i = n\omega_i a_i$, $i = 1,2,\ldots,n$), $\omega = (\omega_1, \omega_2, \ldots, \omega_n)^T$ is the weighting vector of the a_i, with $\omega_i \in [0, 1]$ and the sum of the weights is 1.

With this introduction, the HAIMAM operator can be introduced as follows. Note that the main advantage of this approach is that the WIMAM and the OWAIMAM operators can be used in the same formulation.

Definition 8. A HAIMAM operator of dimension n is a mapping $HAIMAM$: $[0, 1]^n \times [0, 1]^n \rightarrow [0, 1]$ that has an associated weighting vector W of dimension n with $\sum_{j=1}^{n} w_j = 1$ and $w_j \in [0, 1]$, such that:

$$HAIMAM\ (a_1, a_2, \ldots, a_n) = \sum_{j=1}^{n} w_j K_j\ , \tag{34}$$

where K_j represents the jth largest of all the $|x_i - y_i|^* = n\omega_i |x_i - y_i|$ and the $[0 \vee (x_i - y_i)]^* = n\omega_i [0 \vee (x_i - y_i)]$, with $i = 1, 2, \ldots, n$, $\omega = (\omega_1, \omega_2, \ldots, \omega_n)^T$ is the weighting vector of the a_i, with $\omega_i \in [0, 1]$ and the sum of the weights is 1.

As we can see, if $w_j = 1/n$, for all j, we obtain the WIMAM operator and if $\omega_i = 1/n$, for all i, the OWAIMAM operator. If $w_j = 1/n$ and $\omega_i = 1/n$, for all i and j, the NIMAM operator is obtained.

The HAIMAM operator accomplishes similar properties than the OWAIMAM operator. However, it is not idempotent nor commutative. Moreover, we can also study a wide range of families of HAIMAM operators following the methodology explained in Section 4.

6. Choquet Integrals in the IMAM Operator

By using Choquet integrals [28-32], another generalization of the IMAM operator can be developed. It is called the Choquet integral IMAM aggregation (CIIMAMA). Before introducing this new result, let us define the concept of fuzzy measure and the discrete Choquet integral. The fuzzy measure was introduced by Sugeno [50] and it can be defined as follows.

Definition 9. Let X be a universal set $X = \{x_1, x_2, \ldots, x_n\}$ and $P(X)$ the power set of X. A fuzzy measure on X is a set function on m: $P(X) \rightarrow [0, 1]$, that satisfies the following conditions:

$m(\varnothing) = 0$, $m(X) = 1$ (boundary conditions) and
If $A, B \in P(X)$ and $A \subseteq B$, then $m(A) \leq m(B)$ (monotonicity).

The discrete Choquet integral [28] can be defined in the following way.

Definition 10. Let f be a positive real-valued function f: $X \to R^+$ and m be a fuzzy measure on X. The (discrete) Choquet integral of f with respect to m is:

$$C_m(f_1, f_2, \ldots, f_n) = \sum_{i=1}^{n} f_{(i)}\left[m(A_{(i)}) - m(A_{(i-1)})\right], \tag{35}$$

where $(\cdot)$ indicates a permutation on domain and range X such that $f_{(1)} \geq f_{(2)} \geq \ldots \geq f_{(n)}$, i.e. $f_{(i)}$ is the ith largest value in the finite set $\{f_1, f_2, \ldots, f_n\}$, $A_{(i)} = \{x_{(1)}, \ldots, x_{(i)}\}$ $i \geq 1$ and $A_{(0)} = \varnothing$ being $\{x_{(1)}, \ldots, x_{(i)}\}$ in the domain of f such that $f(x_i) = f_i$.

With this information, we can present the CIIMAMA operator as an aggregation operator that uses the Choquet integral and the IMAM operator in the same formulation. It can be defined as follows.

Definition 11. Let m be a fuzzy measure on X. A Choquet integral index of maximum and minimum level aggregation (CIIMAMA) operator of dimension n is a function $CIIMAMA$: $R^n \times R^n \to R$, such that:

$$CIIMAMA(\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_n, y_n\rangle) = \sum_{j=1}^{n} b_j\left[m(A_{(i)}) - m(A_{(i-1)})\right], \tag{36}$$

where b_j is the jth largest of all the $|x_i - y_i|$ and the $[0 \vee (x_i - y_i)]$, the x_i and the y_i are the argument variables of the sets $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_n\}$, $A_{(i)} = \{x_{(1)} \ldots, x_{(i)}\}$, $i \geq 1$ and $A_{(0)} = \varnothing$.

This approach can be generalized by using generalized and quasi-arithmetic means [15,17]. For example, by using quasi-arithmetic means, we get the quasi-arithmetic Choquet integral index of maximum and minimum level aggregation (Quasi-CIIMAMA) operator. It can be defined as follows.

Definition 12. Let m be a fuzzy measure on X. A Quasi-CIIMAMA operator of dimension n is a function $QICDIA$: $R^n \times R^n \times R^n \to R$, such that:

$$QICDIA(\langle u_1, x_1, y_1\rangle, \ldots, \langle u_n, x_n, y_n\rangle) = g^{-1}\left(\sum_{j=1}^{n} g(b_j)\left[m(A_{(i)}) - m(A_{(i-1)})\right]\right), \tag{37}$$

where g is a strictly continuous monotonic function, b_j is the jth largest of all the $|x_i - y_i|$ and the $[0 \vee (x_i - y_i)]$, the x_i and the y_i are the argument variables of the sets $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_n\}$, $A_{(i)} = \{x_{(1)} \ldots, x_{(i)}\}$, $i \geq 1$ and $A_{(0)} = \varnothing$.

7. Multi-Person Decision Making with the OWAIMAM Operator

The OWAIMAM operator can be applied in a wide range of fields. In this paper, we consider a decision-making application in the selection of strategies by using a multi-person analysis. Note that in the literature we find a wide range of methods for decision making [43,45,51-53]. The process to follow can be summarized as follows.

Step 1: Let $A = \{A_1, A_2, \ldots, A_n\}$ be a set of finite alternatives, $C = \{C_1, C_2, \ldots, C_n\}$ a set of finite characteristics (or attributes), forming the matrix $(x_{hi})_{m\times n}$. Let $E = \{E_1, E_2, \ldots, E_p\}$ be a finite set of decision makers. Let $V = (v_1, v_2, \ldots, v_p)$ be the weighting vector of the decision makers such that $\sum_{k=1}^{p} v_k = 1$ and $v_k \in [0, 1]$. Each decision maker provides his own payoff matrix $(x_{hi}^{(k)})_{m\times n}$.

Step 2: Calculate the ideal values of each characteristic in order to form the ideal strategy shown in Table 2. Note that the ideal strategy is an unreal strategy where we imagine an optimal situation where we are able to reach all our objectives, perfectly.

Table 2. Ideal strategy.

	C_1	C_2	…	C_i	…	C_n
I =	y_1	y_2	…	y_i	…	y_n

where I is the ideal strategy expressed by a fuzzy subset, C_i is the ith characteristic to consider and $y_i \in [0, 1]$, $i = 1, 2, \ldots, n$, is a number between 0 and 1 for the ith characteristic. Each decision maker provides his own ideal strategy $y_i^{(k)}$.

Step 3: Calculate the weighting vector W to be used in the OWAIMAM aggregation for each alternative h and characteristic i. Note that $W = (w_1, w_2, \ldots, w_n)$ such that $\sum_{j=1}^{n} w_j = 1$ and $w_j \in [0, 1]$.

Step 4: Comparison between the ideal strategy and the different characteristics considered using the OWAIMAM operator for each expert (person).

Step 5: Use the weighted average (WA) to aggregate the information of the decision makers E by using the weighting vector V. The result is the collective payoff matrix $(x_{hi}, y_{hi})_{m\times n}$. Thus, $(x_{hi}, y_{hi}) = \sum_{k=1}^{p} v_k (x_{hi}^k, y_{hi}^k)$. Note that (x_{hi}, y_{hi}) represents either $|x_{hi} - y_{hi}|$ or $[0 \vee (x_{hi} - y_{hi})]$, for the comparison of each tuple of arguments.

Step 6: Calculate the aggregated results by using the OWAIMAM operator explained in Eq. (4). Consider different types of OWAIMAM operators by using different expressions in the weighting vector as it has been explained in Section 4.

Step 7: Select the alternative/s that provides the best result/s. Moreover, establish a ranking of the alternatives from the most to the less preferred alternative in order to be able to consider more than one selection.

Note that this decision-making process can be summarized using the following aggregation operator that it is called the multi-person – OWAIMAM (MP-OWAIMAM) operator.

Definition 13. A MP-OWAIMAM operator is an aggregation operator that has a weighting vector V of dimension p with $\sum_{k=1}^{p} v_k = 1$ and $v_k \in [0, 1]$ and a weighting vector W of dimension n with $\sum_{j=1}^{n} w_j = 1$ and $w_j \in [0, 1]$, such that:

$$f(\langle(x_1^1, \ldots, x_1^p), (y_1^1, \ldots, y_1^p)\rangle, \ldots, \langle(x_n^1, \ldots, x_n^p), (y_n^1, \ldots, y_n^p)\rangle) = \sum_{j=1}^{n} w_j b_j, \quad (38)$$

where b_j is the jth largest of all the similarities (x_i, y_i) (either $|x_i - y_i|$ or $[0 \vee (x_i - y_i)]$), $(x_i, y_i) = \sum_{k=1}^{p} v_k (x_i^k, y_i^k)$, (x_i, y_i) is the argument variable provided by each person (or expert) represented in the form of individual similarities.

The MP-OWAIMAM operator has similar properties than those explained in Section 3 such as the measures for characterizing the weighting vector W, the distinction between descending and ascending orders, and so on.

The MP-OWAIMAM operator includes a wide range of particular cases following the methodology explained in Section 4. Thus, we can find as special cases:

The multi-person – normalized Hamming distance (MP-NHD) operator.
The multi-person – weighted Hamming distance (MP-WHD) operator.
The multi-person – OWAD (MP-OWAD) operator.
The multi-person – normalized adequacy coefficient (MP-NAC) operator.
The multi-person – weighted adequacy coefficient (MP-WAC) operator.
The multi-person – OWAAC (MP-OWAAC) operator.
The multi-person – NIMAM operator.
The multi-person – OWA (MP-OWA) operator.

Additionally, it is possible to consider more complex formulations by using other types of aggregation operators in the aggregation of the experts opinion because in Definition 12, we assume that the experts opinions are aggregated by using the WA operator. Moreover, note that it is possible to develop a similar model by using Choquet integrals obtaining the multi-person – CIIMAMA (MP-CIIMAMA) operator and by using hybrid averages, obtaining the multi-person – HAIMAM (MP-HAIMAM) operator.

8. Numerical Example

In the following, we are going to present an illustrative example where we will see the applicability of the new approach. We consider a decision making problem regarding the selection of strategies. Different types of OWAIMAM

operators such as the NIMAM, the WIMAM, the OWAIMAM, the AOWAIMAM and the olympic-OWAIMAM are used. The dual results are also considered.

Assume an enterprise that operates in Europe and in North America is considering an expansion for the next year and they consider 5 strategies to follow.

A_1: Expand to Asian market.
A_2: Expand to the South American market.
A_3: Expand to the African market.
A_4: Expand to the Oceanian market.
A_5: Expand to the 4 continents.
A_6: Do not develop any expansion.

In order to evaluate these strategies, the company has brought together a group of experts. They consider different characteristics about the strategies that can be summarized in the following ones: C_1 = Risk of the strategy; C_2 = Difficulty; C_3 = Benefits in the short term; C_4 = Benefits in the mid term; C_5 = Benefits in the long term; C_6 = Other characteristics.

Table 3. Payoff matrix – Expert 1.

	C_1	C_2	C_3	C_4	C_5	C_6
A_1	0.7	0.8	0.9	0.9	0.3	0.6
A_2	0.9	0.7	0.7	0.5	0.5	1
A_3	1	0.5	0.7	0.8	0.6	0.7
A_4	0.7	0.5	0.6	0.7	0.8	0.8
A_5	0.9	0.7	0.2	0.7	0.8	0.8
A_6	0.6	0.8	0.7	0.8	0.7	0.7

Table 4. Payoff matrix – Expert 2.

	C_1	C_2	C_3	C_4	C_5	C_6
A_1	0.6	0.7	0.8	0.9	0.7	0.6
A_2	0.8	0.6	0.7	0.6	0.5	1
A_3	0.9	0.7	0.6	0.8	0.6	0.7
A_4	0.6	0.5	0.6	0.7	0.8	0.9
A_5	0.6	0.7	0.5	0.8	0.8	0.7
A_6	0.7	0.8	0.7	0.9	0.5	0.7

Table 5. Payoff matrix – Expert 3.

	C_1	C_2	C_3	C_4	C_5	C_6
A_1	0.4	0.7	0.8	0.8	0.7	0.8
A_2	0.8	0.7	0.7	0.6	0.5	1
A_3	0.8	0.7	0.6	0.8	0.5	0.8
A_4	0.6	0.5	0.6	0.8	0.6	0.9
A_5	0.5	0.7	0.4	0.8	0.7	0.8
A_6	0.7	0.8	0.6	0.9	0.6	0.6

The group of experts of the company is constituted by three persons that give its own opinion regarding the results obtained with each strategy. The results are shown in Tables 3, 4 and 5. Note that the results are valuations (numbers) between 0 and 1.

According to the objectives of the decision-maker, each expert establishes his own ideal strategy. The results are shown in Table 6.

Table 6. Collective results.

	C_1	C_2	C_3	C_4	C_5	C_6
A_1	0.28	0.07	0.07	0.11	0.38	0.26
A_2	0	0.13	0.2	0.4	0.46	0
A_3	0.03	0.16	0.27	0.17	0.4	0.2
A_4	0.2	0.3	0.3	0.23	0.24	0.07
A_5	0.18	0.1	0.53	0.2	0.2	0.17
A_6	0.16	0	0.24	0.1	0.36	0.28

With this information, we can aggregate it in order to make a decision. First, the information of the three experts is aggregated in order to obtain a collective matrix represented in the form of individual similarities between the available alternatives and the ideal ones. We use the WA to obtain this matrix and assuming that $V = (0.2, 0.4, 0.4)$. The results are shown in Table 7.

Table 7. Ideal strategy.

	C_1	C_2	C_3	C_4	C_5	C_6
E_1	0.9	0.8	0.9	0.9	1	0.9
E_2	0.8	0.8	0.9	1	1	0.9
E_3	0.8	0.8	0.9	1	0.9	1

The group of experts considers the following weighting vector for all the cases: $W = (0.1, 0.1, 0.1, 0.2, 0.2, 0.3)$. In this example, we assume that the group of experts considers the three first characteristics with the Hamming distance and the other three with the adequacy coefficient. The usefulness of the IMAM is that we can use the Hamming distance or the adequacy coefficient depending on the particular interests of the decision maker in the analysis.

With this information, we can aggregate the expected results in order to obtain a representative result for each alternative. First, we consider the NIMAM, the WIMAM, the OWAIMAM, the AOWAIMAM and the olympic-OWAIMAM. Note that in the olympic-OWAIMAM, we consider $w_1 = w_6 = 0$, and for all others $w_j = 1/(n - 2)$. Then, in this case, we have: $w_2 = w_3 = w_4 = w_5 = 0.25$. The results are shown in Table 8.

Table 8. Aggregated results 1.

	NIMAM	WIMAM	OWAIMAM	AOWAIMAM	Olympic
A_1	0.195	0.218	0.149	0.247	0.18
A_2	0.198	0.205	0.132	0.271	0.1825
A_3	0.205	0.22	0.162	0.25	0.2
A_4	0.223	0.195	0.191	0.248	0.2425
A_5	0.23	0.212	0.193	0.284	0.1875
A_6	0.19	0.216	0.14	0.238	0.195

Now, the results obtained by using the OWADIMAM operator are considered. Obviously, the results obtained are the dual of the previous ones. The results are shown in Table 9.

Table 9. Aggregated results 2.

	NDIMAM	WDIMAM	OWADIMAM	AOWADIMAM	Olympic
A_1	0.805	0.782	0.851	0.753	0.82
A_2	0.802	0.795	0.868	0.729	0.8175
A_3	0.795	0.78	0.838	0.75	0.8
A_4	0.777	0.805	0.809	0.752	0.7575
A_5	0.77	0.788	0.807	0.716	0.7125
A_6	0.81	0.784	0.86	0.762	0.805

As we can see, depending on the aggregation operator used the results are different. A_6 is optimal choice with the NIMAM and the AOWAIMAM operator, A_1 with the olympic-OWAIMAM, A_4 with the WIMAM and A_2 with the OWAIMAM operator. Obviously, the same results are found with the dual indexes.

Another interesting issue is to establish an ordering of the alternatives. Note that this is useful when we want to consider more than one alternative. The results are shown in Table 10.

Table 10. Ordering of the strategies.

	Ordering		Ordering
NIMAM	$A_6 \succ A_1 \succ A_2 \succ A_3 \succ A_4 \succ A_5$	NDIMAM	$A_6 \succ A_1 \succ A_2 \succ A_3 \succ A_4 \succ A_5$
WIMAM	$A_4 \succ A_2 \succ A_5 \succ A_6 \succ A_1 \succ A_3$	WDIMAM	$A_4 \succ A_2 \succ A_5 \succ A_6 \succ A_1 \succ A_3$
OWAIMAM	$A_2 \succ A_6 \succ A_1 \succ A_3 \succ A_4 \succ A_5$	OWADIMAM	$A_2 \succ A_6 \succ A_1 \succ A_3 \succ A_4 \succ A_5$
AOWAIMAM	$A_6 \succ A_1 \succ A_4 \succ A_3 \succ A_2 \succ A_5$	AOWADIMAM	$A_6 \succ A_1 \succ A_4 \succ A_3 \succ A_2 \succ A_5$
Olympic	$A_1 \succ A_2 \succ A_5 \succ A_6 \succ A_3 \succ A_4$	Olympic	$A_1 \succ A_2 \succ A_5 \succ A_6 \succ A_3 \succ A_4$

As we can see, depending on the aggregation operator used, the ordering of the strategies is different. Thus, these results may lead to different decisions.

9. Conclusions

We have analyzed the use of the OWA operator in the index of maximum and minimum level. As a result, we have obtained a new aggregation operator: the OWAIMAM operator. This operator is very useful because it provides a parameterized family of aggregation operators in the IMAM operator that includes the maximum, the minimum and the average. The main advantage of the OWAIMAM is that we can manipulate the neutrality of the aggregation so the decision maker can be more or less optimistic according to his interests. We have studied some of its main properties.

We have further extended the OWAIMAM operator by using the HA operator, obtaining the HAIMAM operator. We have seen that this operator is more general because it includes the weighted average and the OWA operator in the same formulation.

We have also studied another extension by using the Choquet integral. We have called it the CIIMAMA operator. Moreover, we have presented a further generalization by using quasi-arithmetic means, the Quasi-CIIMAMA operator.

We have applied the new approach in a multi-person decision-making problem about selection of strategies. We have seen that sometimes, depending on the particular type of OWAIMAM operator used, the results are different. Thus, the decisions of the decision maker may be also different. Moreover, we have developed the MP-OWAIMAM operator as a more general aggregation operator for the multi-person decision-making process.

In future research, we expect to develop further extensions of the OWAIMAM operator by adding new characteristics in the problem such as the use of order inducing variables and applying it to other decision making problems such as product management and investment selection.

Acknowledgements. We would like to thank the anonymous reviewers for valuable comments that have improved the quality of the paper. Support from the projects JC2009-00189 and A/016239/08 is gratefully acknowledged.

References

1. Gil-Lafuente, J.: The index of maximum and minimum level in the selection of players in sport management (In Spanish). Proceedings 10th International Conference of the European Academy of Management and Business Economics (AEDEM), Reggio Calabria, Italy, pp.439-443. (2001)
2. Gil-Aluja, J.: The interactive management of human resources in uncertainty. Kluwer Academic Publishers, Dordrecht, Netherlands. (1998)
3. Gil-Lafuente, A. M.: Fuzzy logic in financial analysis. Springer, Berlin, Germany. (2005)

4. Kaufmann, A.; Gil-Aluja, J.: Introduction to the theory of fuzzy subsets in business management (In Spanish). Milladoiro, Santiago de Compostela, Spain. (1986)
5. Merigó, J. M.; Gil-Lafuente, A. M.: The generalized adequacy coefficient and its application in strategic decision making. Fuzzy Economic Review, Vol. 13, No. 2, 17-36. (2008)
6. Merigó, J. M.; Gil-Lafuente, A. M.: New decision-making techniques and their application in the selection of financial products. Information Sciences, Vol. 180, No. 11, 2085-2094. (2010)
7. Gil-Lafuente, A. M.; Merigó, J. M.: Computational Intelligence in Business and Economics. World Scientific, Singapore. (2010)
8. Gil-Lafuente, J.: Algorithms for excellence: Keys for being successful in sport management (In Spanish). Milladoiro, Vigo, Spain. (2002)
9. Yager, R. R.: On ordered weighted averaging aggregation operators in multi-criteria decision making. IEEE Transactions on Systems, Man and Cybernetics B, Vol. 18, No. 1, 183-190. (1988)
10. Beliakov, G., Pradera, A., Calvo, T.: Aggregation Functions: A Guide for Practitioners. Springer-Verlag, Berlin, Germany. (2007)
11. Canós, L., Liern, V.: Soft Computing-Based Aggregation Methods for Human Resource Management. European Journal of Operational Research, Vol. 189, No. 3, 669-681. (2008)
12. Kacprzyk, J.; Zadrozny, S.: Towards a generalized and unified characterization of individual and collective choice functions under fuzzy and nonfuzzy preferences and majority via ordered weighted average operators. International Journal of Intelligent Systems, Vol. 24, No. 1, 4-26. (2009)
13. Merigó, J. M.; Casanovas, M.: The fuzzy generalized OWA operator and its application in strategic decision making. Cybernetics & Systems, Vol. 41, No. 5, 359-370. (2010)
14. Merigó, J. M.; Casanovas, M.: Induced and heavy aggregation operators with distance measures. Journal of Systems Engineering and Electronics, Vol. 21, No. 3, 431-439. (2010)
15. Merigó, J. M.; Casanovas, M.: The uncertain induced quasi-arithmetic OWA operator. International journal of Intelligent Systems, Vol. 26, No. 1, 1-24. (2011)
16. Merigó, J. M.; Casanovas, M.: Induced and uncertain heavy OWA operators. Computers & Industrial Engineering, Vol. 60, No. 1, 106-116. (2011)
17. Merigó, J. M.; Gil-Lafuente, A. M.: The induced generalized OWA operator. Information Sciences, Vol. 179, No. 6, 729-741. (2009)
18. Yager, R. R.: Nonmonotonic OWA operators. Soft Computing, Vol. 3, No. 3, 187-196. (1999)
19. Yager, R. R.: Heavy OWA operators. Fuzzy Optimization and Decision Making, Vol. 1, No. 4, 379-397. (2002)
20. Yager, R. R.: Norms induced from OWA operators. IEEE Transactions on Fuzzy Systems, Vol. 18, No. 1, 57-66. (2010)
21. Yager, R. R.; Kacprzyk, J.: The ordered weighted averaging operators: Theory and Applications. Kluwer Academic Publishers, Norwell, MA, USA. (1997)
22. Merigó, J. M.; Casanovas, M.: Fuzzy generalized hybrid aggregation operators and its application in decision making. International Journal of Fuzzy Systems, Vol. 12, No. 1, 15-24. (2010)
23. Merigó, J. M.; Casanovas, M.; Martínez, L.: Linguistic aggregation operators for linguistic decision making based on the Dempster-Shafer theory of evidence.

International Journal of Uncertainty, Fuzziness Knowledge-Based Systems, Vol. 18, No. 3, 287-304. (2010)
24. Wei, G. W.: Some induced geometric aggregation operators with intuitionistic fuzzy information and their application to group decision making. Applied Soft Computing, Vol. 10, No. 1, 423-431. (2010)
25. Wei, G. W.; Zhao, X.; Lin, R.: Some induced aggregating operators with fuzzy number intuitionistic fuzzy information and their applications to group decision making. International Journal of Computational Intelligence Systems, Vol. 3, No. 1, 84-95. (2010)
26. Xu, Z. S.; Da, Q. L.: An overview of operators for aggregating information. International Journal of Intelligent Systems, Vol. 18, No. 9, 953-969. (2003)
27. Zhao, H.; Xu, Z. S.; Ni, M.; Liu, S.: Generalized aggregation operators for intuitionistic fuzzy sets. International Journal of Intelligent Systems, Vol. 25, No. 1, 1-30. (2010)
28. Choquet, G.: Theory of Capacities. Annals Institute Fourier, Vol. 5, 131-295. (1953)
29. Merigó, J. M.; Casanovas, M.: Decision making with distance measures and induced aggregation operators. Computers & Industrial Engineering, Vol. 60, No. 1, 66-76. (2011)
30. Mesiar, R.: Choquet-like integrals. Journal of Mathematical Analysis and Applications, Vol. 194, No. 2, 477-488. (1995)
31. Tan, C.; Chen, X.: Induced Choquet ordered averaging operator and its application to group decision making. International journal of Intelligent Systems, Vol. 25, No. 1, 59-82. (2010)
32. Xu, Z. S.: Choquet integrals of weighted intuitionistic fuzzy information. Information Sciences, Vol. 180, No. 5, 726-736. (2010)
33. Figueira, J.; Greco, S.; Ehrgott, M.: Multiple criteria decision analysis: State of the art surveys. Springer, Boston, USA. (2005)
34. Gil-Aluja, J.; Gil-Lafuente, A. M.; Klimova, A.: M-attributes algorithm for the selection of a company to be affected by a public offering. International Journal of Uncertainty, Fuzziness Knowledge-Based Systems, Vol. 17, No. 3, 333-343. (2009)
35. Merigó, J. M.: Fuzzy decision making with immediate probabilities. Computers & Industrial Engineering, Vol. 58, No. 4, 651-657. (2010)
36. Merigó, J. M.; Casanovas, M.: Induced aggregation operators in decision making with the Dempster-Shafer belief structure. International Journal of Intelligent Systems, Vol. 24, No. 8, 934-954. (2009)
37. Xu, Y. J.; Da, Q. L.; Liu, X.: Some properties of linguistic preference relation and its ranking in group decision making. Journal of Systems Engineering and Electronics 21, No. 2, 244-249. (2010)
38. Xu, Z. S.: A method based on distance measure for interval-valued intuitionistic fuzzy group decision making. Information Sciences, Vol. 180, No. 1, 181-190. (2010)
39. Xu, Z. S.: A deviation-based approach to intuitionistic fuzzy multiple attribute group decision making. Group Decision and Negotiation, Vol. 19, No. 1, 57-76. (2010)
40. Xu, Z. S.; Cai, X.: Nonlinear optimization models for multiple attribute group decision making with intuitionistic fuzzy information. International Journal of Intelligent Systems, Vol. 25, No. 6, 489-513. (2010)

41. Xu, Z. S.; Hu, H.: Projection models for intuitionistic fuzzy multiple attribute decision making. International Journal of Information Technology and Decision Making, Vol. 9, No. 2, 267-280. (2010)
42. Zhang, R.; Liu, B.; Liu, S.: A multi-attribute auction model by dominance-based rough sets approach. Computer Science and Information Systems, Vol. 7, No. 4, 843-858. (2010)
43. Wei, G. W.: A method for multiple attribute group decision making based on the ET-WG and ET-OWG operators with 2-tuple linguistic information. Expert Systems with Applications, Vol. 37, No. 12, 7895-7900. (2010)
44. Zarghami, M.; Szidarovsky, F.: On the relation between compromise programming and ordered weighted averaging operator. Information Sciences, Vol. 180, No. 11, 2239-2248. (2010)
45. Zhou, L. G.; Chen, H. Y.: Generalized ordered weighted logarithm aggregation operators and their applications to group decision making. International Journal of Intelligent Systems, Vol. 25, No. 7, 683-707. (2010)
46. Merigó, J. M.; Gil-Lafuente, A. M.: Unification point in methods for the selection of financial products. Fuzzy Economic Review, Vol. 12, No. 1, 35-50. (2007)
47. Ahn, B. S., Park, H.: Parameterized OWA Operator Weights: An Extreme Point Approach. International Journal of Approximate Reasoning, Vol. 51, No. 7, 820-831. (2010)
48. Emrouznejad, A.; Amin, G. R.: Improving minimax disparity model to determine the OWA operator weights. Information Sciences, Vol. 180, No. 8, 1477-1485. (2010)
49. Yager, R. R.: Families of OWA operators. Fuzzy Sets and Systems, Vol. 59, No. 2, 125-148. (1993)
50. Sugeno, M.: Theory of fuzzy integrals and its applications. PhD Thesis, Tokyo Institute of Technology, Japan. (1974)
51. Merigó, J.M.: A unified model between the weighted average and the induced OWA operator. Expert Systems with Applications, Vol. 38, No. 9, 11560-11572. (2011)
52. Merigó, J. M.; Gil-Lafuente, A. M.: Fuzzy induced generalized aggregation operators and its application in multi-person decision making. Expert Systems with Applications. Vol. 38, No. 8, 9761-9772. (2011)
53. Merigó, J. M.; Gil-Lafuente, A. M.; Barcellos, L.: Uncertain induced generalized aggregation operators and its application in the theory of expertons. Fuzzy Economic Review, Vol. 15, No. 2, 25-42. (2010)

José M. Merigó is an Assistant Professor in the Department of Business Administration at the University of Barcelona, Spain. He received a MSc and a PhD degree (with Extraordinary Award) in Business Administration from the University of Barcelona. He also holds a Bachelor Degree in Economics from Lund University, Sweden. He has published more than 150 papers in journals, books and conference proceedings including journals such as *Information Sciences*, *International Journal of Intelligent Systems*, and *Computers & Industrial Engineering*. He is on the editorial board of several journals and scientific committees and serves as a reviewer in a wide range of journals. He is interested in Aggregation Operators, Decision Making and Uncertainty.

Anna M. Gil-Lafuente is an Associate Professor in the Department of Business Administration at the University of Barcelona, Spain. She received a MSc and a PhD degree in Business Administration from the University of Barcelona. She has published more than 150 papers in journals, books and conference proceedings including journals such as *Information Sciences*, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* and *Expert Systems with Applications*. She is co-editor-in-chief of 8 journals published by the Association for Modeling and Simulation in Enterprises (AMSE). She is currently interested in Decision Making, Uncertainty and Finance.

Received: June 6, 2011; Accepted: August 8, 2011.

DOI 10.2298/CSIS101109004J

A Generic Parser for Strings and Trees

Riad S Jabri

Computer Science Department
King Abdullah II School for Information Technology
University of Jordan, Amman 11942 –Jordan
jabri@ju.edu.jo

Abstract. In this paper, we propose a two fold generic parser. First, it simulates the behavior of multiple parsing automata. Second, it parses strings drawn from either a context free grammar, a regular tree grammar, or from both. The proposed parser is based on an approach that defines an extended version of an automaton, called position-parsing automaton (PPA) using concepts from LR and regular tree automata, combined with a newly introduced concept, called state instantiation and transition cloning. It is constructed as a direct mapping from a grammar, represented in an expanded list format. However, PPA is a non-deterministic automaton with a generic bottom–up parsing behavior. Hence, it is efficiently transformed into a reduced one (RBA). The proposed parser is then constructed to simulate the run of the RBA automaton on input strings derived from a respective grammar. Without loss of generality, the proposed parser is used within the framework of pattern matching and code generation. Comparisons with similar and well-known approaches, such as LR and RI, have shown that our parsing algorithm is conceptually simpler and requires less space and states.

Keywords: bottom-up automata, parsing, regular tree grammars.

1. Introduction

Without loss of generality, in this paper we propose a generic parser within the framework of code generators, in which machine instructions are specified by patterns that are drawn either from context-free grammars (LALR), regular tree grammars or from rewrite systems [1, 4, 5,12,15]. Pattern matching is then performed using respective parsing automata to generate an optimal set of machine instructions [6, 15]. The LALR parsing automata are efficient but too restrictive to handle patterns drawn from ambiguous context-free grammars. In contrast, the tree parsers are not restrictive but their optimization is a complex task. Approaches to handle ambiguous context-free grammars have been suggested in [14]. However, they are not intended to handle regular tree grammars. On the other hand, approaches to combine LR parsing and bottom-up tree parsing do exist, such as the one suggested in [9]. It is based on transforming a tree automaton recognizing a regular tree

language to a pushdown automaton recognizing the same language in postfix notation. In addition to the transformation overhead, this approach assumes that the transformed grammar is in normal form, deterministic and without hidden-left and right recursion [9]. Hence, there is a need for a parsing approach that admits ambiguous grammar and maintains the efficiency of LALR parsers and the expressive power of tree parsers. To satisfy such a need, we propose an approach for a hybrid parsing of both general context-free and regular tree grammars. According to the proposed approach, and as a framework for parsing, a generic grammar (GG) is defined as one that can be either instantiated by a context-free grammar or by a regular tree grammar. GG is then represented in an expanded list format (PLF). As such, PLF constitutes a prefix representation of derivation trees for the grammar GG, where the ranked terminals are considered as nonterminals and the recursive terminal symbols are terminalized. Hence, the repeated expansion in the derivation trees is incorporated as recursion-invocation and recursion-termination. The recursion-invocation constitutes an ε-transition from the recursive occurrence to its respective head, while recursion-termination implies an ε-transition from the recursive-head to its respective occurrence. Hence, the recursive occurrence initiates a derivation/reduction path as the one initiated by its respective head and considered as an instance of the original one**.** To distinguish between instances initiated by different recursive occurrences, an instance-identifier (ID) is assigned to each initiated path. Therefore, PLF with incorporated recursion implies initiation and termination of derivations/ reductions paths, as well as respective instances of these paths. A nondeterministic parsing automaton, called position parsing automaton (PPA), is then constructed in terms of a set of state instances and respective parsing actions as a direct mapping from the PLF of the production respective to the start symbol S of GG. PPA is defined based on a newly introduced concept of state and transition instances. According to such concept, Each PPA state (q, (inst)) is defined with an implicit index (inst) = ε. State instances are then created and terminated by appending and deleting instance-identifiers (ID) atop of the index (inst). The PPA transitions are augmented by semantic-actions that are performed at run time to create and terminate instances of PPA states and transitions in accordance to the incorporated recursion-invocation and recursion-termination. In addition to the newly introduced concept of state and transition instantiation, the states and transitions of the PPA automaton are defined based on combined concepts from LR (0) automata, regular tree automata. PPA has been adopted in [8] to construct a pattern matcher, where its parsing behavior has been adapted and synchronized with pattern matching and code selection. In this research, we emphasize the generality and soundness of PPA parsing behavior. Hence, we extend its behavior to cover subtle grammar cases. This includes direct and indirect (hidden) left / right recursion and ambiguous grammars with shift/reduce and reduce/reduce conflicts. In addition, we optimize its mapping from PLF by following a concurrent and a gradual construction approach for both PLF and PPA. As a simulator for tree parsing automata, finite automata and shift–reduce automata, the generic behavior of

PPA has been enriched by conceptual explanation, theory and graphical representation. However, PPA is a nondeterministic automaton. To obtain a more deterministic parsing behavior, PPA is efficiently transformed into a reduced one, called RBA, according to a proposed subset construction approach. The RBA automaton is represented by a parsing table that specifies the parsing actions respective to a given state and each input symbol. A parser is then constructed to simulate the run the RBA automaton on strings derived from a generic grammar. Furthermore, the proposed parser is based on formalizing the parsing process and its solution in a way that guarantees its soundness, generalization and its efficient implementation. This was demonstrated by the proposed theory and by the respective implementation algorithms.

Our parsing approach is motivated by reduction incorporated parsing introduced in [2, 3] and further developed in [10, 14]. According to these approaches, a finite automaton for terminalized grammar (RIA) is constructed to handle the regular parts of the language. Such automaton is then extended by recursive call automaton (RCA) to handle recursion. However, the suggested automaton in [2,3] does not handle hidden left recursion. In addition, the construction approach is based on by hand generated terminalization. Once terminalization has been detected, RCA is constructed. In [14] RIA has been extended to handle hidden-left recursion, while in [10] an automatic computation for terminalization has been suggested. In contrast, our approach handles hidden and direct left/right recursion. Terminalization is handled as recursion-invocation and recursion-termination during the concurrent construction of PLF and PPA. In addition, and rather than, the static creation of RCA, an instance is dynamically created during parsing. This achieved by the newly introduced concept of state instantiations and embedded semantic actions. Furthermore, our approach is embedded by appropriate concepts to parse regular trees. Further comparisons with these approaches and similar ones are given in Section 6.

The remainder of this paper is organized as follows. Section 2 presents preliminaries. Section 3 presents the proposed parsing approach, followed by the definitions of the newly introduced concepts. This includes the proposed nondeterministic parsing automaton (PPA) and the theory on which it is based. Section 4 presents the PPA construction algorithm. Section 5 presents the subset construction and subsequently the proposed parser, followed by a discussion and a conclusion that are given in section 6 and section 7 respectively.

2. Preliminaries

For our further discussions, we assume the following definitions based on the ones given in [1, 8, 11, 12, 13, 15].

Definition 1. The 4-tuple G = (Σ, N, P, S) defines a context-free grammar, where:

- Σ is an alphabet of terminal symbols.
- N is a finite set of nonterminal symbols, where $S \in N$ is the start symbol.
- P is a finite set of productions p having the form p: $A \rightarrow V$, where $A \in N$ and $V \in (\Sigma \cup N)^*$.

Definition 2. A ranked alphabet Σ is a finite set of symbols (operators, constructors) such that each member of Σ has a nonnegative integer, called a rank (arity). The rank is defined by the function *arity*: $\Sigma \rightarrow \mathbf{N}$, where

- **N** denotes the set of natural numbers.
- The terminals having a rank ≥ 1 are called operators and the ones having a rank =0 are called constants.
- The members of Σ are grouped into the subsets $\Sigma_0, \ldots, \Sigma_n$ such that $\Sigma_n = \{a \in \Sigma \mid \text{arity}(a) = i, i \in (0,1,\ldots,n)\}$.

Definition 3. A tree language over Σ, denoted by T (Σ), consists of all possible terms that are inductively defined as:

- If $a \in \Sigma_0$ then $a \in T(\Sigma)$
- If $t_1,\ldots, t_n \in T(\Sigma)$ and $a \in \Sigma_n$ then $a(t_1,\ldots, t_n) \in T(\Sigma)$.
- These terms represent trees with internal nodes and leaves labeled by operators and constants respectively. The number of children of a node labeled by an operator a is *arity* (a).

Definition 4. (Regular tree grammar). The 4-tuple G = (Σ, N, P, S) defines a regular tree grammar, where:

- Σ is a ranked alphabet
- N is a finite set of nonterminals with an assumed rank =0, where $S \in N$ is the start symbol.
- P is a finite set of productions p having the form p: $A \rightarrow V$, where $A \in N$ and $V \in T(\Sigma \cup N)$ is a prefix encoding of a tree over $(\Sigma \cup N)$.

We denote A and V by LHS(p) and RHS(p), to represent the left hand side and the right hand side of the production p respectively. Given p: $A \rightarrow V$ and $t1 \in T(\Sigma \cup N)$, we say t1 derives $t2 \in T(\Sigma \cup N)$ in one-step, denoted by $t1 \Rightarrow t2$, if LHS (p) $\in t1$ and RHS (p) $\in t2$. Applying zero or more such derivation steps on t1 is denoted by $t1 \Rightarrow^* t2$, where each step derives a tree $t \in T(\Sigma \cup N)$.

Definition 5. (Regular tree language). The language generated by the grammar G is defined by the set $L(G) = \{t \mid t \in T(\Sigma) \text{ and } S \Rightarrow^* t\}$.

Definition 6. (Regular tree parsing). Parsing a tree $t \in L(G)$ is the process of constructing a possible S– derivation tree for t, which is inductively defined as a tree with a root labeled by the start grammar symbol S and with children labeled by the RHS(S), where:

- The children labeled by terminals $\in \Sigma_0$ are leaves. The children labeled by nonterminals are interior nodes and constitute roots for V-derivation trees, each one of which is inductively defined as S-derivation tree.

- The children labeled by ranked terminals $\in \Sigma_n$ are interior nodes and constitute roots for sub trees, each one of which is inductively defined as S-derivation tree. However, the children of such sub trees are labeled by the grammar symbols composing the arity of their respective roots.

Definition 7. Generic Grammar (GG) A generic grammar is a 4-tuple GG= (Σ, N, P, S) that is either instantiated by a context-free grammar, as given by Definition 1, or by a regular tree grammar, as given by Definition 4. However, the ranked terminals are nonterminalized as given by Definition 8. Subsequently, in our further discussion, we use the following generic terms.

- Derivation tree (DT)**:** Denotes either an S-derivation tree or a classical one for the context free grammars.
- String: Denotes either a prefix encoding of the input trees or sentences generated by a context free grammar.

Definition 8. Given a $(t_1,\ldots,tn) \in T(\Sigma)$, the ranked terminal $(a \in \Sigma_n)$ is nonterminalized as a $\rightarrow$ t1… tn, where each ranked terminal in (t1,…, tn) is inductively nonterminalized.

Example 1. Given GG = (Σ, N, P, S), it can be instantiated with a regular tree grammar as follows:

- $\Sigma = \{\Sigma_0, \Sigma_1, \Sigma_2\}$, where Σ_0= {c}, Σ_1= {m} and Σ_2 = {+}
- N = {R}, R = S and P= { R→ +(m(c),R) | + (m(R),R) | c }
- Such grammar generates strings (trees) of the forms: c, + (m(c), c),…, +(m(c),+ (m(c),c)).
- GG can be instantiated with a context-free grammar as follows:
- $\Sigma = \Sigma_0$ = {a, b, c}; N = {S, A , B} and S is the start symbol
- P = {S → A b , A → a B , B → c | c B}
- Such grammar generates strings (sentences) of the forms: acb, accb, accccb,...

3. The proposed parsing approach

Given a generic grammar GG, we propose a parsing approach based on simulating the run of a reduced bottom--up automaton (RBA) on strings generated by GG. RBA is obtained as result of a subset construction applied on a proposed nondeterministic automaton (PPA). Such automaton is an extended version of a one introduced in [8]. It is constructed in terms of a set of states and parsing actions as a direct mapping from the production respective to the start symbol S of GG, which is represented in the expanded list format (PLF). According to PLF, a production p: $A \rightarrow \alpha$ in GG is defined as PLF (A) = A (α), where each nonterminal $\neq$ A in α is inductively replaced by the PLF respective to its corresponding production(s). The

recursive occurrence of the symbol A in RHS (p) is not replaced by a corresponding PLF. Only, it is designated as recursive instance of its respective head. In addition, each grammar symbol in PLF is assigned a doted integer as an index to reflect its occurrence order (position), as well as its nesting depth. As such, PLF constitutes a prefix representation of derivation trees for the grammar GG, where the repeated expansion in the derivation trees is represented as an incorporated recursion. Further, the positions assigned to the grammar symbols of PLF represent their paths in a respective derivation tree. Hence, the types of the grammar symbols and their occurrence order in PLF imply derivation/reduction relationships. The PPA states and parsing actions are then defined based on such relationships, using concepts from tree parsing and shift-reduce automata that are combined with newly introduced ones to handle the incorporation of recursion in PLF. Such concepts include state-instantiation, transition-cloning and transitions having embedded semantic-actions. In this section, we present the PLF form and PPA automaton. In the following sections, we present the subset construction of PPA into the reduced RBA automaton and the proposed parser. In addition and where appropriate, we present theory to demonstrate the generality and the soundness of the proposed approach. However, we immediately present an example to illustrate the proposed approach and to facilitate our further discussion.

Example 2. Consider the grammar of Example 1. A PLF form respective to the start grammar symbol S is defined as PLF (S) = S_0 (A_1 ($a_{1.1}$, $B_{1.2}$ ($c_{1.2.1}$)), b_2). It is a prefix representation of the derivation tree for GG, as shown in Fig. 1(a). The grammar symbols are attached an index to represent their respective positions, for examples: The index (0) attached to S, defines S as a head and the indexes assigned to A and b indicate that they constitute the first and the second subordinates of S respectively.

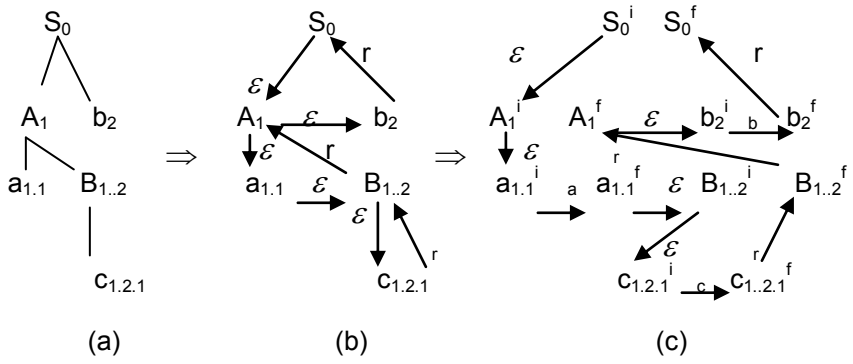


Fig. 1. (a) A derivation tree for the grammar GG of Example1. (b) Derivation/reduction relationships implied by PLF respective to GG. (c) PPA automaton respective to GG.

The derivations/reductions implied by PLF (S) are shown in Fig.1 (b) in terms of the nested paths $S_0 \xrightarrow{\varepsilon} A_1$ **(** $\xrightarrow{\varepsilon}$ ($a_{1.1} \xrightarrow{\varepsilon} B_{1..2}$ ($\xrightarrow{\varepsilon} c_{1.2.1}$) $\xrightarrow{r} B_{1..2}$) $\xrightarrow{r} A_1$ **)** $\xrightarrow{\varepsilon} b_2 \xrightarrow{r} S_0$, where the outermost path indicates the derivation $S_0 \Rightarrow A_1\, b_2$ and the reduction to S_0. The innermost paths indicate the derivations $A_1 \Rightarrow a_{1.1}\, B_{1.2}$ and $B_{1..2} \Rightarrow c_{1.2.1}$ as well as the reductions to $B_{1..2}$ and A_1. Further, the derivations are performed according to the rightmost derivation first, and the reductions according to the leftmost phrase first. Hence, a bottom-up parsing automaton (PPA), as shown in Fig. 1(c), can be constructed by a direct mapping from PLF(S) as the tuple (T, q_{in}, q_{fin}, Q, SPA, RPA), where:

- T = ($\Sigma \cup \varepsilon$) is the input alphabet, where Σ represent strings generated by the GG grammar.
- Q is the set of PPA states, where q_{in} and q_{fin} are the initial and the final ones.
- SPA and RPA are the sets of shift and reduce parsing actions respectively.

The PPA construction proceeds as follows:

- Each grammar symbol $V \in (\Sigma \cup N)$ is represented by a pair of abstract ones, defined as (V^i, V^f), to indicate a prediction and an acceptance of V in an assumed parsing. Consequently, respective PPA states are defined by the pair ($q^i = V^i$, $q^f = V^f$), where: q^i is an initial state instantiated by V^i and acts as a predictor (scanner) for V. The state q^f is a final one, instantiated by V^f and acts as its respective acceptor. Hence, the PPA states are constructed by the set Q = { ($q_p{}^i = Vp^i$, $q_p{}^f = Vp^f$) | $V_p \in$ PLF (p(S))}. For example, Fig. 2(c) shows a transition graph constructed in terms of a set of nodes instantiated by respective grammar symbols from PLF(p(S)). These nodes constitute the set Q. For simplicity, the symbols (q^i , q^f) and (V^i, V^f) are interchangeably used to denote respective PPA states.
- The derivations/reductions implied by PLF (S) are mapped into respective PPA parsing actions as follows:
  - The ε-transitions are mapped into the set {(σ (q_1, ε) = q_j)} $\in$ SPA, for example the ε-transitions $S_0{}^i \xrightarrow{\varepsilon} A_1{}^i$; $b_2{}^f \xrightarrow{r} S_0{}^f$ and $a_{1.1}{}^f \xrightarrow{\varepsilon} B_{1..2}{}^i$ are mapped into σ ($S_0{}^i$, ε) = $A_1{}^i$; σ ($b_2{}^f$ ε) = $S_0{}^f$ and σ ($a_{1.1}{}^f$, ε) = $B_{1..2}{}^i$. Such parsing actions represent the following: ε-transitions from the states instantiated by head grammar symbols to the states of their immediate subordinates; ε-transitions from the states of last subordinates to the states of their respective heads; and ε-transitions from states of grammar symbols to the states of their respective successors (siblings).
  - The derivations of terminal symbols are mapped into the set {(σ (q_1, V) = q_j)} $\in$ SPA of respective move-transitions defined between their corresponding pairs of initial and final

states. For example, the derivation of the symbols a and c are mapped into the move-transitions σ (q_1, a) = $a_{1.1}^{f}$ and σ ($c_{1.2.1}^{i}$, a) = $c_{1..2.1}^{f}$.

- The reduce-transitions $\xrightarrow{r}$ indicate reductions to LHSs of the productions for respective nonterminal symbols. Hence, these transitions are mapped by the set {δ (q, V) = reduce (p)| V $\in$ N, q= V^f and V is LHS (p)} of reduce parsing actions for final states of their respective nonterminals. For example, the reduce-transition $b_2^f \xrightarrow{r} S_0^f$ is mapped as the reduce parsing action δ (S_0^f , S) = reduce(S → A b).

Once PPA has been constructed, a subset construction (ε-closure) is then applied on the resulting automaton to obtain a reduced one (RBA), as shown in Fig.2, where:

- The RBA states are constructed as the set Q = { q_0= (S_0^i, A_1^i , $a_{1.1}^i$), q_1 = ($C_{1.21}^i$, $B_{1.2}^i$, $a_{1.1}^f$), q_2 = (b_2^i, A_1^f , $B_{1.2}^f$, $c_{1.2.1}^f$), q_3 = (S_0^f , b_2^f)} of ε-closures
- The RBA parsing actions are constructed as shift parsing actions defined by the set SPA= {ζ (q_0, a) = q_1, ζ (q_1, c) = q_2, ζ (q_2, b) = q_3} and reduce parsing actions defined by the set RPA = {δ (q_2, B) = reduce (B → c), δ (q_2, A) = reduce (A →aB), δ (q_3, S) = reduce (S →Ab)}.
- Finally, the run of RBA on the input acb proceeds according to the transitions $q_0 \xrightarrow{a} q_1 \xrightarrow{c} q_2 \xrightarrow{b} q_3$, where the reductions B → c, A →aB and S →Ab are performed at the states q_2, q_2 and q_3 respectively.

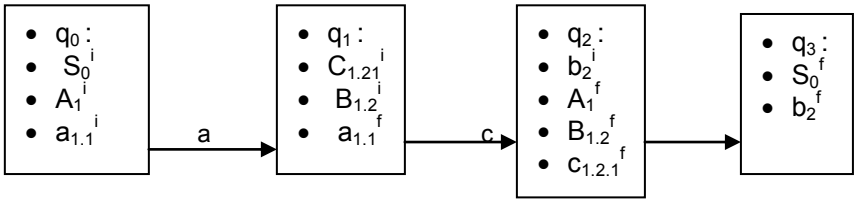


Fig. 2. RBA automaton for the grammar of Example 1

Example 2 demonstrates the proposed parsing approach using a simple context-free grammar. However, such approach handles regular tree grammars and subtle cases such as embedded recursion and grammar productions with different alternatives, as presented in the following sections.

3.1. The production list format (PLF)

Let the 4-tuple (Σ, N, P, S) be a GG grammar, where: (p: A →α) ∈P and α= (V1…Vj…Vn) ∈ (Σ∪N)*. We inductively define the production list format respective to p as PLF (A) = A (PLF (V1)),…, PLF(Vj),…, PLF(Vn)), where:

- Each nonterminal Vj is replaced by the PLF (Vj) defined for its respective production.

- Each ranked terminal Vj is rewritten by its respective PLF (Vj), defined as PLF (Vj) = (PLF (Vj1),…,PLF (Vji),…,PLF (Vjn)), where: Vj (Vj1, Vji,...,Vjn) $\in T(\Sigma \cup N)$.
- PLF(Vj) = Vj, if Vj $\in \Sigma_0$.
- Each grammar symbol in PLF (A) is assigned an index reflecting the position at which it occurs within PLF (A). The index is computed using a function (IND) that is inductively defined as:
  - IND (Vj) = { ε}, if Vj $\in \Sigma_0$.
  - IND (PLF(Vj)) = {0, 1. IND (PLF (V1j)),…,n. IND (PLF (Vnj))}, if Vj $\in$ N and $\exists$ a production (Vj $\rightarrow$ V1j… Vnj) $\in$ P
  - IND (PLF(Vj)) = { ε, 1. IND (PLF (V1j)),…,n. IND (PLF (Vnj))}, if Vj $\in \Sigma$n and Vj(V1j… Vnj) $\in T(\Sigma \cup N)$.

To cover the alternative productions and the productions having embedded recursion, the definition for PLF is extended as follows:

- Let (p: A $\rightarrow\alpha$) be a production having recursion, where α= V1…Ai…Vn. The PLF respective to p is defined as PLF (A) = A_{r*0} (PLF (V1),..., A_{r*i} ,…,PLF (Vn)), where the grammar symbol A is expanded by its definition (production) while its respective recursive-occurrence Ai is rewritten, without further expansion (terminalized). In addition, the head A_{r*0} and the recursive-occurrence A_{r*i} are designated by appending the mark (r*) as a prefix to their respective index.
- Let (p: A $\rightarrow$ { αn}) be a production with alternatives. We represent p as A$\rightarrow$ A(1) | A(2) |…|A(j)|…| A(n), where A(j) $\rightarrow$αj is the alternative (j). The PLF form respective to p is then defined as PLF (A) = A0 ((PLF (A (1)) |….| PLF (A (n))), where the grammar symbols of different alternatives are designated by the number of the alternative to which they belong. However, they are indexed according to their positions in the designated alternative.

PLF in its extended form constitutes a prefix representation of alternative derivation trees with incorporated recursion. Hence, it implies alternative sequences of derivations and reductions, where the embedded recursion is incorporated as recursion-invocation and recursion-termination. The recursion-invocation constitutes an ε-transition from the recursive occurrence to its respective head, while recursion-termination implies an ε-transition from the recursive-head to its respective occurrence. Hence, the recursive occurrence initiates a derivation/reduction path as the one initiated by its respective head and considered as an instance of the original one**.** To distinguish between instances initiated by different recursive occurrences, an instance-identifier (ID) is assigned to each initiated path.

Example 3. Let GG be a regular tree grammar as given in Example1. A PLF form respective to the start symbol R is defined as

PLF (R) = Rr^*_0 (PLF (R (1)) | PLF (R (2)) | PLF (R(3)),where:

PLF (R(1) = Rr^*_0 (1) ($+_1$(1) ($m_{1.1}$(1) ($c_{1.1.1}$(1)), $Rr^*_{1.2}$(1)) ;
PLF (R(2) = Rr^*_0 (2) ($+_1$(2) ($m_{1.1}$(2)($Rr^*_{1.1.1}$(2)), $Rr^*_{1.2}$(2)) and

PLF (R(3) = Rr^*_0 (3)(c_1(3)).

PLF(R) is a prefix representation of three alternative derivation trees with incorporated recursion, as shown in Fig. 3, where **$Rr^*_{1.2}$(1)**, **$Rr^*_{1.1.1}$(2)** and **$Rr^*_{1..2}$(2)** designate such recursion. Consequently, PLF(R) implies three alternative sequences of derivations and reductions as shown in Fig. 4.

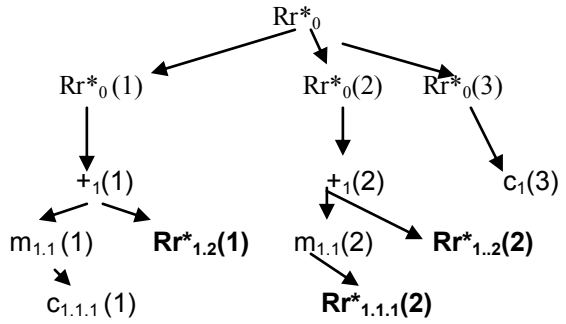


Fig. 3. Derivations trees respective to the grammar of Example 3

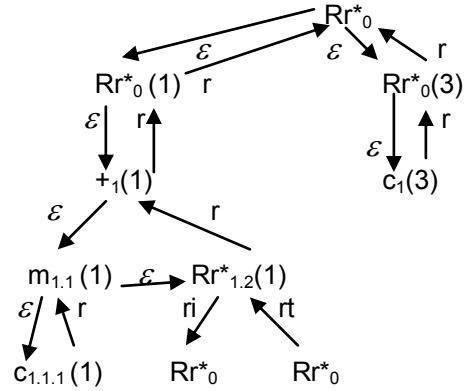


Fig. 4. The derivations/ reductions implied by PLF respective to the grammar (3).

The recursion-invocation is denoted by ri and constitutes the ε-transition from $Rr^*_{1.2}(1) \xrightarrow{ri} Rr^*_0$, while the recursion-termination is denoted by rt and constitutes the ε-transition $Rr^*_0 \xrightarrow{rt} Rr^*_{1.2}(1)$. Hence, the recursive occurrence ($Rr^*_{1.2}(1)$) initiates a derivation/reduction path as the one initiated by its respective head (Rr^*_0). An instance-identifier (ID) is assigned to the initiated path from $Rr^*_{1.2}(1)$ as ID= $Rr^*_{1.2}(1)$, while the original one has ID= ε. The instantiated path ends upon a reduction to the recursive-head, where then recursion-termination is performed. Hence, PLF with incorporated recursion implies initiation and termination of derivations/ reductions paths, as well as respective instances of these paths.

3.2. The position parsing automata (PPA)

PPA is a nondeterministic automaton constructed in terms of a set of states and parsing actions respective to a GG grammar, represented in its expanded list format PLF(S). For example, Fig. 5 shows PPA respective to the grammar of Example 3. As such, PPA is defined as given below based on the following concepts and assumptions for its respective states and parsing actions:

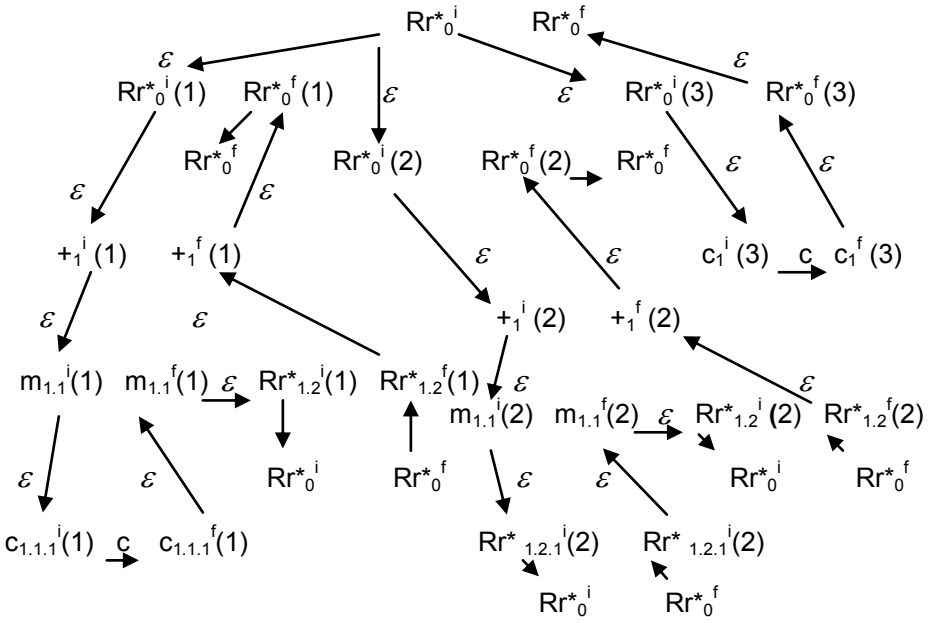


Fig. 5. PPA respective to the grammar of Example 3.

- GG is either instantiated by a context-free grammar or as a regular tree grammar. The set of productions P is generic and is either instantiated with grammar productions or ranked trees (terms). Hence, PPA includes parsing actions respective to the derivations and reductions of ranked trees. The derivations include shifting (reading) the subordinates of their respective ranked terminal. The reductions indicate the completion of such read. Therefore, reading the subordinates of a given ranked terminal are considered as a respective reduction, denoted by coherent-read, for example the string m(c) represents the ranked terminal m. Its respective derivations / reductions as shown in Fig.4 are: $m_{1.1}$ (1) $\xrightarrow{r}$ $c_{1.1.1}$ (1)) $\xrightarrow{r}$ $m_{1.1}$ (1).They include an ε -transition and a reduce-transition which are mapped into the PPA states $m_{1.1}^{i}(1)$, $m_{1.1}^{f}(1)$, $c_{1.1.1}^{i}(1)$ and $c_{1.1.1}^{f}(1)$ as shown in Fig. 5 with the following parsing actions: σ ($m_{1.1}^{i}(1)$, ε) = $c_{1.1.1}^{i}(1) \in SPA, \sigma$ ($c_{1.1.1}^{i}(1)$, c) = $c_{1.1.1}^{f}(1) \in SPA$, σ ($c_{1.1.1}^{f}(1)$, ε) =

$m_{1.1}^{f}(1) \in$ SPA and δ($m_{1.1}^{f}(1)$, m) = reduce (m →m(c)), where the subordinate c of m is read, and a reduction to the nonterminalized m is performed. Such reduction is considered as a coherent read of m(c).

- PLF (S) implies instances of alternative derivations/ reductions, distinguished by respective instance identifiers (ID), as explained in Example 3. Therefore, such PLF is mapped into instances of alternative of PPA states and parsing actions. Such mapping proceeds as given in Example 2, where PPA is formulated as the tuple (T, q_{in}, q_{fin}, Q, SPA, RPA). However, it is extended to include instances of alternative states and parsing actions as follows:
  - The set Q = { ($q_p^i = Vp^i$, $q_p^f = Vp^f$) | $V_p \in$ PLF (S)} of PPA states, as given in Example 1, is extended by the concept of a state-instance to have the form Q = {(q_p^i (alt)(inst) = Vp^i(alt)(inst)), (q_p^f(alt)(inst) = Vp^f(alt)(inst))}, where the index (alt) represents the alternative to which each pair of states (q_p^i, q_p^f) belongs and (inst) is an implicit index to represent different instances of the pair. Initially, the PPA states are created with the implicit index (inst) = ε . At run-time, state-instances are created upon initiation of an instance of a derivation/reduction path. Such creation is indicated by appending the instance-identifier (ID) of the initiated derivation/reduction path atop of the implicit index (inst) of its respective states. For example, let the derivations/ reductions implied by PLF (S) be as given in Fig.4. A possible path initiated by the transition $Rr^*_{1.2}(1) \xrightarrow{ri} Rr^*_0$ is: $Rr^*_0 \xrightarrow{ri} R_{r^*0}(3) \xrightarrow{\varepsilon} c_1$ (3) Originally, such path has respective PPA states $R_{r^*0}^i(3)(\varepsilon)$ and c_1^i (3)(ε) as shown in Fig.5. However, upon the transition $Rr^*_{1.2}(1) \xrightarrow{ri} Rr^*_0$, respective state instances are created as $R_{r^*0}^i(3)((\varepsilon)(Rr^*_{1.2}(1))$ and c_1^i (3)((ε)$Rr^*_{1.2}(1)$). Since recursion-invocation and recursion-termination establish instances of derivation/reduction paths with nested life-time, the (inst) used as an index for PPA states is organized as a stack structure, onto which an ID is pushed upon recursion-invocation and thereafter popped upon recursion-termination.
  - To handle the initiation and termination of instances of PPA states, the specifications of the parsing actions SPA and RPA are extended by semantic actions which are performed at run time as integral parts of the performed transitions and reductions. Thus, SPA and RBA are specified as SPA = {σ (q_1(alt)(inst), V) = q_2(alt)(inst)) :: { semantic-action} and RPA = { δ (q_1(alt)(inst), V) = reduce(p(V)) :: { semantic-action} respectively. The specified semantic-actions constitute program segments defined in accordance with recursion-invocation and recursion-termination as follows:
    - The recursion-invocation is initiated by a transition of the form: σ (q_i(alt)(inst), V) = (q_{rj}(alt)(inst)), where q_i is a state which immediately precedes the one respective to a recursive-occurrence(q_{rj}(alt)(inst)). The ε–transition σ (q_{rj}(alt)(inst), ε) = (q_{r0j}(alt)(inst)) :: { recursion-initiation } is then performed to the

respective recursive-head (q_{r0}(alt)(inst)), where recursion-initiation is a semantic-action defined as: Top(inst(q_{r0})) = ID (q_{rj}) to push the instance-identifier atop of the implied index (inst) of q_{r0} . Further on, instances of PPA states are subsequently created according to the transitions respective to q_{r0}. Such creation is achieved by augmenting each transition σ (q_i(alt)(inst), V) = q_j(alt)(inst)) $\in$ SPA by an implicit semantic-action defined as Top(inst(q_j)) = Top(inst(q_i)) to propagate the instance-identifier from q_i to q_j. Hence any transition σ (q_{r0}(alt)(inst), V) = q_j(alt)(inst)) $\in$ SPA will propagate ID (q_{rj}) and a respective instance of a derivation/reduction path will be established. For example, and considering Fig. 4, the PPA transition which initiates an instance of the path $Rr^*_{1.2}(1) \xrightarrow{ri} Rr^*_0$ is: σ ($Rr^*_{1.2}(1)(\varepsilon), \varepsilon$) = Rr^*_0 (ε) :: { recursion-initiation} $\in$ SPA, where the ε –transition to $R_{r^*0}^{\ i}$ and { recursion-initiation} are performed. As a result, the instance-identifier $Rr^*_{1.2}(1)$ of the initiated path is pushed atop of (ins) respective to Rr^*_0 (ε). Subsequent transitions in accordance with current input are then performed. These transitions create instances of PPA states respective to the initiated derivations/reductions. Assuming an input (c), the subsequent PPA transitions, are: $R_{r^*0}^{\ i}(\varepsilon)(Rr^*_{1.2}(1)) \xrightarrow{\varepsilon} (R_{r^*0}^{\ i}(3)(\varepsilon)(Rr^*_{1.2}(1)) \xrightarrow{\varepsilon} (c_{1.1.1}^{\ i}(3)(\varepsilon)(Rr^*_{1.2}(1)) \xrightarrow{c}$ $c_{1.1.1}^{\ f}(3)(\varepsilon)(Rr^*_{1.2}(1))$,as shown in Fig.5.

- The recursion-termination is established by a transition from a state which immediately precedes the one respective to final state of a recursive-head, where then an ε –transition is performed to the final state respective recursive-occurrence. Hence, the reduce parsing action respective to the final state q of a recursive-head is extended by semantic-action, denoted by recursion-termination to execute a program segment ({IF (Top (ID (q)) $\neq \varepsilon$) { t = Pop(ID(q)); perform ε –transition to t }) that terminates the initiated path and returns the control (ε –transition) to the final state respective to a recursive-occurrence. For example, and assuming an input (c) the above-illustrated path is terminated by the following sequence of PPA parsing actions: $\sigma(c_{1.1.1}^{\ f}(1)\ (\varepsilon)(Rr^*_{1.2}(1)), \varepsilon) = R_{r^*0}^{\ f}(3)(\varepsilon)(Rr^*_{1.2}(1))$, $\sigma(R_{r^*0}^{\ f}(3)(\varepsilon)\ (Rr^*_{1.2}(1), \varepsilon) = R_{r^*0}^{\ f}(\varepsilon)(Rr^*_{1.2}(1))$. Once a transition to $R_{r^*0}^{\ f}$ (3) has taken place, the reduce parsing action $\delta(R_{r^*0}^{\ f}(3), R(3))$ = reduce($R \rightarrow c$)::{recursion-termination} and the augmented semantic-action are performed. As a result, $Rr^*_{1.2}(1)$ is popped from (inst) respective to $R_{r^*0}^{\ f}$ (3), and ε –transition to $Rr^*_{1.2}(1)$ is performed. In addition, (inst) respective to $Rr^*_{1.2}(1)$ is established as ε .
- To handle direct and indirect left recursion, the recursive-heads, the recursive-occurrences and their mutual ε –transitions are designated as cyclic. The cyclic ε –transitions and {recursion-initiation} are not performed. Instead, an instance of the recursive-head is created with respect to the recursive-occurrence as having all the transitions other

than the cyclic ones. In addition to its transitions, a cyclic recursion-termination (CRT) is added to the parsing actions of the final states respective to the recursive- heads. CRT constitutes a default ε – transitions to the final states respective to the recursive-occurrences, as Illustrated later by Example 9.

Based on the above assumptions, the definition of PPA is formalized as follows:

Definition 9. (Position Parsing Automaton). Let (Σ, N, P, S) be a GG grammar and PLF(S) be the PLF form respective to the start grammar symbol S. The 5-tuple PPA (p) = (T, Q, q_{in}, q_{fin}, SA, SPA, RPA, CR) constitutes an extended definition for the position parsing automaton (PPA), where:

1. T = ($\Sigma \cup \varepsilon$) and Σ represent strings generated by the GG grammar.
2. q_{in} and q_{fin} are the initial and final states.
3. Q = $\bigcup_{alt=1}^{n}$ (q_p^i (alt)(ε) = Vp^i(alt)(ε), q_p^f (alt)(ε) = Vp^f(alt)(ε)) is the set the PPA states respective to the individual grammar symbols Vp in n alternatives of PLF(p(S)). Having a stack-structured index (inst), each PPA state constitutes a run-time nested state-instance created by PPA parsing actions in accordance with a dynamically incorporated recursion. Initially the PPA states are created with (inst) = (ε).
4. SA = {recursion-initiation, instance-propagation, recursion-termination} is a set of semantic-actions that are responsible for initiation, creation and termination of instances of PPA states and transitions with respect to embedded recursion. These actions are embedded within the PPA parsing actions, and executed when such actions are applied during parsing.
5. SPA: σ (q_1(Alt)(inst), V) = q_2 (Alt)(inst),) :: {semantic-action} is a move parsing action that specifies the subsequent PPA state $q_2 \in Q$ for a given state $q_1 \in Q$ and a given grammar symbol $V \in T$. In addition, and whenever the transition is applied during parsing, the transition performs instance-propagation as implied semantic action and {semantic-action} ∈ SA as explicit one, if the transition is augmented with the later.
6. RPA: δ (q(Alt)(inst), V) = reduce(r) :: { semantic-action} is a reduce parsing action that performs a reduction rule (r), for every $V \in N$, $q \in Q$ such that q is the respective state to Vp^f and V is the LHS (r). RPA performs the indicated {semantic-action}, If the reduction is associated with such action.
7. CR: λ (q(Alt)(inst),V) = coherent read V (V1…Vn) is a parsing action, defined for every $V \in \Sigma_n$ and $q \in Q$ such that q is the respective state to V^f. It represents the completion of the parsing process for the ranked terminal symbol V and its subordinate symbols (V1…Vn).

Based on the presented definition for the PPA, its construction is reduced to a direct mapping from the PLF (S) using the function: M (PLF(S)) → {PPA.Q, PPA. PPA.SPA, PPA.RPA, PPA.CR}, where M (PLF(S)) performs two major steps. In each step, it scans PLF(S) from left to right and considers the grammar symbols of each alternative of PLF(S) according to their occurrence order. However, the first step constructs the set of PPA states (PPA.Q) as union of the ones respective to the start symbol S_{r*0} and to the grammar symbols of each alternative PLF (P(S(j). Hence, PPA.Q is

constructed as the set { (q_{in} = $S_{r*0}{}^{l}$),(q_{fin} = $S_{r*0}{}^{f}$), $\bigcup_{alt=1}^{n} \bigcup_{i}$ ((q_i^i (alt)(ε) = Vi^i(j)(ID)), (q_i^f (alt)(ε)=Vi^f(alt)(ε))). The second step establishes the PPA parsing actions for the start symbol S_{r*0} and for each grammar symbol Vi (j) in every alternative PLF (S(j)), according to their order and using Definition 9 as a mapping scheme.

3.3. Soundness and generality of the PPA construction

Let the 4-tuple (Σ, N, P, S) be a GG grammar, where $A \to \alpha \in P$ and ($\alpha \in T(\Sigma \cup N)$ or $\alpha \in (\Sigma \cup N)$). The constructed automaton PPA = (T, Q, q_{in}, q_{fin}, SPA, RPA, $CR_{)}$ using the mapping function M (PLF (A), as given in section 3.2, constitutes a two fold generic parsing automaton. First, its parsing behavior simulates tree parsing automata and shift–reduce automata, but with reduced stack activities. Second, it parses hybrid strings drawn from a given type of grammar, augmented by definitions from another type of a grammar, for examples:

- A context-free grammar augmented by constructs from regular tree grammar.
- A regular tree grammar extended by context- free grammar constructs.

The validity of the PPA properties and the soundness of its construction approach are demonstrated by the following lemmas.

Lemma 1. PPA constitutes a bottom up parsing automaton that simulates shift – reduce automata.

- **Proof.** Let GG be instantiated by a context-free G. Let a rightmost derivation be ($S \Rightarrow \beta A w \Rightarrow \beta \eta Y w \Rightarrow \beta \eta c w$). In shift-reduce parser, the reductions are performed according to the right most derivations, but in a reverse order. Thus, the reduction to Y is performed followed by one to A. In our approach, the above rightmost derivation is simulated by the following sequence of transitions: $S^i \xrightarrow{\varepsilon} \ldots \to First(\beta)^i \to \ldots \to Last(\beta)^f \to A^i \xrightarrow{\varepsilon} First(\eta)^l \to \ldots \to Last(\eta)^f \to Y^i \to c^i \to \ldots. \to c^f \to Y^f \to A^f \to First(w)^i \to \ldots \to Last(w)^f$, where Y^f occurs before A^f ,that is, the reduction to Y is performed first. Thus, the reductions performed by our automata are according to the rightmost derivations, but in reverse order. Further more, a handle in a shift-reduce parser is defined as the right side of a production that is formed on the top of the parsing stack [1]. Once, such a handle is formed a reduction is performed. For this purpose, the parser performs the following stack activities: push subsequent terminal (input) symbols, pop parsing stack symbols formed as a handle and push its respective nonterminal. In contrast, the PPA shift activity is defined as ε -transition and move transition on a terminal symbol. A handle is

formed as a result of a sequence of transitions between the initial and the final states of a respective nonterminal. Subsequently, PPA reduction to a respective nonterminal is performed upon the ε-transition from the final state of the last handle's symbol to the final state of the respective nonterminal, for example, the sequence ($Y^i \xrightarrow{\varepsilon} c^i \xrightarrow{c} c^f \xrightarrow{\varepsilon} Y^f$) forms the handle (c) respective to (Y). Upon the ε-transition from c^f to Y^f_{pi}, the reduction to Y is performed. Hence, PPA simulates a shift-reduce automaton. Furthermore, it parses the regular parts of the language as a finite automaton. However, it behaves as a pushdown automaton by using state instantiation with an instance identifier organized as a stack to handle recursion. □

Lemma 2 PPA constitutes a bottom up parsing automaton that simulates the run of regular tree automata.

Proof. Let GG be instantiated by a regular tree grammar, where Σ_0= (b,x,y) Σ_2= c, (S,A)$\in$ N and P= { S $\rightarrow$c(Ab), A$\rightarrow$xy}. let S$\Rightarrow$cAb$\Rightarrow$cxyb be a GG derivation. Let a bottom-up regular tree automaton be defined by the 4-tuple RA = (Σ, Q, δ, Q^f), where: Σ is the input (ranked) alphabet, Q is a set of automaton states, δ = { Q x Σ_j x Q^j | j $\geq$0} is a set of transitions and Q^f is a set of final states. RA constructs S-derivation tree for cxyb as composed of sub trees constructed according to the following order: A(x,y), c(A,b) and S(c). Such construction is obtained a result of the following transitions: (q_x, x), (q_y, y), (q_A, A, q_x, q_y), (q_b, b), (q_c, c, q_A, q_b) and (q_S, S, q_c). On other hand, the run of PPA on cxyb simulates the same construction order, but according to the following sequence of states transitions: $S^i_0 \rightarrow c^i_1 \rightarrow A^i_{1.1} \rightarrow x^i_{1.1.1} \rightarrow x^f_{1.i.1} \rightarrow y^i_{1.1.2} \rightarrow y^f_{1.1.2} \rightarrow A^f_{1.1} \rightarrow b^i_{1.2} \rightarrow b^f_{1,2} \rightarrow c^f_1 \rightarrow S^f_0$, where: the reductions: A$\rightarrow$xy, c$\rightarrow$Ab (coherent read) and S $\rightarrow$c(Ab) are performed at $A^f_{1.1}$, c^f_1 and S^f_0 respectively. Thus, the reduction order represents a rightmost derivation in reverse order and it is equivalent to the bottom–up construction of the S-derivation tree. Furthermore, RA and PPA have the same interpretation of their transitions with respect to the input cxyb.For example,
the RA transitions (q_x, x) and (q_A, A, q_x, q_y) are considered as the mappings x$\rightarrow$ q_x and (A, q_x, q_y) $\rightarrow$ q_A. In contrast, the respective PPA transitions: $x^i_{1.1.1} \xrightarrow{x} x^f_{1.i.1}$ and $A^i_{1.1} \xrightarrow{x} \ldots \xrightarrow{x} A^f_{1.1}$, with implied reduction A$\rightarrow$xy, are considered as the mappings: x$\rightarrow$ $x^f_{1.i.1}$ and (A, $x^f_{1.i.1}$, $y^f_{1.1.2}$) $\rightarrow$ $A^f_{1.1}$ respectively. □

4. The PPA construction algorithm

Given a GG grammar, PPA (S) is then constructed as a direct mapping from its respective PLF(S), using the mapping function M (PLF(S)) as discussed in section 3. However, such function performs two passes (steps) over a fully expanded PLF(S). During the first pass, the PPA states are constructed, while

during second one the PPA parsing actions are constructed. Since PLF(S) is prefix representations of derivation trees respective to GG, its full expansion is equivalent to the construction of such trees. In this section, we propose an alternative but more efficient approach and derive a respective construction algorithm. The proposed approach is a one-pass and based on applying M (PLF(S)) over PLF(S) while it is being expanded in incremental way. For this purpose, we consider a non-expanded form of PLF(S) and a respective but partially constructed PPA as follows:

- Let $S \rightarrow \alpha$, where α = (V1…Vj…Vn) $\in$ ($\Sigma \cup$ N). A non inductive form of PLF(S) is defined as :
- NPLF(S) = S_0 ($V1_{p1}$…Vj_{pj}…Vn_{pn}), where NPLF(S) implies a derivation sub
- tree, denoted PDT (S), with a root labeled by S_0 and children labeled by
- $V1_{p1}$, …,Vj_{pj},…, and Vn_{pn}.
- A partially constructed PPA (PPPA) respective to NPLF(S) is defined as the one that is obtained as a result of applying a modified version of M(NPLF(S)) over PDT(S). Such version is defined as a mapping function M (S_0, Children(PDT(S)) which assumes that the states respective to the root of PDT (S_0) has been created and performs the following:
- Create PPA states respective to the children of PDT($V1_{p1}$…Vj_{pj}…Vn_{pn})
- Establish the parsing actions that cover the transitions between the root and children; the transitions between the children; and the transitions of the states instantiated by recursive occurrences of the grammar symbols.

The construction of PPA is then proceeds according to Algorithm 1 as given below. Assuming an input consisting of the NPLF forms respective to a given GG grammar, Algorithm 1 constructs the PPA(S), using the following data structures and functions:

- The first step of the mapping function is implemented by two functions: CreateNode and CreatePDT to handle the construction of the individual PDTs. The function CreateNode constructs the individual nodes of PDT instantiated with respective grammar symbols. The nodes are indexed by the positions of grammar symbols as they occur within the NPLF forms. The function CreatePDT constructs the PDT respective to a given NPLF form. CreatePDT returns a record of two fields. The first field represents the root of the subtree PDT and the second one represents the children of the PDT as an array of nodes created by the function CreateNode.
- The second step of the mapping function is implemented by two functions: CreateGSPA and by ConstructPPA to handle the construction of the individual PPPA. The function CreateGSPA has a parameter of type Node and returns it's respective initial and final states. Also, CreateGSPA establishes the parsing actions respective to these states, including the ones with recursive occurrences. The function ConsructPPA constructs a PPPA respective to a transmitted PDT as a parameter. ConsructPPA calls the function CreateGSPA to create the states and the parsing actions for the individual nodes of the PDT's children. Then, it establishes the parsing actions that cover the transitions between the root and children; the

transitions between the children; and the transitions of the states, instantiated by recursive occurrences of the grammar symbols.

Algorithm 1

Input: A GG grammar , with start symbol S and its respective NPLF forms

Output : A bottom up automaton PPA = (Q, q_{in}, q_{fin}, SPA, PAR, CR)

Method: An incremental construction of PLF forms, coupled with their gradual transformation into the PPA(S), according to the program given in Fig. 8.

```
Nod0 = Create-node(S, , 0, S); GSPA = CreateGSPA (Node0);
PPA.qin= GSPA.States[1].initial; PPA.qfin = GSPA.States[1].final;
For (each alterntive of NPLF(S))
{PDT = CreateNewPDT(Node0);
 ConstructPPPA( PDT.root, PDT.children)};}
Set–of–Current-PDT = Set–of–Current-PDT ∪ PDT;
While (Set–of –Current-PDT ≠ ε )
{Current-PDT-Constructor = Select-next–PDT (Set–of–Current-PDT);
 Set–of–current-PDT = Set–of–Current-PDT \ Current-PDT-Constructor;
 New-PDT-Roots = (Current-PDT-Constructor).ChildrenLevel
 For i to MaxSize (New-PDT-Roots)
 {New-Root= New-PDT-Roots[i];
  For m = 1 to MaxAltenative (New-Root )
  { Alternative-Node = New-Root [m];
   if AlternativeNode is terminal { }
   Elseif { PDT= CreatePDT(AlternativeNode);
     Set–of–Current-PPT = Set–of–Current-PDT ∪ PDT ;
     ConstructPPPA( PDT.root, PDT.children); } } }
```

Fig. 6. A program for the construction of the PPA automaton

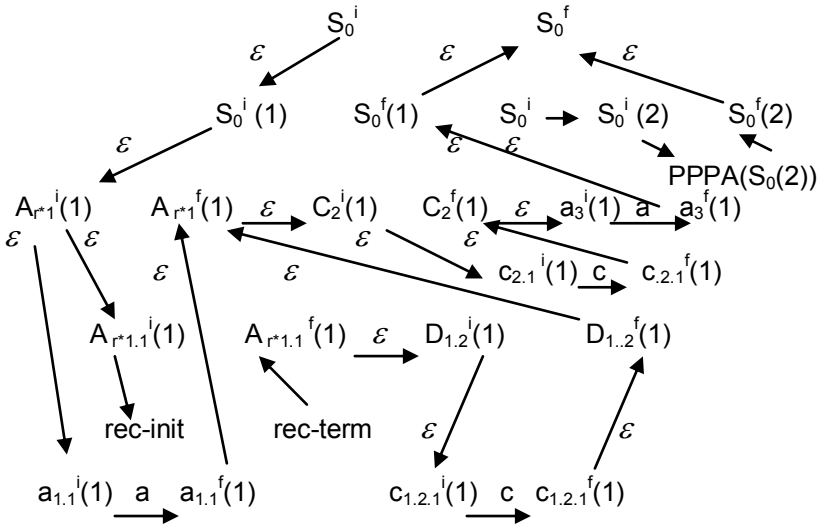


Fig. 7. A partial PPA automaton for the grammar of Example 4

Example 4. Let G = (Σ, N, P, S) be a context free grammar, where: Σ = {a, b}; N= {A, B, C, D}; P= {p1: S→ACa; p2:S→BDb; p3:A→AD; p4:A→a; p5:B→Bc; p6: B→b; p7: C→c p8:D→c. This grammar has direct left recursion and reduce-reduce conflicts. The NPLF forms respective to G are as follows: NPLF(S_0) = S_0 (A_{r*1} (1), C_2(1), a_3(1) | B_{r*1} (1), D_2(1), b_3(1)); NPLF(A_{r*1} (1) = A_{r*0} (A_{r*1} (1) , D_2(1) | a_1(1)); NPLF(C_2 (1) = C_0 (c_1(1); NPLF(B_{r*1} (1)= B_{r*0} (B_{r*1} (1) , C_2(1) | b_1(1)); NPLF(D_2 (1) = D_0 (c_1(1)). Fig.7 shows the transition graph of PPA(S(1)) respective to alternative (1) of NPLF(S_0). Its incremental construction according to algorithm 1 proceeds as follows:

- At steps 1.1 and 1.2, the root PDT is constructed as composed of the single node $Node_0 = S_0$ with ε - transitions to the roots ($Node_1(1) = S_0(1)$, $Node_0(2) = S_0(2)$) of two alternative PDTs. The respective PPPA automaton is constructed in terms of the following:

– The PPA states: PPA.Q = {(PPA.$q_{in} = S_0^i$), (PPA.$q_{fin} = S_0^f$), ($q_0^i(1) = S_0^i(1)$), ($q_0^f(1) = S_0^f(1)$), ($q_0^i(2) = S_0^i(2)$), ($q_0^f(2) = S_0^f(2)$)}

– The PPA parsing action: PPA.SPA = {($\delta(q_{in}, \varepsilon) = q_0^i(1)$),($\delta(q_{in}, \varepsilon) = q_0^i(2)$), ($\delta(q_{in}, \varepsilon) = q_0^i(3)$), ($\delta(q_0^f(1), \varepsilon) = .q_{fin}$), ($\delta(q_0^f(2), \varepsilon) = .q_{fin}$)}

– The PPA parsing action: $\delta(q_0^f(1), S)$= reduce (r1(S)); δ (q_0^f (2), S) = reduce (r2(R).

- At step 2, The PDT respective to the root $Node_0$ (1) is formed as: $Node_1(1) = A_{r*1}$ (1), $Node_2$ (1) = C_2 (1) and $Node_3$ (1) = a (1) respectively. The respective PPPA automaton is constructed as consisting of the following:

– The PPA states: PPA.Q = PPA.Q ∪ { ($q_1^i(1) = A_{r*1}^i(1)$)) ,($q_2^i(2) = C_2^i(1)(2)$) ,($q_3^i(1) = a_3^i(1)$), ($q_1^f(1) = A_{r*1}^f(1)(1)$), ($q_2^f(1) = C_2^f(1)$), ($q_3^f(1) = a_3^f(1)$}.

– The parsing actions: PPA.PAS = PPA.PAS ∪ {(σ ($q_0^i(1)$, ε)= ($q_1^i(1)$) , (σ ($q_1^f(1)$, ε)= ($q_2^i(1)$), (σ ($q_2^f(1)$, ε)= ($q_3^i(1)$), (σ ($q_3^i(1)$,a)= $q_3^f(1)$), (σ ($q_3^f(1)$, ε)= $q_0^f(1)$).

- During the second iteration, the children of the ($Node_1(1)$ and $Node_2(1)$, are considered as roots for which subsequent PDTs and PPPA are constructed .This process is iterated until no further PDT can be constructed. As a result, the construction of PPA(S(1)) automaton respective to to alternative (1) of NPLF(S_0), is completed, as given in Fig.7.
- PPA(S(1)) contains the cyclic transition $A_{r*1}^i(1) \rightarrow A_{r*1.1}^i(1) \rightarrow A_{r*1}^i(1)$. Hence, the transition $A_{r*1}^i(1) \xrightarrow{\varepsilon} A_{r*1.1}^i(1)$ and rec-init are freezed, and rec-term ($A_{r*1}^f(1) \xrightarrow{\varepsilon} A_{r*1.1}^f(1)$) is considered as a cyclic one.

5. The Subset Construction Algorithm

In this section, we propose a subset construction (ε-closure) for PPA (G). It is an extension to the one for nondeterministic finite automata as given in [1].

Such extension is needed to cover a wider class of grammars including regular tree and context free grammars. The subset construction algorithm, denoted by Algorithm 2 is given below. Having the PPA states PPA.Q and their respective parsing actions (PPA.PAS, PPA.PAR, RBA.CR) as an input, Algorithm 2 constructs a reduced bottom-up automata RBA (G), represented by its respective states RBA.Q and a parsing table PAT. The table PAT is organized as matrix of the form: ParsingAction array [$q_o \ldots q_n$, V1…Vn] , where $q_o \ldots q_n \in$ RPA.Q and V1…Vn $\in \sum$(the input alphabet of GG). The individual entries of ParsingAction specify the transitions, the reductions and the semantic actions to be made by RBA(G) during its run on an input alphabet, generated from the grammar G. Algorithm 2 computes the RBA(G) states and their respective parsing actions using an ε-closure function [1]. This function has a parameter of type state and returns set of states, constituting the ε-closure of the transmitted parameter. The function closes the initial states and the final states respective to the different grammar symbol types. However, it does not close the initial states instantiated by grammar symbols of type ranked terminals and the grammar symbols of type recursive instances. The steps of proposed algorithm handle their ε-closures, taking into consideration their peculiarities. It is worth mentioning that the ε-closure for the initial states is equivalent to the kernel item in LR parsing, while the one for final states is equivalent to the complete item.

Algorithm 2

Input: A nondeterministic PPA automaton represented by its respective states (PPA. q_{in}, PPA. q_{fin} , PPA.Q) and parsing actions (PPA.PAS, PPA.PAR and PPA.CR).

Output: A reduced bottom-up automata RBA(G) represented by its respective states (RPA.q_{in}, RPA.q_{fin}, RPA.Q) , parsing actions (RPA.PAS, RBA.PAR , RBA.CR) and by a parsing table PAT.

Method: Apply the subset construction on the states of the PPA(G), according to the following steps:

Step0:

- Initially, apply the ε-closure function on PPA. q_{in} to obtain its respective RPA.q_{in}
- Add PPA.qin to the set of RBA (G) states (RBA.Q), marked as unprocessed one.

Step1:

- Select an unprocessed state from the set (RBA.Q).
- Group the alternative states from which the selected state is composed into two classes. The first one includes the initial states instantiated by recursive instances. The second class includes the initial states respective to terminals and ranked terminals.

Step 2: Perform actions respective to each class as follows

2.1: Actions for the class of type recursive instance

- Create new RBA states for each state(s) in the group, instantiated by their respective recursive instances.

- Add the new states (s) to the set RBA.Q, marked as processed ones.
- Add to the table PAT parsing actions of type "move" from the selected state to the new one (s), augmented with semantic action (recursion-initiation).
- Set the parsing actions for the new state (s), as the ones for the state instantiated by the occurrence of its respective head. Compute ε-closure for the PPA state, instantiated by final symbol respective to the new state (s). Create new RBA state, instantiated by such closure. Add the new states to the set RBA.Q, marked as unprocessed.
- The actions respective to recursive instances designated as cyclic are the same as the above. However, their respective ε–transitions and {recursion-initiation} are designated as cyclic.

Table 1. The RBA parsing table for the grammar of Example 5

State	Input Symbols Parsing actions: Move(M), reduce(R) and semantic action (S)		
	+	m	c
q_0	M(q1, q2)		M(q3)
q1		M(q4)	
q2		M(q5), S(rec-initiation)	
q3	R(r3: R $\rightarrow$ c) S(rec-termination)	R(r3: R $\rightarrow$ c) S(rec-termination)	R(r3: R $\rightarrow$ c) S(rec-termination)
q4	M(q7 q8), c(m(c)) S(rec-initiation)		
q5	M(q1, q2)		q3
q6	M(q10), S(Initial(q11)) c(m(R))		
q8	M(q1, q2)		q3
q9	R(r), c(+(m(c,R)) R(r1: R $\rightarrow$ +(m(c,R)) S(rec-termination)	R(r), c(+(m(c,R)) R(r1:R $\rightarrow$ +(m(c,R))R(r), S(rec-termination)	R(r), c(+(m(c,R)) R(r1: R $\rightarrow$ +(m(c,R))R(r) S(rec-termination)
q10	M(q1, q2)		q3
q11	R(r), c(+(m(c),R)) R(r2: R $\rightarrow$ +(m(R),R)) S(rec-termination)	R(r), c(+(m(c),R)) R(r2: R $\rightarrow$ +(m(R),R)) S(rec-termination)	R(r), c(+(m(c),R)) R(r2: R $\rightarrow$ +(m(R),R)) S(rec-termination)

2.2: Actions for the class of type terminals and ranked terminals

- For each state in the class, select its respective parsing actions of type move (transition) as specified by the parsing table PPA.PAS.

- Compute the ε-closures for each destination state as defined by each selected transition.
- Create new RBA states, instantiated by the ε-closures.
- Add the new states to the set RBA.Q, marked as unprocessed ones.
- Add to the table PAT parsing actions "move" respective to the grammar symbol ,instantiating the state ; the selected state ;and the new ones.
- For each alternative of the new states of type "final" , add to the table PAT the respective parsing actions "Reduce or coherent read" as specified by the PPA parsing actions , including the augmented semantic-action, if any.

Example 5 Applying the subset construction on the PPA automaton of Example 3, will produce the RBA automaton, represented by Table 1 as its respective parsing table.

Example 6 Applying the subset construction on the PPA automaton of Example 4, will produce the RBA automaton, represented by its respective parsing table, given as Table 2.

5.1. The PA- Parser

In this section, we propose a parser, denoted PA-Parser, that simulates in pseudo-parallel the run of RBA (G) automaton on input strings, generated by the grammar G. In addition to the input string, the PA-Parser consults a parsing-table PAT respective to RBA(G), as constructed by the subset construction algorithm. The PA-Parser produces alternative parsing paths which represent a bottom-up construction of derivation trees respective to the input string. During paring, these paths are constructed in terms of performed state transitions and reductions as follows:

- The RBA initial state(q_{in}) is considered as the intial derivation. Hence, a respective derivation/reduction path is created as parsing-path$_{ind}$ = q_{in}, where ind =1 indicates the nesting depth of the path.
- For each alternative (J) of a subsequent transition or a recursion termination (q), a continuation for the current parsing-path$_{ind}$ is created as parsing-path$_{ind.j}$ = parsing-path$_{ind}$ $\cup$ q.

Based on the above- mentioned assumptions, PA-Parser is implemented by Algorithm3.

Algorithm 3 The PA- Parser

Input: An input string and the RBA(G) automaton respective to a grammar G and represented by its respective parsing-table PAT.

Output: Successful and erroneous parsing paths represented as a set of respective state transitions and reductions.

Table 2. The RBA parsing table for the grammar of Example 6

State	Input Symbols Parsing actions: Move(M), reduce(R) and semantic action (S)		
	a	c	b
q_0	M(q2); ε "-tran(q_{1r}^{i}) ; S(rec-initiation)"		M(q4); ε "-tran(q_{1r}^{i}) ; S(rec-initiation)"
q_{1r}^{i}	M(q1); ε "-tran(q_{1r}^{i}) ; S(rec-initiation)"		
q_{2r}^{i}		M(q5)	
q2		M(q7); R(r4: A $\rightarrow$ a) S(rec-termination)"	
q_{3r}^{i}			M(q4); ε "-tran(q_{3r}^{i}) ; S(rec-initiation)"
q_{3r}^{f}		M(q6)	
q4	R(r6: B $\rightarrow$ b) S(rec-ermination)"	M(q8); R(r6: B $\rightarrow$ b) S(rec-termination)"	R(r6: B $\rightarrow$ b) S(rec-termination)"
q5	R(r8: D $\rightarrow$ c); R(r3:A $\rightarrow$ AD); S(rec-termination)"	M(q7); R(r8: D $\rightarrow$ c);R(r3: A $\rightarrow$ AD);S(rec-termination)"	R(r8: D $\rightarrow$ c); R(r3: A $\rightarrow$ AD);S(rec-termination)"
q6	R(r5:B $\rightarrow$ Bc); S(rec-termination)"	M(q8);R(r5:B $\rightarrow$ Bc); S(rec-termination)"	R(r5:B $\rightarrow$ Bc); S(rec-termination)"
q7	M(q9);R(r7: C $\rightarrow$ c));	R(r7:C $\rightarrow$ c));	R(r7: C $\rightarrow$ c));
q8	R(r8: D $\rightarrow$ c)	R(r8: D $\rightarrow$ c)	M(q10);R(r8: D $\rightarrow$ c)
q9	R(r1: S $\rightarrow$ ACa);	R(r1: S $\rightarrow$ ACa))	R(r1: S $\rightarrow$ ACa))
q10	R(r1: S $\rightarrow$ BDb);	R(r1: S $\rightarrow$ BDb);	R(r1: S $\rightarrow$ BDb);

Method:

Initially, the parsing algorithm considers the initial state of RBA (G) as the current parser state (ind, q_{in} (ε)) as well as the initial parsing path. Each state is considered as having its respective instance-identifier initialized to ε , In addition, and as a transition-state, it is associated with an attribute, denoted by ind, to indicate the parsing path to which it belongs. The algorithm then, iteratively, consults the parsing table PAT entry respective to the pair (current state, current input symbol) and performs the following:

- Determine the subsequent parser states , perform the implicit semantic action for the propagation of the state instance-identifiers and create respective parsing paths

- If the consulted parsing table entry specifies a parsing action reduction, the respective reduction is added to the current parsing path .
- If this entry specifies semantic action of type recursion-termination, the computed return state is considered as a continuation that is added to the set of the parser's next states and to respective parsing path.

Finally, upon reaching the end of the input string, the set { parsing-path$_{ind}$} is produced as an output, consisting of parsing paths in which RBA(G) has reached some of its final states and erroneous ones, otherwise.

Example 7. Considering RBA automaton as given in Example 5, the run PA-Parser on the input + (m(c), c) proceeds as shown in Table 3, where two parsing paths are formulated and produced as an output:
parsing-path$_{1,1.xxx}$ = {(q0, ε), (q1, ε), (q4, ε),(q7, ε), (q8, ε 9), CR(m(c), (q3, ε 9),(q9, ε),R(r3: R $\rightarrow$ c), CR(m(R), R(r3: R $\rightarrow$ c),CR(+(m(c),R)), R(r1: R $\rightarrow$ +(m(c,R))} and
parsing-path$_{1,2.xxx}$ ={(q0, ε), (q2, ε),(q5, ε 6), (q3, ε) (q6, ε), (q10, ε 11), R(r3: R $\rightarrow$ c), CR(m(R), (q3, ε 11) (q11, ε), R(r3: R $\rightarrow$ c), CR(+(m(R),R), R(r2: R $\rightarrow$ +(m(R),R) }. These paths are equivalent to a bottom-up construction of their respective derivation trees (Fig. 4)

Table 3. The parse of the input +(m (c),c) by PA-Parser for grammar (5)

Current-parser-state	Parsing behavior		
	Current input	Parsing –action-Move(M) Next parser states	Parsing–actions: Reduce(R),Semantic-action (S), Coherent-read(CR)
(q0, ε)	+	{M(q1, ε), M (q2, ε)}	
(q1, ε) (q2, ε)	m	M(q4, ε) M(q5, ε 6)	 S(rec-intiation (q6))
(q4, ε) (q5, ε 6)	c	M(q7, ε) M(q8, ε 9) M(q3, ε) M(q6, ε) M(q10, ε 11)	CR(m(c), S(rec-intiation (q9)) R(r3: R $\rightarrow$ c), CR(m(R) S(rec-termination(q6)) S(rec-intiation (q11))
(q8, ε 9) (q10, ε 11)	c	M(q3, ε 9) M(q9, ε) M(q3, ε 11) M(q11, ε)	R(r3: R $\rightarrow$ c) S(rec-termination (q9) C(+(m(c,R)) R(r1: R $\rightarrow$ +(m(c,R)) R(r3: R $\rightarrow$ c) S(rec-termination (q11) C(+(m(R),R)) R(r2: R $\rightarrow$ +(m(R),R))

Example 8. Considering RBA automaton as given in Example 6, the run PA-Parser on the input acca proceeds as shown in Table 4, where three parsing paths are formulated and produced as an output:

parsing-path$_{1,1.xxx}$ = {(q0, ε), (q2, ε), R(r4:A $\rightarrow$ a), (q7, ε), R(r7: C $\rightarrow$ c) Error}

parsing-path$_{1,2.xxx}$ = {(q0, ε 1r), (q2, ε 1r), R(r4:A $\rightarrow$ a),(q1r, ε 1r) ,(q5, ε 1r), R(r8:D $\rightarrow$ c),R(r3:A $\rightarrow$ AD),(q7, ε 1r), R(r7: C $\rightarrow$ c), (q9, ε 1r), R(r1: S $\rightarrow$ ACa)}

parsing-path$_{1,2,2.xxx}$={(q0, ε 1r),(q2, ε 1r),R(r4:A $\rightarrow$ a),(q1r, ε 1r),(q5, ε 1r), R(r8:D $\rightarrow$ c), R(r3: A $\rightarrow$ AD), (q1r, ε 1r),(q5, ε 1r), R(r8:D $\rightarrow$ c),R(r3:A $\rightarrow$ AD), (q1r, ε 1r), error}

Among these parsing paths, parsing-path$_{1,2.xxx}$ constitutes a successful one.

Table 4. The parse of input acca by PA-Parser for the grammar of Example 8

state	Parsing behavior		
	input	Parsing–action-Move Next parser states	Parsing –action- Reduce Semantic-action
(q0, ε) (q0, ε 1r)	a	M(q2, ε) M (q2, ε 1r) M (q1r, ε 1r)	R(r4: A $\rightarrow$ a) R(r4:A $\rightarrow$ a);S(rec-termination(q1r))"
(q2, ε) (q1r, ε 1r)	c	M (q7, ε) M (q5, ε 1r)) M (q5, ε 1r)) M (q1r, ε 1r)	R(r7: C $\rightarrow$ c), R(r8:D $\rightarrow$ c), R(r3: A $\rightarrow$ AD), R(r8:D $\rightarrow$ c), R(r3: A $\rightarrow$ AD),S(rec-termination (q1r))"
(q7, ε) (q5, ε 1r) (q1r, ε 1r)	c	Error M(q7, ε 1r) M (q5, ε 1r)) M (q1r, ε 1r)	 R(r7: C $\rightarrow$ c), R(r8:D $\rightarrow$ c), R(r3: A $\rightarrow$ AD),S(rec-termination (q1r))"
(q7, ε 1r) (q1r, ε 1r)	a	M(q9, ε 1r)	R(r1: SAC $\rightarrow$ a), R(r8:D $\rightarrow$ c), R(r3: A $\rightarrow$ AD),S(rec-termination (q1r))"

Example 9. Let G = (Σ, N, P, S) be a grammar with left and embedded recursion. Where (a,+) $\in$ Σ, (E,F) $\in$ N and P= { E $\rightarrow$ E+F, E $\rightarrow$ F, F $\rightarrow$ a, F $\rightarrow$ (E)). Applying the subset construction Algorithm 2 on the PPA automaton respective to the grammar G, will produce the RBA automaton, represented by 12 states and their respective parsing actions. The PA- Parser

behavior on the input a+((a+a)) formulates the following parsing path as an output.

parsing-path$_{1,1.xxx}$ = {(q0, ε), (q1, ε 1), (q6, ε 1), R(r3: F $\rightarrow$ a), R(r2: F $\rightarrow$ E, (q2, ε) (q7, ε), (q8, ε), (q9, ε 3.1) (q3, ε 3.1 1.2) (q6, ε 3.1 1.2) R(r3: F $\rightarrow$ a), R(r2: F $\rightarrow$ E) (q2, ε 3.1 1.2), (q7, ε 3.1 1.2) (q10, ε 3.1 1.2), R(r3: F $\rightarrow$ a), R(r1: E $\rightarrow$ E+F), (q5, ε 3.1), (q12, ε 3.1), R(r4: F $\rightarrow$ (E)), (q1o, ε), (q13, ε), R(r4: F $\rightarrow$ (E)), R(r1: E $\rightarrow$ E+F)}.

6. Discussion

The experiments and the analysis of the derived algorithms for the proposed generic parser have shown the following:

1. The algorithms are characterized by the following calculated complexity:

   - The PPA(G) construction algorithm (Algorithm 1) produces O(2G) states and O(2G+(G+1)+G)) transitions, where G = $\sum_{i=1}^{n} |\, pi \,|$ is the sum of the length(|*pi*|) of the individual productions. The construction time is O(L*(|N+Σ n|*MAX(|*pi*|, i=1,…n))+ (MAX(|*pi*|, i=1,…n)+1))* ALD $\leq C * G$, where C is constant reflecting the levels (L) of the grammar's derivation tree and the number of alternative definitions (ALD).
   - The PPA (G) subset construction algorithm (Algorithm 2) has a runtime O ((|Σ +Σ n +Nr|* |ε - Transitions (Σ +Σ n +Nr)| $\leq$ O (G^2)* s, where s is the number of the PAA reduced states.
   - The parsing algorithm requires a time
     O (| shift-transitions| +|reduce-transitions|)*|input pattern|.
   - The size of the parsing table is O (|Σ | * |s|).

   To illustrate the above calculated complexity, we consider the grammar of Example 5, the construction of its respective PPA (Fig. 5) and RBA (Table 1) have the following characteristics:

- The size of the grammar G = 12; ALD =3; L=3 ; s = 12; | shift-transitions| = 19; |reduce-transitions|= 18.
- Number of PPA states and transitions (ε-transitions, move, reductions, coherent read) = 61.
- PPA construction (Algorithm 1) time is characterized by O(60) $\leq$ 6* 12, where the dominant operations are ConstructPPPA and CreatePDT.
- The subset construction (Algorithm 2) time is characterized by O(384) $\leq$ 1728, where the dominant operations are EmptyClosure and Add(PAT, parsing actions).
- The parsing table is a matrix of 36 elements. The parse of the input string +(m(c),c) is characterized by O(148), where the dominant operations are the access of the parsing table and the computation of the subsequent states.

1. The PA–Parser has a reduced nondeterministic behavior on an input drawn from ambiguous grammar. The parser generates multiple parsing paths. Since the parser is to be used in code selection, such paths are used to select pattern matches subject to minimization criteria. For example, the output of the parser on input +(m(c),c) drawn from the grammar (5) ,as shown in Table 3, represents two pattern matches; +(m(c), R)) and +(m(R),R)).The pattern with the minimum cost is then selected as the one that matches the input. Thus a pattern matcher can be adapted to the behavior of PA-Parser. However, an opposite approach has been suggested in [8], where PPA has been adopted and tightly coupled with the construction of a general pattern matcher.
2. The PA–Parser has a deterministic behavior on input drawn from non-ambiguous context free grammars. This is demonstrated by examples 2 and 9, where only one parsing path is constructed for the given input.

In addition to its generic behavior, a general comparison of PA-Parser with other bottom-up parsing algorithms such as LR, RI [1, 10] has proved that our algorithm is conceptually simpler and requires less states. The simplicity is achieved based on the fact that our approach is tabular and uses a variation of finite automata and its subset construction. Thus, it features their simplicity, as well as their performance with additional overhead due to the embedded semantic actions. However, a particular comparison with similar approaches is as follows:

- LR parsers require that the input grammar is a deterministic [1]. In contrast, the input grammar for PA-parser is generic which can be either instantiated by regular tree or by deterministic and nondeterministic context-free grammars Further more, parsing the same string by both parsers has shown that the PA-parser has less number of moves (shifts) by 20% than the ones for LR(0) as demonstrated by Example 9 , where the LR(0) [1] automaton for the same grammar consists of 12 states, while our parser consists of 15 states. In addition to the absence of parsing stack activities, no goto transitions on nonterminals are used by our parser. Hence, their pre-computation and run time overheads are eliminated.
- GLR parsers [14] cover nondeterministic context-free grammars by using a graph structured stack constructed at run time to represent in pseudo-parallel multiple parse contexts. In contrast, the proposed parser is based on a nondeterministic predictive automaton, the states and the parsing actions of which represents multiple parse contexts in terms of alternative derivation /reduction paths. At run time, these are regenerated in terms of alternative parsing paths (sequence of transitions and reductions) with respect to an input string. Further more, applying our parsing approach on a pathological example ($S \rightarrow SSS \mid SS \mid a$) as given in [14], a considerable reductions in number of states and transitions (number of visited edges) are achieved. The number of the states is fixed, but they are instantiated. Hence, a trade off is made between a space and parsing time, due to states instantiation.
- Reduction incorporated parsers as introduced in [2] and further optimized in [3] are based on constructing a tier (RIA) that is extended to a pushdown automaton by RCA to handle recursion. In contrast our approach uses a nondeterministic automaton that is augmented by semantic actions to

dynamically create instances of RCA states during parsing. Compared to the tier constructed in [2], PPA has less number of states. Also, Its optimization to RBA produces an automaton with the same size as the optimized version of the pushdown automaton as given in [3]. This is demonstrated by Example 9 using the same grammar given in [3]. Our optimization approach is based on a subset construction (ε-closure). Hence, it is more efficient than the heuristic construction steps given in [3].

- Deterministic pushdown automata have been used to recognize regular tree languages as suggested in [9]. However, such use is based on creating context-free grammar that generates a regular language in postfix form. Such a grammar is in Reversed Griebach Normal Form [9]. In contrast, our approach is based on instantiating a generic grammar (GG) by a regular tree grammar that is then mapped into a recognizing automaton. GG is assumed to be a general context-free grammar and no need to transform the input string into a postfix notation.
- Shift-resolve parser [7] is based on a nondeterministic automation which is then determinized using an approach that generalizes similar construction for LR parsers. Using two stacks, it performs reductions with a pushback down to point where reductions should take place. In contrast, our approach generalizes similar construction for deterministic finite automata. The reductions are performed where they should take place using no parsing stack. The parse of the same string by the shift-resolve, as given in [7] and by our approach, as given in Example 8, shows a reduction in parsing-table size as well as in parsing steps.

7. Conclusion

In this paper, we have proposed and implemented a new parsing approach that is characterized by its soundness, generality and efficiency. The parsing approach is based on an extended version of a recently developed position parsing automaton (PPA). The states and the transitions of the PAA are defined based on concepts from the LR (0) items, the finite deterministic automata and a newly introduced concept of the so called state instantiations. The PAA constitutes a nondeterministic bottom–up automaton that is transformed into a reduced one (RBA) in efficient way. Such automaton simulates the parsing behavior of tree automata as well as the shift-reduce automata. Due to their simplified construction principle, the construction overhead for both PPA and RBA is maintained to a minimum. Considering grammars used by similar approaches, both have been shown as powerful parsing models for ambiguous context-free grammar as well as for regular tree grammars. Although, the considered grammars are not as sophisticated as real languages, they are representative ones. Compared to similar approaches, their respective parsing by the proposed one has produced less parser size and fewer shifts-reduce parsing steps. In fact, RBA is a finite automaton that is dynamically extended to incorporate recursion. Such

extension is based on embedded semantic actions to create instances of the RBA states and transitions. Hence, it constitutes an additional overhead during parsing. However, this overhead is reduced due to the instantiation approach. According to such an approach, each RBA state is attached an index and subsequently several state instances can be created and terminated by appending and deleting different instance identifiers atop of the state's attached index. Thus, the space required by state instantiations is minimized and a trade off is made between space, RBA construction and parsing time. As a future work, further experiments well be performed toward achieving more deterministic behavior for the ambiguous grammars at a further reduction of the instantiation cost.

References

1. Aho A.V., Lam M., Sethi R. and Ullman J.D: Compilers Principles, Techniques &Tools, Second edition. Addison Wesley (2007)
2. Ayock J., Horspool R. N.: Faster Generalized LR Parsing. Compiler Construction, LNCS 1575, 32-46 (1999)
3. Ayock J., Horspool R. N., Janousek J., Melichar B.: Even Faster Generalized LR Parsing. Acta Infomatica 37(9), 633-651 (2001)
4. Borchardt B.: Code selection by tree series transducers. LNCS 3317, 57-67 (2005)
5. Cleophas L., Hemerik K. and Zwaan C.: Two related algorithms for root-to frontier tree pattern matching. International Journal of Foundation of Computer Science 17(6),1235-1272 (2006)
6. Ferdenand C., Seidi H., and Wilhelm R.: Tree automata for code selection. Acta Informatica 31(80), 741-760 (1994)
7. Galves J. F., Schmitz S., Farree J.:Shift-Resolve Parsing: Simple, Unbounded Lookahead, Linear Time. Lecture Notes in Computer Science 4094: 253-264 (2006)
8. Jabri S.: Pattern Matching Based on Regular Tree Grammars. International Journal of Electrical
9. Janousek J., Melichar B.: On Regular Tree Language and Deterministic Pushdown Automata. Acta Infomatica 46(7), 533-547 (2009)
10. Johnstone A. and Scott E.: Automatic recursion engineering of reduction incorporated parsers. Science of Computer Programming 68, 95-110 (2007)
11. Katoen J.-P., Nymeyer A.: Pattern-matching algorithm based on terms rewriting systems. Theoretical Computer Science 2378, 237-251 (2000)
12. Madhaven M. et al.: Techniques for Optimal Code Generation. ACM Transaction on Programming Languages and Systems 22(6),972-1000 (2000)
13. Nymeyer A., Katoen J.P.: Code generation based on formal BURS theory and heuristic search. Acta Infomatica 34(8), 597-635 (1997)
14. Scott E. and Johnstone A.: Generalized Bottom- up parsers with reduced stack activity. The Computer Journal 48 (5),565-587 (2005)
15. Shankar P., Ganttati A. , Yuvraj A.R. and Madhaven M.: A new algorithm for linear tree pattern matching. Theoretical Computer Science, 242, 125-142 (2000)

Riad Sadeddeen Jabri. Received the M.E. and Ph.D. in Computer Engineering from Higher Institute for Mechanical and Electrical Engineering/ Bulgaria in 1976 and 1981 respectively. He is currently a Professor and dean of Faculty of Science at Philadelphia University on leave from University of Jordan. His research interests are compilers, formal and programming languages, software systems and networks.

Received: November 9, 2010; Accepted: March 30, 2011.

DOI: 10.2298/CSIS110210045V

Using Lightweight Formal Methods to Model Class and Object Diagrams

Fernando Valles-Barajas

Department of Information Technology, Faculty of Engineering, Universidad Regiomontana
15 de Mayo 567 Pte., C.P. 64000 Colonia Centro, Monterrey, Nuevo León, México
phone: +52 81 8220-4733
fernando.valles@acm.org, fernando.valles@ieee.org

Abstract. In this paper a formal model for class and object diagrams is presented. To make the model the author used Alloy, which is a three-in-one package: a modeling language that constructs software models, a formal method that guides the construction of software models and an analyzer that helps find inconsistencies in software models. In the proposed model the entities that form class and object diagrams, as well as the rules that govern how these elements can be connected, are specified.

Keywords: Alloy, Formal Methods, UML.

1. Introduction

A model is a simplified representation of a system; it is useful for documenting, modeling, communicating and analyzing software systems [31]. One of the mechanisms available for making a software model is Unified Modeling Language (UML), which uses a graphical notation [30]. There are several diagrams in UML, each of these have a specific purpose; for example a class diagram represents the entities of a system as well as the relations between them. A class diagram also represents the attributes of system entities, operations carried out by these entities and the responsibilities assigned to them. There are several integrated development environments (IDEs) that facilitate the process of making UML diagrams; for instance ArgoUML, Borland Together and IBM Rational Rose.

Motivation of the paper: In some of the IDEs mentioned above, a novice user can build an inconsistent UML diagram; for example it is possible to build a class diagram with two classes associated by composition in both directions with Borland Together; in other words the *whole-part* class is also a component of the *part-of* class. In the personal opinion of the author of this paper, the reason for this problem is a weak design of the IDEs mentioned above. Because formal methods have been used to construct models without ambiguity and inconsistencies and to have a strong design [28], in this paper the formal specification for a tool that builds class and object diagrams is proposed. Formal methods use mathematical notations to specify and analyze software models

[17]. The formal method used in this paper is Alloy; which was developed by the Massachusetts Institute of Technology (MIT) software design group. As will be seen, this formal method is also a modeling language that creates software models in a step-by-step style as well as an analyzer that helps to find inconsistencies in software models.

Related works: UML is a modeling language used in software processes (e.g. Rational Unified Process (RUP)) to generate software models. Because of its simplicity, this language has been widely applied in the software industry. One of the disadvantages of UML models is that these artifacts can be interpreted differently by two members of a software team.

Formal methods are useful techniques to formally specify the syntax of UML diagrams. Models obtained using these techniques are unambiguous. For example, in [14] Z language is used to specify the syntax of class diagrams. The author remarks that Z is especially useful to formally specify class diagrams involving recursive structures.

Object-Z is an extension of Z language that includes object-oriented concepts (inheritance, polymorphism and encapsulation among others). This language is used in [20] to specify the syntax of class diagrams. Because UML and Object-Z are based both on object-oriented concepts, the mapping between these two languages is more natural than the mapping between UML and Z. The definition of UML classes in Z and Object-Z language supports this idea. UML classes specify the structure, behavior and associations shared by a set of objects. While in [14], UML classes were modeled using two Z schemes; one for defining class structures and other for defining behavior, in [20] using Object-Z only one container was necessary to define UML classes. Besides of this, an Object-Z element defining classes can inherit variables and operations from one element that has already been defined. Another paper that formalizes UML class diagrams using Object-Z is [21].

In [2] the authors compare several approaches based on Z and Object-Z to model class diagrams and one of the conclusions made is that thanks to the object-oriented concepts of Object-Z, this modeling language produces more concise models than Z language.

Another modeling language that has been used to specify the syntax of UML class diagrams is the Prototype Verification System Specification Language (PVS-SL), see [4]. In [24] the authors present a tool that analyzes the syntax and semantics of Object Constraint Language (OCL) constraints together with UML models and translates them into PVS-SL. The authors tested the proposed tool representing the Sieve of Eratosthenes algorithm, which is an algorithm that finds prime numbers, in a UML class diagram and OCL constraints and then this model was converted to a PVS-SL model.

In the analysis phase, it is mandatory to have a notation that allows software developers to generate requirement models. These models should be done so that good communication between users and the development team is established. As the authors of [15] state, UML has demonstrated to be a good option in developing requirements models, however these models cannot be analyzed

and reasoning based on these models is difficult. To overcome these problems the authors of [15] propose a tool that receives Extensible Markup Language (XML) class diagram specifications as input and generates RAISE Specification Language (RLS) class diagram specification as output. The tool is able to analyze and reason of the original class diagrams. It is important to mention that modeling tools (e.g. ArgoUML) can store class diagrams in an XML format; so the input for the proposed tools is easy to obtain.

OCL is the de facto modeling language for adding constraints to UML diagrams (see [1]). This language was developed because UML constructs alone cannot precisely model situations occurring in real world problems [37].
USE is a UML-based Specification Environment that allows software developers to model UML class diagrams along with OCL constraints ([16]). USE makes it possible to check the consistency of UML models by generating instances of class diagrams.

The authors in [8] argue that OCL has weak semantics and that reasoning from UML diagrams along with OCL constraints is difficult. These authors state that by transforming from OCL to the Isabelle/HOL theorem prover, property analysis of UML diagrams is possible.

Models consist of a dynamic and a static part. UML has a variety of diagrams covering these two parts. For example the dynamic part may be modeled by state machine diagrams and the static part by class diagrams. In [26] the authors present a technique to transform class and state machine diagrams into B modeling language. Models represented in the B notation can be analyzed.

Papers [32], [33], [35], [34], [36] demonstrate that Alloy, which is the modeling language used in this paper, has been successfully used to formally specify UML diagrams. In [32] the author uses Alloy to model use case diagrams. In [33] state machine diagrams are formalized and an extended version of this paper was later presented in [35]. In [34] a model of System Modeling Language (SysML) requirements diagrams using Alloy is included. It is important to mention that SysML is a UML profile.
UML collaborations model interactions between entities working together to accomplish a specific task. In [36], the author formally specify Collaborations.

Other techniques that do not apply formal methods have been used to verify the correctness of UML diagrams. In [19] the authors use constraint programming for this purpose. UML diagrams along with OCL constraints are first specified as a constraint satisfaction problem and then properties of UML class diagrams are verified with the UMLtoCSP tool.

Methods having roots in artificial intelligence have also been used to specify UML diagrams and to later reason from these models. For example description logic, which is used in artificial intelligence to represent knowledge, is used in [6] and [10] with this objective.

Coalgebra is used in computer science to specify the states of a system. This technique is applied in [27] to represent the semantics of class and use case diagrams. Associations among classes are interpreted as coalgebraic ob-

servers. The generalization hierarchy of classes is specified by the inheritance morphism among them.

Graph theory is applied in [22] to model class diagrams as graphs. Class diagrams are mapped as nodes with connecting edges.

Formal methods are based on mathematical logic; for example, Alloy is a modeling language based on set theory, first order logic and relational logic. Equational logic was developed to help in the formal development of programs. A novel method based on equational logic to model class diagrams is presented in [11].

Paper [9] does not apply formal methods to UML class diagrams but it formalizes aggregation and composition relationships. As the authors mention, there are some UML concepts that are not well specified in the Object Management Group (OMG) documentation of UML. Two examples of these poorly specified concepts are the aggregation and composition relationships. The paper models these two relationships as subclasses of a more generic class called the whole-part relationship. In UML 1.x these relationships were modeled as meta-attributes. With the addition in the UML meta-model of one class for each relationship, related characteristics of the aggregation and composition relationships can be attached to theses classes. The following characteristics of these two new classes are considered in the author's proposal: shareability, separability, mutability, configurationality, lifetime dependency and existential dependency.

As the reader may notice, the papers reported so far are based on the transformation of UML class diagrams to a more formal notation to get analyzable and precise models. According with [12], the disadvantage of this approach is that this transformation requires a strong background in mathematics and, unfortunately, most practitioners do not fulfill this requirement. In [12], the author proposes a technique based on the manipulation of UML class diagrams by using some transformation rules. These rules allow to know if one class diagram is a conjecture of another class diagram. The method presented by the author does not require a strong knowledge of mathematics.

In [23] the authors study the application of graph transformation, which a mature field, to describe the semantics of class, object and state diagrams. The state of a system is represented by object diagrams and these are in turn represented by graphs. A change in the system state is described by using two graphs (representing previous and later states) and by transformation rules. The paper does not present analysis or reasoning of class diagrams based on graph transformation.

Structure of the paper: In this section the motivation of the paper has been given. The following section contains some basic concepts of Alloy; which is the formal method that will be used to model class and object diagrams. Section 3 introduces the necessary concepts of class and object diagrams to understand the model proposed in the paper. Section 4 contains the formal model. In section 5 a comparison of Alloy with other formal methods is presented. Concluding remarks are given in section 6.

2. A brief introduction to Alloy

Alloy is a modeling language that has three mathematical tools for making software models: first order logic, relational calculus and set theory [18]. A system is modeled by representing each system entity and the relations between them. The entities are represented as signatures, which are similar to classes, and the relations as fields of the signatures. Constraints between entities can be specified near signatures as signature facts or outside signatures as facts. Functions are used to specify reusing expressions. Basic constraints can be specified in the relations between signatures by using multiplicity keywords. Alloy has the following multiplicity keywords: **one** (exactly one), **lone** (zero or one), **some** (one or more) and **set** (any number). Properties of the model elements can be expressed by using predicates. Alloy has several relational calculus operators that allow to construct complex expressions; for instance the $\wedge$ and $*$ respectively denote the transitive closure and the reflexive-transitive closure of a relation and they are useful for constructing expressions in a relational calculus style. This modeling language also has operators to represent expressions using first order logic; e.g. implication ($\Rightarrow$), not (!), and ($\&\&$), or ($||$) and the bi-implication ($\Leftrightarrow$). The operators for manipulating sets are: in, union ($+$), intersection ($\&$), difference ($-$) and equality ($=$). By using all these operators a modeler can specify software models without ambiguity.

A model in Alloy captures not only the static view of the system but also its dynamic view. It is possible to specify operations that affect the state of the system by using predicates. When a predicate is used to model an operation, the pre-conditions and the post-conditions must be specified.

An attractive characteristic of Alloy is that it does not need a tool to type the formal specification; as will be seen this is written using American Standard Code for Information Interchange (ASCII) characters.

3. Class diagrams

A class diagram is one of the static diagrams of UML [5]. It is used to represent the entities of a system and the relations between them. A more detailed class diagram can include the features of the entities as well as their responsibilities. There are two types of features: structural and dynamic [13]. Structural features can be subdivided in attributes and associations. Attributes correspond to variables in programming languages. Due to the fact that the associations between classes are represented as variables in programming languages, these are also considered to be structural features. The dynamic part of the classes are the operations, which are implemented by methods in a programming language. There are five types of relations between classes: association, aggregation, composition, generalization and dependency.

An association represents a relation between objects of the same level [29]; this is represented using a solid line connecting two related classes.

Aggregation and composition are types of associations [31]. They are useful for representing an entity that is composed of smaller entities. These relations

can be logical or physical. The classes that participate in these kinds of associations belong to different abstraction levels; one is the *whole-part* and the other is the *part-of*. Instead of using an association labeled with *whole-part* and *part-of* strings, the solid line that joins two related classes is adorned with a diamond near the *whole-part*. Three important characteristics of the composition relation are:

- if the *whole-part* is destroyed the smaller parts are also destroyed.
- a *part-of* may be a part of only one composite at any time [7].
- the diamond near the *whole-part* is filled (in an aggregation relation this diamond is empty).

Generalization is the process of finding the common features of some classes and then putting these common features into a class. This process is useful to inherit common features from one class to others. The class that passes features is called superclass and the classes that receive the features are called subclasses.

Generalization is also a binary relation between a superclass and a subclass. The generalization relationship is represented as a solid line with a hollow arrowhead at the superclass end. In a class diagram several generalizations are usually arranged forming an inheritance hierarchy; this should not be either too deep or too wide [7]. The set of classes above a class c in the inheritance hierarchy are ancestors of this class and they are obtained by using the transitive closure operator. The set of classes below a class c in the inheritance hierarchy are descendants of this class and they also are obtained by using the transitive closure operator [31]. When an instance is generated from a class c, no matter how many ancestors are in this class, only one object is generated with the features of class c and the ancestors of this class [3]. A class inherits the features of its ancestors as well as their relations [25]. Classes in an inheritance hierarchy can be divided into two types: abstract and concrete. It is not possible to generate an instance from an abstract class. Because of this when a superclass is abstract then the subclasses of this class form a partition [25].

The dependency relation represents the situation in which a change in one element affects other elements. An example of the dependency relation is when a class sends a message to other class; if the receiver of the message changes its interface, then the sender must be changed so that the two classes are still able to communicate. Other examples of dependencies are when a class c_1 has a class c_2 as a parameter or as a local variable for one of its operations or when a class c_2 is in the global scope of a class c_1. According with [13] the dependency relation is anti-transitive. This is formally defined as: $\forall c_1, c_2, c_3 : C\ (c_1 R c_2\ \&\ c_2 R c_3) \Rightarrow not\ (c_1 R c_3)$.

4. A formal model for object and class diagrams

In this section a formal model for class and object diagrams is given. The model was divided in five parts: in the first part an initial model is presented, the second

part specifies the association and dependency relations, the third part presents the generalization relation. The four part specifies the aggregation and composition relation. The last part formalizes the operation definition.

Convention used in this paper: In the models presented in this section, for the purpose of clarity, the keywords of the modeling language are printed in **bold font**.

4.1. The initial model

Fig. 1 presents an initial model for class and object diagrams. Line 1 specifies the name of the model (classModel) and the directory for this model (umlMetamodels). Next, in lines 2 and 3, some Alloy libraries are included in the model. Alloy has some built-in libraries that can be used to save time; in this case the model is using two libraries; one to manipulate booleans and another to manipulate relations. Line 6 defines a signature for class diagrams. As was mentioned in section 2 the relations between entities (represented in the model as signatures) are specified using fields of signatures; for instance, the *classes* field of the signature *ClassDiagram* denotes a binary relation between the *ClassDiagram* signature and the *Class* signature. Multiplicity keywords can be used in a relation to constrain the number of participants; for example, the multiplicity keyword **some** in the *classes* field means that a class diagram has one or more classes. Multiplicity keywords can be used also to constraint the instances that can be generated from one signature; the keyword **one** in the *ClassDiagram* signature indicates that only one instance from this signature can be generated.

Some ternary relations are defined in the *ClassDiagram* signature; for example the *association* field is a ternary relation of the form:
association: ClassDiagram$\rightarrow$classes$\rightarrow$classes. With the definition of this relation, a class diagram knows all classes that are related by the association relation. A model can be documented in Alloy using comments; for example in line 12 a comment was defined (a comment in Alloy begins with a double-dash). The last line of fig. 1 is the **run** command; by using this command the modeler can generate instances of the model to see if there are inconsistencies.

It is important to mention that in the initial model, the inheritance relation was not defined inside the object diagram signature; the reason for this is to remark the idea that the inheritance relation is really a relation between classes (see section 3).

4.2. Association and dependency relations

Fig. 2 contains the part of the model where association and dependency relations are specified. In line 1 one function, which is a reusing expression, is defined. The function *validRelation* checks if two objects o_1 and o_2 can be related using some specific relation. In line 5 the predicate *antiTransitive* is defined to constrain the dependency relation. As was explained in the previous section, Alloy has a module to manipulate relations, but the anti-transitive property is

```
1  module umlMetamodels/classModel
2  open util/boolean as booleans
3  open util/relation as relations

4  sig Name{}

5  sig Class{ name: Name }

6  one sig ClassDiagram{
7    classes: some Class,
8    association, dependency, inheritance, aggregation,
9      composition: classes->classes
10   }

11 sig Object{
12   -- models anonymous objects
13   name: lone Name,

14   type: Class
15   }

16 one sig ObjectDiagram{
17   objects: some Object,
18   association, dependency, aggregation,
19     composition: objects->objects
20   }

21 showInstance: run{} for 7
```

Fig. 1. Initial model

not defined in that module. In signatures *ClassDiagram* and *ObjectDiagram* the dependency relation was constrained to be anti-transitive.

4.3. Generalization relation

Figs. 3, 4 and 5 present the part of the model where the generalization relation is specified. In line 1 of fig. 3 the signature *GeneralizationStructure* was defined; this signature represents an inheritance hierarchy. The field *setName* is used to document the criterion that was used to form a classification; in early versions of UML this concept was called discriminator. Some constraints over the generalization structure were defined in line 6. The *isComplete* field is used to indicate that all subclasses of class c have been defined. The *isDisjoint* field implies that inheritance of a class c from two subclasses of a disjoint generalization structure is not allowed. The *isOverlapping* field is the opposite of the *isDisjoint* field. Two signature facts were defined in the signature *GeneralizationStructure*. Line 9 specifies that superclasses and subclasses defined in the *GeneralizationStructure* signature must belong to the *ClassDiagram* signature. Line 10 declares that there must be a mapping, in the *ClassDiagram* signature, between the superclasses and the subclasses defined in the *GeneralizationStructure* signature.

```
1  fun validRelation[ r: univ->univ ]:Object->Object{
2    { o1, o2: Object |
3      o1.type->o2.type in r.Class->Class.r }
4    }

5  pred antiTransitive[r: univ->univ, S: set univ]{
6    all s1, s2, s3: S |
7      (s1->s2 + s2->s3) in r => s1->s3 not in r
8    }

9  one sig ClassDiagram{

10   -- as before }
11   {
12   antiTransitive[dependency, classes]
13   }

14 one sig ObjectDiagram{

15   -- as before }
16   {
17   association in
18     validRelation[ClassDiagram.association]

19   dependency in
20     validRelation[ClassDiagram.dependency]

21   antiTransitive[dependency, objects]
22   }
```

Fig. 2. The association and dependency relations

As can be seen in fig. 3, a class is composed by structural and dynamic features. The static features defined in *Class* signatures were properties; associations were defined in the *ClassDiagram* signature as relations (see fig. 1). The *isSubstitutable* boolean field is used to indicate if a subclass is used to substitute one of its superclasses. A class that does not stand alone and is used to add behavior and structure to other classes is called a mixin class; it is used in multiple inheritance relationship [31].

Fig. 4 presents part II of the generalization relation. The properties for the generalization relation are defined in the *ClassDiagram* signature: acyclic, irreflexive and antisymmetric. Two constraints about the form of the inheritance hierarchy were defined: an inheritance hierarchy should not be too deep or too wide. Line 14 constrains the model to only single inheritance. This line was written as a comment because most programming languages do accept multiple inheritance. Line 19 of fig. 4 defines the *isDisjoint* property defined in fig. 3. In line 26 a signature fact was defined to specify that a class not only inherits the features of its ancestors but also their associations. In fig. 4 a predicate was defined to model the *isComplete* property of the generalization relation; in this predicate the precondition and post condition were defined.

The last part of the generalization relation is shown in fig. 5. This figure defines the concept of classification. An object usually belongs to a class but

```
1  sig GeneralizationStructure{
2    superClass: Class,
3    subClasses: some Class,

4    -- set names are also known as discriminators
5    setName: Name,
6    isIncomplete, isDisjoint, isOverlapping: Bool}
7    {
8    let cd = ClassDiagram{
9      (superClass + subClasses) in cd.classes

10     some c: (cd.inheritance).(cd.classes) |
11       (c = superClass) &&
12       (subClasses = cd.inheritance[c])
13     }
14   }

15 abstract sig Feature{}
16 sig Property, Operation extends Feature{}

17 sig Class{
18   -- as before

19   features: some Feature,

20   isSubstitutable,
21   isAMixin,

22   isRoot, isLeaf, isAbstract: Bool}
23   {
24   let cdi = ClassDiagram.inheritance{
25     --  a root class that has ancestors
26     -- is not permitted
27     no cdi.this => isRoot = True
28       else isRoot = False

29     --  a leaf class that has descendants
30     -- is not allowed
31     no this.cdi => isLeaf = True
32       else isLeaf = False

33     -- an abstract class must have descendants
34     -- and it can not be a leaf class
35     some (this & cdi.this) && (isLeaf != True)
36       => isAbstract = True
37     else
38       isAbstract = False
39     }

40   isAMixin = True => isAbstract = True
41   }
```

Fig. 3. The generalization relation; part I

there is no restriction indicating that an object cannot belong to more than one class. The field *classification* stores the possible classes to which an object can belong. Line 16 models the fact that an object contains the features of its type (class) and ancestors of this type.

```
one sig ClassDiagram{

  -- as before }
  {
  -- as before

  -- constraints on the generalization relation --

  acyclic[inheritance, classes]
  irreflexive[inheritance]
  antisymmetric[inheritance]

  -- controls the width of the inheritance hierarchy
  all c: inheritance.classes | #(c.inheritance) =< 6

  -- controls the depth of inheritance hierarchy
  all c: classes | #(c.^inheritance) =< 5

  -- avoids multiple inheritance
  -- all c: classes.inheritance | one inheritance.c

  all c: inheritance.classes |
    some ge: GeneralizationStructure |
      c = ge.superClass

  -- models the disjoint property of generalization
  let i = inheritance |
    all c1: i.classes | #(c1.i) > 1 &&
      (superClass.c1).isDisjoint in True =>
        no c2: classes.i |
          #(i.c2) > 1 && i.c2 in c1.i

  --  a class inheritances the associations of
  -- its ancestors
  all c: classes | all a: getAncestors[c] |
    c->a.association in association
  }

-- models the isComplete property of generalization
pred addDescendant[cd, cd': ClassDiagram, f, s: Class]{
  -- pre conditions
  cd != cd'
  f in cd.classes
  some ge: GeneralizationStructure |
    ge.superClass = f && ge.isIncomplete = True
  f->s not in cd.inheritance

  -- post conditions
  cd'.inheritance = cd.inheritance + f->s
  }

fun getMultipleInheritance[ cd: ClassDiagram ]:
  set Class{
  { c: (cd.classes).(cd.inheritance) |
    #(cd.inheritance).c > 1  }
  }
```

Fig. 4. The generalization relation; part II

4.4. Aggregation and composition relation

The last relations that were defined were the aggregation and composition relations. Both relations have the following properties: acyclic, irreflexive and an-

```
sig Object{
  -- as before

  features: some Feature,

  classification: set Class,
  multipleClassification: Bool}
  {
  type in ClassDiagram.classes

  -- models multiple classification
  classification in
    getGeneralizationStructure[ type ].subClasses
  #classification > 1 =>
    multipleClassification in True

  --  an object contains the features of
  -- its type (class) and the ancestors of
  -- this type
  let i = ClassDiagram.inheritance |
    -- * is the reflexive transitive closure operator
    features =
      { f: Feature | f in *i.type.features }
  }

fun getAncestors[c: Class]: set Class{
  -- ^ is the transitive closure operator
  { x: Class | x in ^(ClassDiagram.inheritance).c }
  }

fun getDescendants[c: Class]: set Class{
  { x: Class | x in c.^(ClassDiagram.inheritance) }
  }

fun getGeneralizationStructure[ c: Class ]:
  set GeneralizationStructure{
  { ge: GeneralizationStructure | ge.superClass = c }
  }

pred mutateClassification[ o: Object, c: set Class ]{
  o.classification = c
  }
```

Fig. 5. The generalization relation; part III

tisymmetric. Two operations were defined related to these relations. The *deleteObject* predicate models the fact that an object that is *part-of* another object in a composition relation cannot exist outside the *whole-part*.

4.5. Operations

Fig. 7 contains the definition of operations. This model shows how Alloy deals with sequences. As the reader may notice, an operation is composed of a sequence of parameters (see the Operation signature). The specification of an operation could have been defined as a set of parameters, but this would have been incorrect because the parameter positions within an operation do matter.

```
1  one sig ClassDiagram{

2    -- as before }
3    {
4    -- as before

5    acyclic[aggregation, classes]
6    irreflexive[aggregation]
7    antisymmetric[aggregation]

8    acyclic[composition, classes]
9    irreflexive[composition]
10   antisymmetric[composition]
11   }

12 one sig ObjectDiagram{

13   -- as before }
14   {
15   -- as before

16   composition in
17     validRelation[ClassDiagram.composition]
18   aggregation in
19    validRelation[ClassDiagram.aggregation]

20   acyclic[aggregation, objects]
21   irreflexive[aggregation]
22   antisymmetric[aggregation]

23   acyclic[composition, objects]
24   irreflexive[composition]
25   antisymmetric[composition]
26   }

27 pred delObject[ od, od': ObjectDiagram, o: Object]{
28   -- pre conditions
29   o in od.objects
30   not o in (od.aggregation).(od.objects)

31   -- post conditions
32   od'.objects = od.objects - o
33   }

34 --   this operation takes into account the
35 -- composition relation
36 pred addPart[ od, od': ObjectDiagram, w, p: Object ]{
37   -- pre conditions
38   (w + p) in od.objects
39   w->p in validRelation[ClassDiagram.composition]
40   not w->p in od.composition
41   -- no sharing
42   not p in od.composition[Object]

43   -- post conditions
44   od'.composition = od.composition + w->p
45   }
```

Fig. 6. The aggregation and composition relation

```
1  open util/boolean as booleans

2  enum VisibilityKind{ Private, Public, Protected }

3  sig Name{}

4  sig Type{}

5  sig Parameter{
6    name: Name,
7    type: Type
8    }

9  sig Operation{
10   name: Name,
11   visibilityKind: VisibilityKind,
12   parameters: seq Parameter,
13   isAbstract: Bool }
14   {
15     -- parameter names should be unique
16     all disj i, j: #parameters | no (parameters[i].name & parameters[j].name)
17   }


18 sig Class{
19   ops: some Operation,
20   isAbstract: Bool }
21   {
22     -- operations should have different signatures
23     all disj op1, op2: ops | op1.name = op2.name
24       && #(op1.parameters) = #(op2.parameters) => all i: #op1.parameters |
25         op1.parameters[i].type = op2.parameters[i].type => op1 = op2

26     -- if a class has an abstract operation, it must be abstract
27     ops.isAbstract = True => isAbstract = True

28     -- abstract classes have at least one abstract operation
29     isAbstract = True => some o: ops | o.isAbstract = True
30   }

31 fact{
32   -- classes should have different operations
33   all disj c1, c2: Class | no (c1.ops & c2.ops)
34   }

35 showInstance: run{} for 4
```

Fig. 7. Definition of operations

5. Comparison of Alloy with other formal methods

A valid question the reader might have is: what are the advantages of Alloy in comparison to other formal methods? To answer this question let us analyze an excerpt of the meta-model, which is presented in fig. 8. In this fig., the inheritance relationship between classes and objects is emphasized.

One of the most important characteristics of Alloy is that it is possible to get an instance of the model without actually writing any single line of code. Fig. 9 shows an instance of the meta-model except (the name of the inheritance rela-

tion was changed to *isATypeOf*). The reader may notice that this fig. presents some problems, one of these is that the instance of the object diagram does not correspond to the definition of the class diagram instance; the class diagram instance states that Class3 is a type of Class1 (Class3 is a subclass of Class1); however in the object diagram instance Object2, which is an instance of Class3, inherits from Object3, which an instance of Class2. In order for the object diagram instance to correspond to the class diagram instance, the type of Object3 should be Class1.

To fix this problem, lines 5 and 6 of fig. 10 were added to the model. After these lines were added another instance was generated; see fig. 11. The previous problem was solved but the model still has some inconsistencies, e.g. some classes and objects inherit to themselves (Class1, Class3 and Object3). Lines 8 and 15 of fig. 10 solved this problem. After these lines were added to the model, no flaws were found (see fig. 12).

Even thought the last instance does not present the error that class C_1 inherits from class C_2 and at the same time class C_2 inherits from class C_1, it is possible using the meta-model to generate an instance having this inconsistency. Lines 8 and 15 were added to avoid this possible problem.

```
1  module umlMetamodels/classModel

2  sig Name{}

3  sig Class{ name: Name }

4  one sig ClassDiagram{
5    classes: some Class,
6    inheritance : classes->classes
7    }

8  sig Object{
9    name: lone Name,
10   type: Class
11   }

12 one sig ObjectDiagram{
13   objects:  some Object,
14   inheritance: objects->objects
15   }
```

Fig. 8. An excerpt of the meta-model

6. Conclusions

In this paper a formal specification for a tool that builds class and object diagrams was given. As can be seen in the model of this paper, Alloy uses propositional logic, relational calculus and set theory to build software models. Even

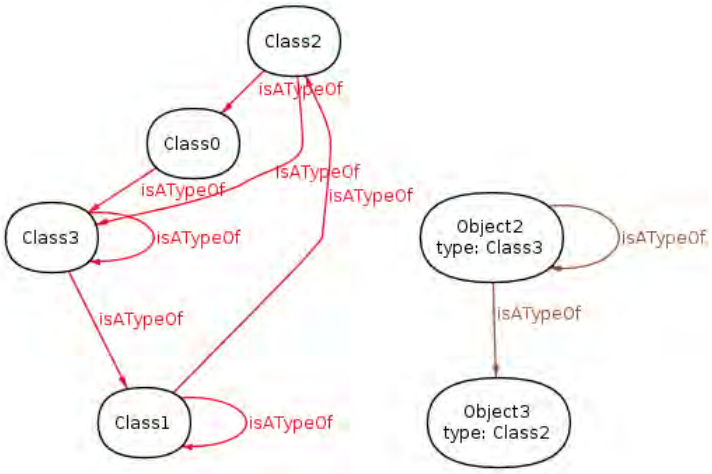


Fig. 9. Initial instance

```
1  one sig ObjectDiagram{
2    -- as before }
3    {
4     -- valid relation
5     all o1, o2: objects | o1->o2 in inheritance =>
6      o1.type->o2.type in ClassDiagram.inheritance
7    }

8  pred noSymmetric[r: univ->univ]{ no ~r & r }
9  pred noReflexive[r: univ->univ]{ no iden & r }
10 pred acyclic[r: univ->univ]{ no iden &  ~r }

11 fact{
12   let cdi = ClassDiagram.inheritance | {
13     noReflexive[cdi]
14     acyclic[cdi]
15     noSymmetric[cdi]
16     }
17   }
```

Fig. 10. Corrections for the excerpt of the meta-model

thought Alloy does not have recursive functions, the transitivity closure can be used to iterate over a relation. The author of this paper believes that the simple but powerful notation of Alloy makes it possible to build software models without ambiguities. The author also believes that the specification obtained from the application of Alloy, will be a great requirements document for the implementation of an object and class diagram modeling tool.

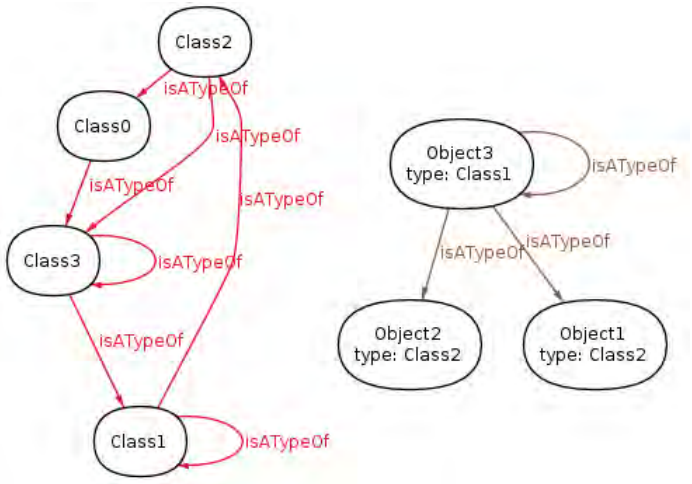


Fig. 11. Second instance

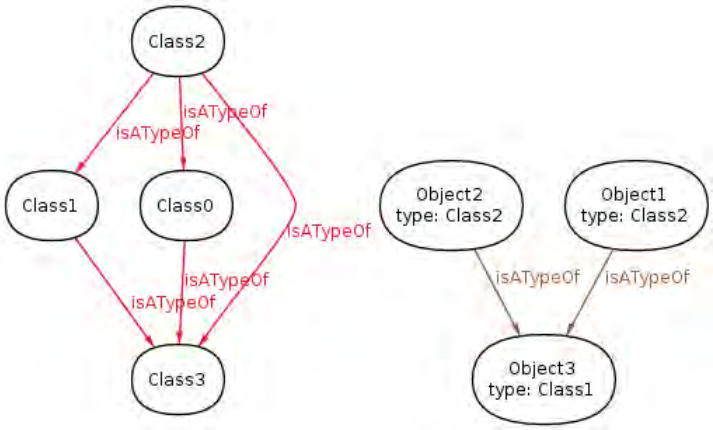


Fig. 12. Third instance

References

1. UML 2.0 superstructure specification. Tech. rep., Object Management Group (OMG) (2005)
2. Amalio, N., Polack, F.: Comparison of formalization approaches of UML class constructs in Z and Object-Z. In: ZB 2003: Formal Specification and Development in Z and B. Springer-Verlag (2003)
3. Ambler, S.W.: The Elements of UML 2.0 Style. Cambridge University Press (2005)
4. Aredo, D.B.: Formalizing UML class diagrams in PVS. In: Workshop on Rigorous Modeling and Analysis with the UML: Challenges and Limitations, at OOPSLA'99. Denver, Colorado, USA (1999)

5. Bennett, S., Skelton, J., Lunn, K.: UML. Schaums outlines, USA (2005)
6. Berardi, D., Calvanese, D., Giacomo, G.D.: Reasoning on UML class diagrams using description logic based systems. In: Proc. of the KI2001 Workshop on Applications of Description Logics. CEUR Electronic Workshop Proceedings (2001)
7. Booch, G., Rumbaugh, J., Jacobson, I.: The Unified Modeling Language User Guide. Addison-Wesley Professional, 2^{nd} edn. (2005)
8. Brucker, A.D., Wolff, B.: A proposal for a formal OCL semantics in Isabelle/HOL. Lecture Notes in Computer Science 2410(147-175) (2002)
9. Bruel, J., Henderson-Sellers, B., Barbier, F., Parc, A.L., France, R.: Improving the UML metamodel to rigorously specify aggregation and composition. In: 7^{th} International Conference on Object-Oriented Information Systems (OOIS '01) (2001)
10. Cali, A., Calvanese, D., Giacomo, G.D., Lenzerini, M.: A formal framework for reasoning on UML class diagrams. In: Proceedings of the 13^{th} International Symposium on Foundations of Intelligent Systems, LNCS 2366. Springer-Verlag (2002)
11. Clavel, M., Egea, M.: Equational specification of UML+OCL static class diagrams. Tech. rep. (2006)
12. Evans, A.S.: Reasoning with UML class diagrams. In: Proceedings of the Second IEEE Workshop on Industrial Strength Formal Specification Techniques. IEEE Computer Society, Washington, DC, USA (1998)
13. Fowler, M.: UML Distilled: A Brief Guide to the Standard Object Modeling Language. Addison-Wesley, 3^{rd} edn. (2003)
14. France, R.B., Bruel, J.M., Larrondo-Petrie, M.M., Shroff, M.: Exploring the semantics of UML type structures with Z. In: Proceedings of the Second IFIP International Conference on Formal Methods for Open Object-based Distributed Systems (FMOODS'97) (1997)
15. Funes, A., George, C.: Formal fundations in RSL for UML class diagrams. Tech. rep., The United Nations University (2002)
16. Gogolla, M., Bohling, J., Richters, M.: Validating UML and OCL models in USE by automatic snapshot generation. Software and System Modeling 4(4), 386–398 (2005)
17. Hassan, W., Logrippo, L.: Governance Policies for Privacy Access Control and their Interactions. IOS Press, Canada (2005)
18. Jackson, D.: Software Abstractions: Logic, Language and Analysis. MIT Press, England (2006)
19. Jordi Cabot, R.C., Riera, D.: Verification of UML/OCL class diagrams using constraint programming. In: ICST Workshop on Model Driven Engineering, Verification and Validation: Integrating Verification and Validation in MDE (MoDeVVa'2008) (2008)
20. Kim, S.K., Carrington, D.: Formalizing the UML class diagram using Object-Z. In: 2^{nd} International Conference on The Unified Modeling Language. Springer, Fort Collins, Colorado (1999)
21. Kim, S.K., Carrington, D.: A formal mapping between UML models and Object-Z specifications. In: ZB 2000, LNCS 1878. Springer-Verlag (2000)
22. Kleppe, A., Rensink, A.: A graph-based semantics for UML class and object diagrams. Tech. Rep. TR-CTIT-08-06, Enschede (January 2008)
23. Kuske, S., Gogolla, M., Kollmann, R., Kreowski, H.J.: An integrated semantics for UML class, object and state diagrams based on graph transformation. In: Lecture Notes in Computer Science. pp. 11–28. Springer (2002)
24. Kyas, M., Fecher, H., de Boer, F.S., Jacob, J., Hooman, J., Zwaag, M.V.D., Arons, T., Kugler, H.: Formalizing UML models and OCL constraints in PVS. Electronic Notes in Theoretical Computer Science 115, 39–47 (January 2005)

25. Larman, C.: Applying UML and Patterns : An Introduction to Object-Oriented Analysis and Design and Iterative Development. Prentice Hall, USA, 3^{rd} edn. (2004)
26. Ledang, H., Souquiéres, J.: Modeling class operations in B: Application to UML behavioral diagrams. In: 16^{th} Annual International Conference on Automated Software Engineering, 2001 (ASE 2001) (2001)
27. Meng, S., Aichernig, B.K.: Towards a coalgebraic semantics of UML: Class diagrams and use cases. Tech. rep., The United Nations University (2003)
28. Nolte, T.: Exploring Filesystem Synchronization with Lightweight Modeling and Analysis. Ph.D. thesis, M.I.T., U.S.A. (2002)
29. Pender, T.: UML Bible. John Wiley & Sons (2003)
30. Pilone, B., Pitman, N.: UML 2.0 in a Nutshell. O'Reilly (2005)
31. Rumbaugh, J., Jacobson, I., Booch, G.: The Unified Modeling Language Reference Manual. Addison-Wesley Professional, 2^{nd} edn. (2005)
32. Valles-Barajas, F.: A formal model for a requirements engineering tool. In: First Alloy Workshop, co-located with the 14^{th} ACM/SIGSOFT Symposium on Foundations of Software Engineering (FSE'06). ACM, Portland, Oregon (2006)
33. Valles-Barajas, F.: A formal model for a state machine modeling tool: A lightweight approach. In: 3rd IEEE Systems and Software Week (SASW 2007). IEEE/NASA, Baltimore, Maryland (2007)
34. Valles-Barajas, F.: A metamodel for the requirements diagrams of SysML. IEEE Latin America Transactions 8(3), 259–268 (2009)
35. Valles-Barajas, F.: Use of a lightweight formal method to model the static aspects of state machines. Journal of Innovations in Systems and Software Engineering (ISSE): A NASA Journal 5(4), 255–264 (2009)
36. Valles-Barajas, F.: A precise specification for the modeling of collaboratios. Malaysian Journal of Computer Science 23(1), 18–36 (2010)
37. Warmer, J., Kleppe, A.: The Object Constraint Language: Getting Your Models Ready for MDA. Addison-Wesley Professional, 2^{nd} edn. (2003)

Fernando Valles-Barajas obtained a graduate degree in Computer Science at Center for Research and Graduate He received an MS in Control Engineering (1997) and a PhD in Artificial Intelligence (2001) from Monterrey He was a research assistant at Mechatronics department of ITESM campus Monterrey (1997-2001).

He received certification as a PSP Developer from the Software Engineering Institute of Carnegie Mellon

He is member of the IEEE and ACM.

His research interests include topics in Software Engineering and Control Engineering.

Currently he is full-time professor in the Information Technology Department at Universidad Regiomontana, Monterrey, Nuevo León, México. He also teaches modules in Software Engineering.

Received: February 10, 2011; Accepted: August 29, 2011.

DOI 10.2298/CSIS110413071W

S-TRAP: Optimization and Evaluation of Timely Recovery to Any Point-in-time (TRAP)

Chao Wang[1], Zhanhuai Li[1], Na Hu[1] and Yanming Nie[1]

[1] School of Computer Science, Northwestern Polytechnical University
710072 Xi'an, China
{wch, huna601, yanming}@mail.nwpu.edu.cn, lizhh@nwpu.edu.cn

Abstract. This paper presents S-TRAP, a novel block-level CDP (Continuous Data Protection) recovery mechanism based on TRAP. In accordance with a certain time interval L, S-TRAP breaks down the parity chain of TRAP and generates new sub-chains. Besides, S-TRAP introduces previous block cache which can reduce the negative impact on the primary storage system. Both mathematical analysis and experimental evaluation demonstrate that S-TRAP not only has the advantage of high recovery efficiency and reliability, but also further reduces the parity storage usage. Even more important, S-TRAP reduces the negative impacts on primary storage system performance to a large extent.

Keywords: data protection, snapshot, CDP, data backup, data recovery.

1. Introduction and Related Work

With explosive growth of networked information services and e-commerce, data protection has become one of the major issues for business organizations, government institutions and individuals [17]. Due to the close coupling between information service and the maturity of storage technology, failures such as hardware/software defects, human errors, virus attacks, and power outage can cause data damage or data loss. Recent work [11, 20] has demonstrated that the loss caused by data unavailability or data damage can be up to millions of dollars per hour. In order to protect data from possible failures and to recover data in case of failures, data protection technology is very necessary [26].

In disaster recovery theory, RPO (Recovery Point Objective) and RTO (Recovery Time Objective) [4, 11] are two key criteria to evaluate data protection mechanisms. RPO measures the data loss by time while RTO reflects the time duration spent on data recovery. Depending on the different RPOs, we summarize existing data protection mechanisms in three categories as shown in Fig. 1.

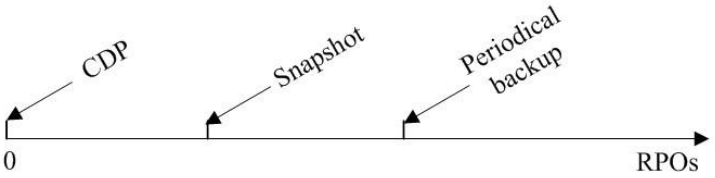


Fig. 1. Data Protection mechanisms for different RPOs

Periodical backup. Traditional periodical backup [2, 22, 27] has been widely adopted in data protection systems. Typically, backups are done on a daily, weekly or monthly basis. Since the backups can only be done in an offline manner, there may be exist the problems of data unavailability during the backup procedure. Besides, a huge amount of storage has to be occupied to keep the backups, so data compression technologies are often used to save the backup storage usage. As a result, periodical backup consumes too much time and storage space and also degrades system performance.

Snapshot. A snapshot is a point-in-time image of a collection of data which allows online backup. Generally, there are two common snapshot mechanisms: COW (Copy-On-Write) [1, 21, 23, 32, 33] and ROW (Redirect-On-Write) [6, 19, 30]. In order to save storage space, COW snapshot obtains and maintains the previous images of a data block upon the first write operation to that data block. Different from COW, ROW snapshot redirects all the write operations to the snapshot storage. Snapshots can be integrated in disk arrays [14], volume managers, file systems [3, 8, 9, 12, 21, 28, 29] and backup systems. Compared with periodical backup, snapshot can be created online, use less storage space, and has less negative impact to application performance. But it still cannot achieve timely recovery to any point in time.

Continuous data protection (CDP). CDP can provide timely recovery to any point in time at block level. Traditional CDP mechanism [4, 13, 18] keeps a log of changed data for each data block in timestamp order. Performance overhead and recovery efficiency are the important parameters to measure CDP system, much work [5, 16, 25, 35] has been done. Mariner [16] is a block-level CDP system that is designed to minimize the performance overhead associated with block update logging. Zhu and Chiueh [35] proposed a portable and efficient user-level CDP system, which incurs minimal performance overhead. TH-CDP [25] mainly focuses on general purpose, easy recovery and fast check pointing for fast recovery. Rather than per block basis schemes, GSP_BCDP [5] provides faster recovery by per volume basis.

But on the other hand, the huge amount storage space, which is required to keep log, limits the application and development of CDP. Some research efforts, such as Clotho [7], are made to reduce the storage space usage of traditional CDP to increase adoption of data de-duplication technologies. However, these methods cannot effectively solve log space usage problem of traditional CDP. TRAP [31, 33, 34] is a novel CDP mechanism, by compressing and saving the XOR parity among changed data blocks along

the time dimension, it can improve space efficiency significantly. But its recovery efficiency and reliability is low; what's more, it has a great negative impact on storage system performance. ST-CDP [15], a TRAP-based optimal implementation, improves the recovery efficiency and reliability by adding snapshots between parity chains, but still has great impact on storage system performance.

To overcome the shortcomings of existing continuous data protection mechanisms, in this paper we presented a novel block level CDP architecture based on TRAP, named S-TRAP. In accordance with certain time interval L, S-TRAP breaks down the parity chain of TRAP and generates new sub-chains. In addition, the Previous Block Cache which contains partial images of data blocks at the time point *T-1* is leveraged to reduce the negative impact on primary storage system. In comparison with ST-CDP, S-TRAP not only has the advantage of fast recovery efficiency and high reliability, but also further reduces the parity storage occupation. Even more important, S-TRAP reduces the negative impact on primary storage system performance to a large extent. Formal analysis and preliminary experiment result show that S-TRAP have low RTO and high reliability, while retaining lower storage overhead and less infection on primary storage system.

The rest of this paper is organized as follows: Section 2 gives a brief overview of the TRAP architecture. Section 3 presents our optimal CDP mechanism S-TRAP in detail. In Section 4, we use mathematical model to guide our design in terms of six aspects, including recovery time, parity storage space usage, reliability, impact on system performance, recovery direction and optimal L. Section 5 describes the experiment settings. Numerical result and discussions are presented in Section 6. We conclude the paper in Section 7.

2. Brief Overview of TRAP

Instead of keeping all versions of a data block as being modified by write operations, TRAP only keeps a parity log of each write on the parity storage. Fig. 2 shows the basic design of TRAP. Suppose that at time point t, a write operation is submitted to block B which updates its content from B_{t-1} to B_t. TRAP generates the new parity of block B as follows:

$$P_t = B_{t-1} \oplus B_t \tag{1}$$

where $\oplus$ is a bitwise XOR operator. And then TRAP appends the new parity P_t to the parity chain $(P_1, P_2, \cdots, P_{t-1}, P_t)$.

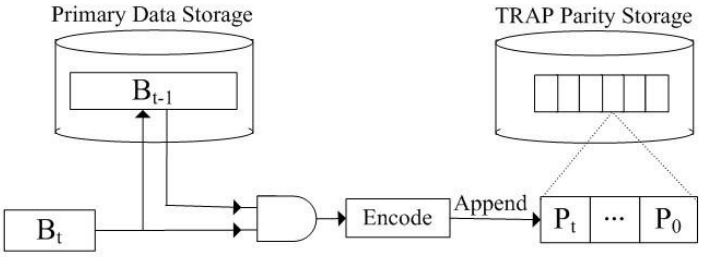


Fig. 2. Basic design of TRAP architecture

Extensive experiments [33] have demonstrated a very strong content locality that only 5%-20% of bits inside a data block are actually changed by a write operation. Because of the characteristics of bitwise XOR operation, the parity can reflect the exact changes made by the new write operation at the bit level, i.e., except a very small portion of nonzero bits, most bits in the new parity are zeros. Consequently the parities can be compressed easily with a high compression ratio. Therefore, the storage space required to keep track of write operations can be greatly reduced by orders of magnitude.

TRAP architecture provides a double-way data recovery capability, either forward or backward, referred to as redo and undo, respectively. Consider the parity chain $(P_1, P_2, \cdots, P_{t-1}, P_t)$ corresponding to data block B. Suppose we have both the initial and latest images of block B, referred to as B_0 and B_t. If we need to recovery block B to the previous image of time point $r\ (0 < r < t)$, TRAP performs the following process:

$$B_r = B_0 \oplus P_1 \oplus P_2 \oplus \cdots \oplus P_r \tag{2}$$

where B_r denotes the data image of block B at time point r.

The above process represents a typical forward/redo recovery. Similarly, to recover from the opposite direction, TRAP performs backward/undo recovery according to the following process:

$$B_r = P_{r+1} \oplus P_{r+2} \oplus \cdots \oplus P_t \oplus B_t \tag{3}$$

Definition 1 For any recovery time point r, the recovery chain of block B is $(P_1, P_2, \cdots, P_{r-1}, P_r)$ with length r, or $(P_{r+1}, P_{r+2}, \cdots, P_{t-1}, P_t)$ with length $t-r$.

We can infer from Equation (2) and (3) that any damaged parity in the recovery chain will lead to the failure of the whole recovery process.

TRAP has a negative impact on primary storage system performance. This is because when generating parity, TRAP has to firstly acquire the previous data image from primary storage system which can definitely increases the I/O response time. Although TRAP can benefit from the parity generation component of RAID (RAID 4, 5, etc.), it also has some limitations: (a) only the RAID with XOR parity can be chosen; (b) it requires implementing TRAP

in the RAID controller; (c) both primary storage and parity storage have to be under the same RAID controller. In addition, we can also infer from Equation (2) and (3) that the recovery time becomes longer as the parity chain gets longer, and the recovery efficiency decreases as the length of the recovery chain increases. The invalid parity results in the invalidity of the corresponding recovery chains, obviously, the failure probability is larger in a longer recovery chain. Therefore, the reliability of TRAP also decreases with the growth of recovery chain length increases. All these restrictions can limit the application of TRAP.

ST-CDP is an optimal implementation of continuous data protection in Linux kernel based on TRAP. By adding snapshots between parity chains, ST-CDP can improve the recovery efficiency and reliability, but the snapshots increase the parity storage usage as well. Furthermore, ST-CDP also has to obtain the previous data image from primary storage; therefore, high I/O response problem still exists.

3. Design of S-TRAP Mechanism

3.1. Parity recording

In accordance with a certain time interval L, S-TRAP divides the parity chain of TRAP and generates new sub-chains. Consecutive L parities from a sub-chain are illustrated in Fig. 3. Different form ST-CDP, S-TRAP does not add snapshots between parity chains. Instead, we use a special generation method for the FIRST parity of each sub-chain. Consider data block B, the first parity of each sub-chain is generated as:

$$P_{kL+1}{}' = B_0 \oplus B_{kL+1} \tag{4}$$

while the generating method of the other parities is consistent with TRAP, as shown in Equation (1).

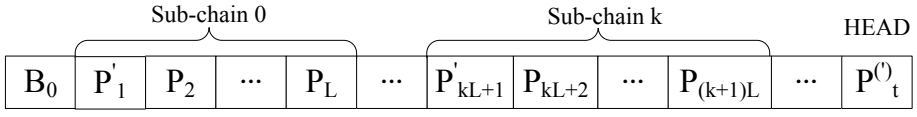


Fig. 3. Parity chain of S-TRAP

Definition 2 For any data block B, $\left(\mathrm{P'}_{\mathrm{kL+1}}, P_{kL+2}, \cdots, P_{(k+1)L}\right)$ is the $k-th$ sub-chain.

Different from TRAP, the idea of our approach breaks down parity chain to sub-chains. Since all parities of write operations are kept on backup storage with the form of SUB-chains in order to provide Timely Recovery to Any Point in time, we name our approach as S-TRAP.

3.2. Previous Block Cache

Caching is one of the major technologies used to improve I/O performance in disk array. However, if TRAP wants to takes the advantage of the cache in disk array to reduce the negative impact on primary storage system, it would require: (a) implementing TRAP architecture in the disk array controller; (b) both primary storage and parity storage to be under the same disk array controller. In order to eliminate these restrictions, S-TRAP manages a separate cache to improve its I/O efficiency, which is called Previous Block Cache (PBC).

Suppose that at time point t, a write operation is submitted to block B, S-TRAP firstly checks in PBC. If there is a cache hit, S-TRAP:

(a) commits the new data image of block B directly to primary storage system,

(b) generates parity by previous data image of block B in Cache, and commits it to parity storage system, and

(c) replaces the new image of block B into Cache according to replacement algorithm;

Otherwise, it:

(a) gets the previous image of block B from primary storage system,

(b) commits the new data image of block B directly to primary storage system,

(c) generates parity by previous data image of block B in Cache, and commits it to parity storage system, and

(d) replaces the new image of block B into Cache according to replacement algorithm.

Upon cache hitting, S-TRAP saves one additional I/O operation. Therefore, PBC can reduce the negative impact on primary storage system.

3.3. Data Recovery

Without loss of generality, consider the sub-chain $\left(\mathrm{P}'_{\mathrm{kL+1}}, P_{kL+2}, \cdots, P_{(k+1)L}\right)$ which corresponds to data block B. In order to restore it to the previous recovery time point $r = kL + i\ (1 \le i \le L)$, S-TRAP performs the following process:

$$B_r = B_{kL+i} = B_0 \oplus P'_{kL+1} \oplus P_{kL+2} \oplus \cdots \oplus P_{kL+i} \quad (5)$$

Due to the particularity of the first parity in each sub-chain, we do analysis under two conditions:

(a) $r = kL$, recovery time point r is the time point which associates with the first parity in the $k-th$ sub-chain. In this case, S-TRAP only needs to perform $B_r = B_0 \oplus P'_{kL+1}$ to do a forward/redo recovery. The length of recovery chain $\left(P'_{kL+1}\right)$ is 1 and XOR operation only needs to be performed once. So, recovery efficiency is high, and

(b) $r = kL + i,\ 0 < i \le L$, this is the most general case, and recovery time point *r* falls on the time point which associates with other parities of the $k-th$ sub-chain. S-TRAP performs forward/redo recovery based on Equation (5). The length of recovery chain $\left(P'_{kL+1}, P_{kL+2}, P_{kL+3}, \cdots, P_{kL+i}\right)$ is i, so XOR operation needs to be performed i times.

4. Formal Analysis of S-TRAP

In this session, we mathematically model the S-TRAP mechanisms in detail, including recovery time, parity storage space usage, reliability, and its impacts on system performance and recovery direction. In addition, we calculate the key optimal L value, which has a major impact on parity storage space usage and recovery time. In order to facilitate the description, we define the following notations as shown in Table 1.

Table 1. Definition of Symbols

Symbol	Definition
L	The length of the sub-chain
IO_{rate}	I/O throughput of storage system
B_{size}	Size of the data block (in MB)
C	Compression ratio of parity P
C'	Compression ratio of the special parity P'
T_{dec}	Decoding time of parity
T_{XOR}	XOR operation time

C' is the compression ratio of the special parity. The special parity reflects the exact changes of many write operations. It has less zero bits, therefore C' is smaller than C. To simplify the discussion, we assume that C' is a constant. In addition, because of the similarity of two recovery directions of TRAP (i.e., forward/redo and backward/undo), we only select the forward/redo recovery in the following discussion.

4.1. Recovery Time

Theorem 1 For any recovery time point r, with using S-TRAP mechanism, the maximum length of recovery chain is L, and the expected length of recovery length is $(L+1)/2$.

Proof. For any recovery time point r, without loss of generality, it certainly falls on the time interval associated with one of the sub-chains $\left(\mathrm{P'}_{\mathrm{kL+1}}, P_{kL+2}, \cdots, P_{(k+1)L}\right)$, and it is uniformly distributed among $kL+1$ and $(k+1)L$.

(a) The worst case is that the recovery time point r is the time point which is associated with the last parity of sub-chain k, i.e., $r=(k+1)L$. Therefore, the maximum length of recovery chain is L.

(b) Because recovery time r is uniformly distributed on a closed interval $\left[kL+1,(k+1)L\right]$, the expected length of recovery length is:

$$E\left(L_r\right)=\sum_{k=1}^{L} k\times P\left\{x=k\right\}=\frac{L+1}{2} \tag{6}$$

Theorem 2 For any recovery time point r, the recovery time of TRAP is proportional to r, and the recovery time of S-TRAP is proportional to L.

Proof.

(a) The recovery time of TRAP is:

$$T_{TRAP}=\frac{B_{size}}{IO_{rate}}+\left(T_{dec}+T_{xor}+\frac{B_{size}}{C\cdot IO_{rate}}\right)\cdot r+\frac{B_{size}}{IO_{rate}} \tag{7}$$

where the first item is the time of getting and transferring the initial image B_0, the second is the time of decoding, and the last is the time of transferring and recording the final image. It can be deducted from equation (7) that the recovery time of TRAP is proportional to recovery time point r.

(b) The recovery time of S-TRAP is:

$$T_{S-TRAP}=\frac{B_{size}}{IO_{rate}}+\left(T_{dec}+\frac{B_{size}}{C'\cdot IO_{rate}}\right)+\left(T_{dec}+T_{xor}+\frac{B_{size}}{C\cdot IO_{rate}}\right)\cdot E\left(L_r\right)+\frac{B_{size}}{IO_{ra} \tag{8}$$

where the first item is the time of getting and transferring the initial image B_0, the second is the decoding time of the first parity in sub-chain, the third is the decoding time of other parities, and the last is the time of transferring and recording the final image. We substitute Equation (6) to Equation (8), so we can draw a conclusion that the recovery time of S-TRAP is proportional to L and independent of the recovery time point r.

In the S-TRAP mechanism, the expected number of parities which is involved in XOR operation is $(L+1)/2$ (L for the worst case); therefore, the

recovery time of S-TRAP always fluctuates within a small range which is defined by L value and Equation (8). As a result, S-TRAP has high and stable recovery efficiency.

The recovery time of both ST-CDP and S-TRAP is independent of RPO, but is dependent on d and L. The recovery time is similar between two mechanisms, which consist of 1) getting and transferring snapshot, and 2) decoding and transferring the final image. When the d value of ST-CDP equals to the L value of S-TRAP, both mechanisms have the same numbers of XOR operations. Thus, the recovery time of two mechanisms is approximately the same.

4.2. Parity Storage Space Usage

Theorem 3 Compared with TRAP, the additional parity storage space usage ratio of S-TRAP is inversely proportional to L.

Proof. Without loss of generality, consider data block B at the latest time point t,

The parity storage space of TRAP is:

$$S_{TRAP} = \frac{B_{size} \cdot t}{C} \tag{9}$$

The parity storage space of S-TRAP is:

$$\begin{aligned} S_{S-TRAP} &= B_{size} \cdot \frac{1}{C'} \cdot \left\lceil \frac{t}{L} \right\rceil + B_{size} \cdot \frac{1}{C} \cdot \left(t - \left\lceil \frac{t}{L} \right\rceil \right) \\ &\approx \frac{B_{size}}{C'} \cdot \frac{t}{L} + \frac{\mathrm{B}_{size}}{C} \cdot \left(t - \frac{t}{L} \right) \end{aligned} \tag{10}$$

where the first item is the storage space usage of the first parities in sub-chains, the second item is the storage space usage of the other parities.

Additional storage space usage is:

$$\Delta S = \frac{B_{size} \cdot t}{L} \cdot \left(\frac{1}{C'} - \frac{1}{C} \right) \tag{11}$$

Additional parity storage space usage ratio is:

$$\frac{\Delta S}{S_{TRAP}} = \frac{1}{L} \cdot \left(\frac{C}{C'} - 1 \right) \tag{12}$$

Therefore, it can be drawn from Equation (12) that the additional parity storage space usage ratio of S-TRAP is inversely proportional to L.

As mentioned above, S-TRAP neither adds any snapshots between parity chains, nor increases the total count of parities. When the snapshot interval d

of ST-CDP equals to the sub-chain length L of S-TRAP, the parity storage usage of S-TRAP is smaller than that of ST-CDP.

Theorem 4 When d equals to L, the storage usage of S-TRAP is smaller than ST-CDP.

The parity storage space of ST-CDP is:

$$S_{ST-CDP} = B_{size} \cdot \left\lfloor \frac{t}{d} \right\rfloor + \frac{B_{size}}{C} \cdot t \tag{13}$$

Delta parity storage usage between ST-CDP and S-TRAP is:

$$\Delta S = B_{size} \cdot \left\lceil \frac{t}{d} \right\rceil \cdot \left(1 - \frac{1}{C'}\right) + \frac{B_{size}}{C} \cdot \left\lceil \frac{t}{d} \right\rceil \tag{14}$$

where $C'(C' \geq 1)$ is the compression ratio of special parity, thus, $\Delta S > 0$.

Therefore, the storage usage of S-TRAP is smaller than that of ST-CDP when d equals to L.

4.3. Reliability

Due to hardware and/or software failures, disasters or outages, both the initial image and parities are likely to be damaged. It is obvious that invalid parity results in the invalidity of the corresponding recovery chains and the invalid recovery chain can make us unable to recover data to previous time points. Therefore, improving the reliability of recovery chain is extremely important for CDP.

In S-TRAP mechanism, the valid probability of recovery chain only correlates to L, and would not decrease as time point grows; while the valid probability of recovery chain of TRAP decreases with the growth of recovery chain length.

Definition 3 The invalid probability of initial image and parity is p.

Theorem 5 For any recovery time point r, with using S-TRAP mechanism, the valid probability of recovery chain is $(1-p)^{L+1}$ in the worst case, and the mathematical expectation of the valid probability of recovery chain is $\frac{1}{pL} \cdot \left[(1-p)^2 - (1-p)^{L+2} \right]$.

Proof. For any recovery time point r, without loss of generality, it certainly falls on the time interval which is associated with one of the sub-chains $\left(\mathrm{P'}_{\mathrm{kL+1}}, P_{kL+2}, \cdots, P_{(k+1)L}\right)$, and it is uniformly distributed among $kL+1$ and $(k+1)L$.

(a) The worst case is that we have the recovery time point $r = (k+1)L$, and the length of recovery chain reaches maximum value L. In order to ensure the validity of the recovery chain, both the initial image and all parities in the

sub-chain must be valid. Therefore, the valid probability of recovery chain is minimized in such case:

$$\mathrm{P}_{S-TRAP} = (1-p)^{L+1} \tag{15}$$

(b) Because recovery time r is uniformly distributed on closed interval $[kL+1,(k+1)L]$, the expected valid probability of recovery chain is:

$$E(\mathrm{P}_{S-TRAP}) = \frac{1}{L}\cdot\sum_{k=1}^{L}(1-p)^{k+1} = \frac{1}{pL}\cdot\left[(1-p)^{2}-(1-p)^{L+2}\right] \tag{16}$$

Theorem 6 For any recovery time point r, with using TRAP mechanism, the valid probability of recovery chain is decreasing with the increase of time point r.

Proof. For any recovery time point r, with using TRAP mechanism, all the parities from time point 1 to r, and the initial image should be valid to ensure the validity of recovery chain. So, the valid probability of recovery chain is:

$$\mathrm{P}_{TRAP} = (1-p)^{r+1} \tag{17}$$

According to Equation (17), the valid probability of recovery chain for TRAP mechanism is decreasing with the increase of time point r.

Based on the above analysis, compared with TRAP mechanism, the valid probability of recovery chain in S-TRAP mechanism is only correlated to L, and would not decrease with the growth of time point. Therefore, S-TRAP can provide much higher reliability than TRAP can.

S-TRAP does not insert snapshot between parity chains as ST-CDP, so initial image is required for every recovery operation. Therefore, this causes a single point of failure.

4.4. Impact on system performance

For each write operation:

(a) both TRAP and ST-CDP have to get the image of previous time points from primary storage system first, and thereafter the new image can be committed to parity storage; while

(b) S-TRAP firstly checks whether the block cache is hit or not. If there is a hit, it gets the image of previous time point from PBC directly, and then one I/O operation is saved for primary storage system. Cache hit makes committing the new image to primary storage faster, thus reduced the IO response time of primary storage system. If there is a miss, S-TRAP has to get the image from primary storage as well, and then puts the image into PBC according to certain replacement algorithm.

Therefore, for each write operation, both TRAP and ST-CDP need one additional disk I/O on primary storage, while S-TRAP depends on whether cache is hit or not. Increasing the size of PBC can improve the hit ratio, thus,

S-TRAP saves more I/O operations for primary storage system accordingly. Meanwhile, S-TRAP has a higher memory footprint than TRAP and ST-CDP do.

4.5. Recovery Direction

In S-TRAP mechanism, the special parity in each sub-chain is generated by bitwise XOR operation between initial and the latest image of data block. Due to the special generation method, S-TRAP could only recover in one direction (i.e., forward/redo). Compared with the double-way recovery of TRAP and ST-CDP, S-TRAP cannot recover data when the initial image is damaged. However, except for the initial image stored in primary storage, S-TRAP also keeps an addition backup in parity storage. This lowers the probability of an invalid initial image. Moreover, S-TRAP can also escalate to double-way recovery easily by adding an additional normal parity at the first time point of each sub-chain.

4.6. Optimal L value

Recovery time and parity storage space usage are the two key factors to measure the recovery cost of CDP mechanism. Next, we will come out with a definition for the recovery cost function of S-TRAP.

Definition 4 The recovery cost of S-TRAP is the product of its recovery time and parity storage space usage:

$$\begin{aligned} G(L) &= T_{S-TRAP}(L) \cdot S_{S-TRAP}(L) \\ &= (A_1 + A_2 L) \cdot \left(A_3 + \frac{A_4}{L} \right) \end{aligned} \tag{18}$$

where

$$A_1 = \frac{2 \cdot B_{size}}{IO_{rate}} + \left(T_{dec} + T_{xor} + \frac{B_{size}}{C' \cdot IO_{rate}} \right) - \frac{1}{2} \left(T_{dec} + T_{xor} + \frac{B_{size}}{C \cdot IO_{rate}} \right)$$

$$A_2 = \frac{1}{2} \left(T_{dec} + T_{xor} + \frac{B_{size}}{C \cdot IO_{rate}} \right);$$

$$A_3 = \frac{B_{size} \cdot t}{C};$$

$$A_4 = B_{size} \cdot t \cdot \left(\frac{1}{C'} - \frac{1}{C} \right)$$

The minimum value of $G(L)$ exists when $L = L_0 = \sqrt{\frac{A_1 A_4}{A_2 A_3}}$. Since L is an integer, we choose the integer closest to L_0 as optimal L value.

5. Experiment Setup

In order to evaluate the S-TRAP mechanism, we designed and implemented a prototyped system. Fig. 4 shows the architecture of this prototype which integrates such three mechanisms, as TRAP, ST-CDP and S-TRAP. The prototype is a block device driver layered below file system, database system or other applications in the Linux kernel and therefore it works transparently to upper-level applications. The prototype captures write requests from upper levels and all the mechanisms keep all parities of each block at any time point according to the description in previous sections; In addition, all the mechanisms use the same open-source compression library, i.e., ***zlib***. Meanwhile, all the mechanisms are implemented independently of RAID controller. According to the discussion of ST-CDP [15], we set the d value of ST-CDP to 85. In addition, we chose five different block sizes: 4KB, 8KB, 16KB, 32KB and 64KB. Experimental configurations are listed in Table 2.

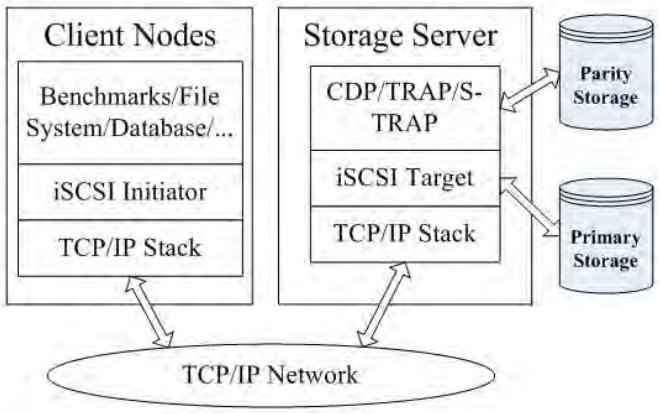


Fig. 4. System architecture of prototype

In addition, we have to determine the storage structure of parities on parity storage volume. Generally, there are two types of storage structure, timestamp-ordered and block address-ordered. Keeping parities in timestamp order does not need pre-allocating storage space for each data block. But it has such disadvantages as: (1) need to search previous parity when generating new parity, I/O costly; and (2) need to traverse parity log repeatedly when recovering data, once for each data block, so the recovery

algorithm complexity is high. For block address order, it is easy to recover because all the parities of each data block are kept together, so the recovery algorithm complexity is low. But here an efficiency parity space management algorithm is needed. For the purpose of getting an accurate evaluation, we choose block address-ordered to keep parities in the prototype for both TRAP and S-TRAP.

Table 2. Experiment Environments

Configuration	Client Nodes and Storage Server
CPU	AMD Opteron 850, 2.4GHz, 64bit
Memory	8GB ECC DDR RAM
Disk	73G SCSI Disk
OS	Red Hat Enterprise Server with kernel 2.6.18
Switch	Asus GigaX1124, Gigabit
NIC	1 Gbps PCI Ethernet Adapter

Appropriate workloads are important for performance studies [10]. We use popular benchmarks in our experiments, which are TPC-C and IoMeter. TPC-C is a well-known benchmark used to model the operation terminal of businesses where real-time transactions are processed. We choose one of the TPC-C implementations developed by the Hammerora Project [24] for Oracle database. According to the TPC-C specification, data tables for five warehouses with 25 users were built in order to issue transactional workloads to Oracle database. IoMeter is a flexible and configurable file system benchmark. In our experiments, we choose the mode of 30% writes and 70% reads for measuring the performance of file system and block device.

6. Results and Discussions

6.1. Impacts on system performance

In all the three mechanisms, the computation of bitwise XOR and decompression of parities may introduce additional overhead to primary storage system, and such overhead may negatively impact application performance accordingly. For the purpose of quantifying such impact, we measured the computation time of XOR and decompression as shown in Table 3. The block size refers to the decompression unit which is the size of parity before compression. It can be seen that both XOR and decompression computations are only take tens to hundreds of microseconds. Compared with the computation time of disk I/O, these times can be neglected in most cases.

Table 3. Measured Computation Time

Block Size (KB)	XOR Time (ms)	Decompress Time (ms)
4	0.025132	0.070960
8	0.050406	0.130094
16	0.101033	0.220106
32	0.211721	0.374015
64	0.420478	0.613474

Except for the slight impact caused by computation of bitwise XOR and decompression for parities, parity recording and reading, which needs disk I/O operation, have even more negative impact on primary storage system. In order to see quantitatively how much the impact is, we carried out the following experiments.

In S-TRAP mechanism, PBC is important to mitigate the negative impact on primary storage system. In order to find the suitable size of PBC, we measure the average I/O response time for different PBC size. With block size 4KB and 64KB, we run IoMeter benchmark (70% reads and 30% writes) for one hour on S-TRAP configured with different PBC size. As shown in Fig. 5, the average I/O response time decreases with PBC size increases, and becomes stable after 128MB. So we choose 128MB as the PBC size in the following experiments.

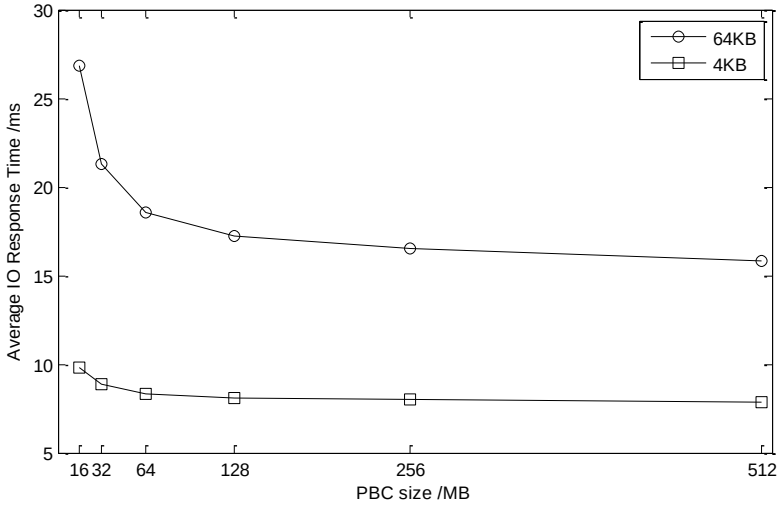


Fig. 5. Average IO Response Time over PBC size

With five different block sizes, we run IoMeter benchmark (70% reads and 30% writes) for one hour on TRAP, ST-CDP and S-TRAP, respectively. The results of original primary storage system with no data protection mechanism are used as a base line to show the negative impacts of the three CDP

mechanisms. Fig. 6 shows the measured results of average I/O response time for 70% writes and 30% reads of IoMeter benchmark.

As shown in Fig. 6, we observed in our experiments that all the three CDP mechanisms have negative performance impacts; TRAP has the most while S-TRAP the least. For block size 64KB, compared with original storage system, TRAP's average I/O response time is about 66.07% longer, 70.18% and 31.48% for ST-CDP and S-TRAP, respectively. It can be seen that S-TRAP reduces the negative impact on primary storage system about 52.35% compared with TRAP.

The reason is that, with the introduction of Previous Block Cache, the number of additional I/O reduced, and the impact on primary system reduced accordingly. After neglecting the slight impact caused by computation of bitwise XOR and decompress, TRAP and ST-CDP still have one additional I/O to get the image of pervious time point when recording parities. Thus, S-TRAP has less impact.

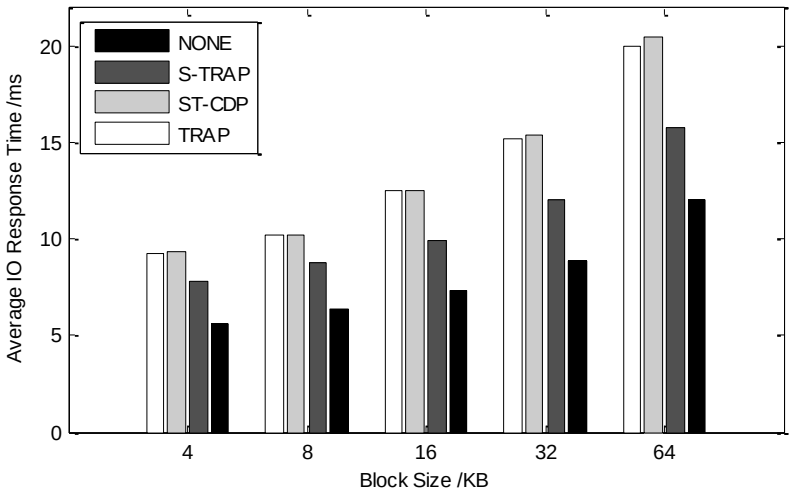


Fig. 6. Average I/O response time comparison

6.2. Sub-chain Length L

In S-TRAP mechanism, both recovery time and parity storage space usage heavily depend on the length of sub-chain length. According to the definition of recovery cost in Def. 4, through the trade-off between recovery time and parity space usage, recovery cost obtains the minimum when $L = L_0$. In order to find the appropriate L_0, we run TPC-C benchmark for one hour on S-TRAP configured with different sub-chain length L while the block size is 16KB, and then we perform recoveries for different configurations. The measured

recovery time and parity storage usage are shown in Figs. 7 and 8. The recovery cost under Def. 4 is shown in Fig 9.

As shown in Figs. 7 and 8, with the increasing of sub-chain length L, the parity storage usage decreases while the recovery time grows. In Fig. 9, we observed that the recovery cost obtains the minimum when sub-chain is 90. So we choose 90 as the sub-chain length in the following experiments.

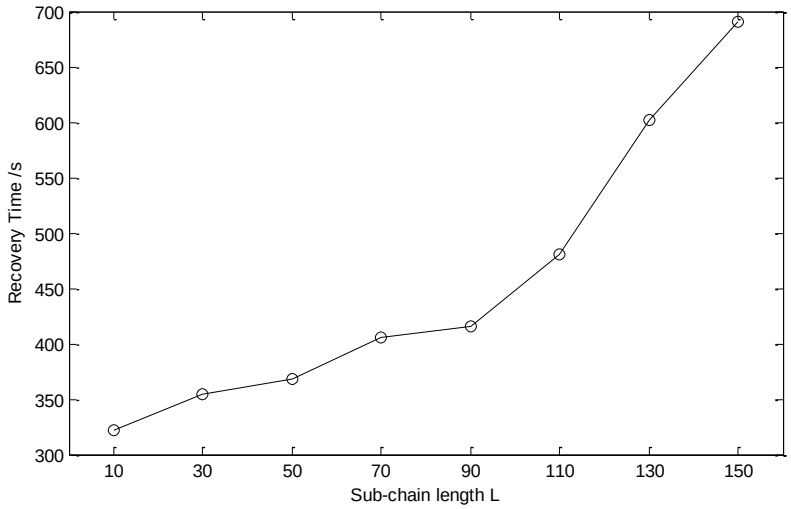


Fig. 7. Recovery time of S-TRAP configured with different sub-chain length

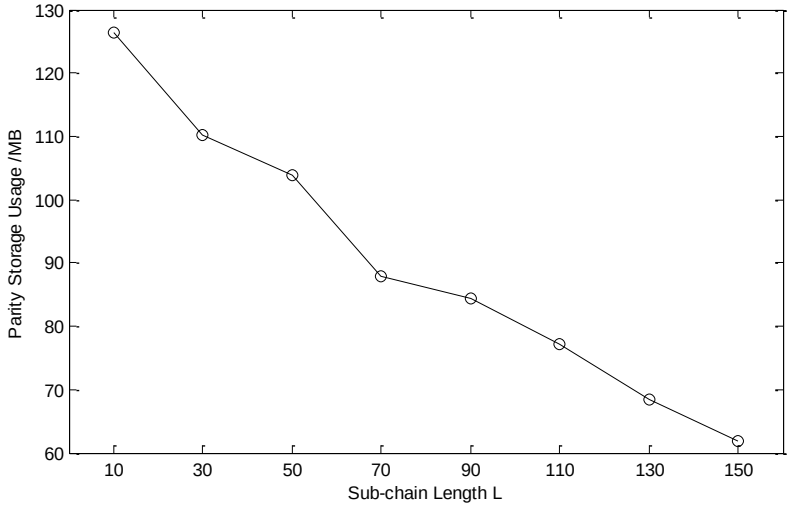


Fig. 8. Parity storage usage of S-TRAP configured with different sub-chain length

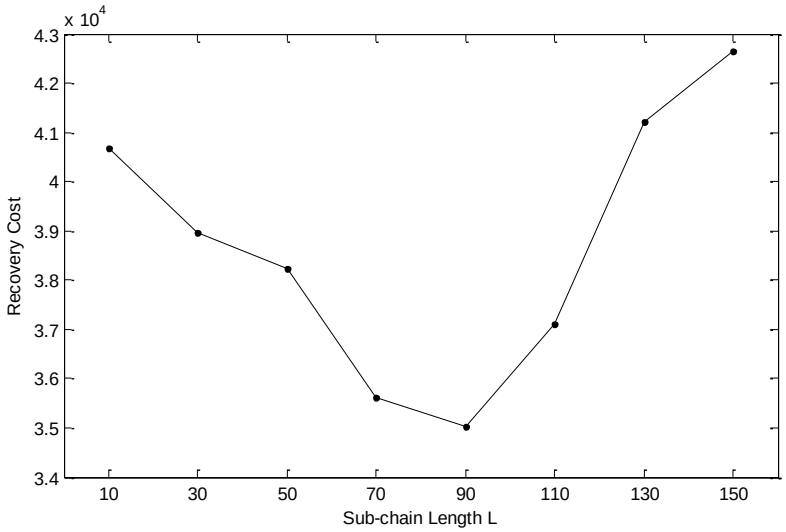


Fig. 9. Recovery cost of S-TRAP configured with different sub-chain length

6.3. Parity storage space usage

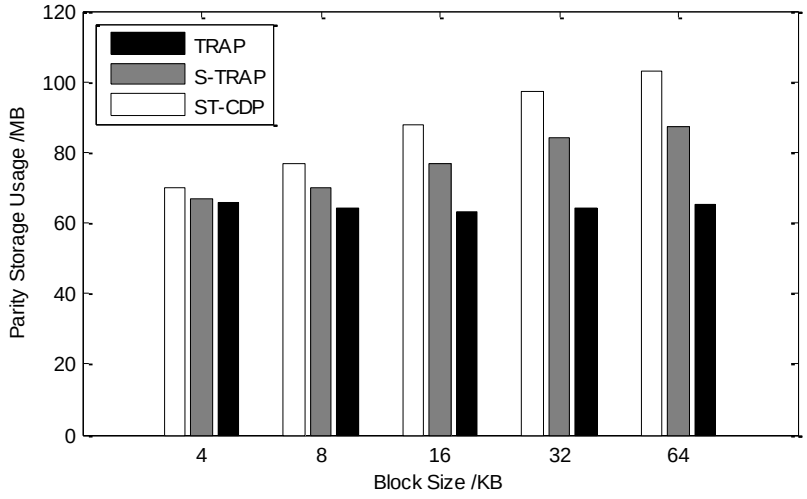


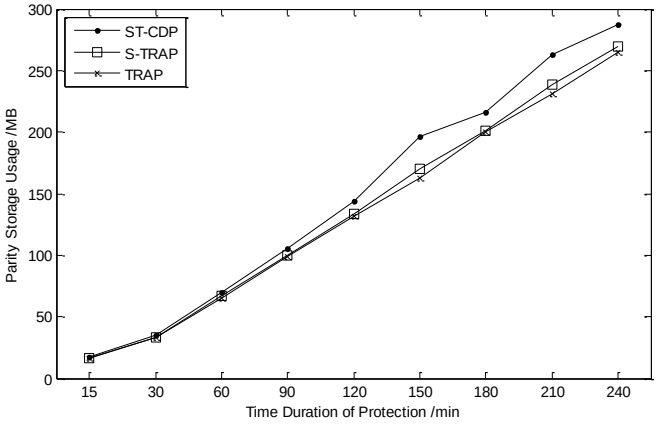
Fig. 10. Parity storage space usage comparison

This experiment is to measure the amount of parity storage space usage of different data protection mechanisms. With five different block sizes, we run TPC-C benchmark for one hour on TRAP, ST-CDP and S-TRAP,

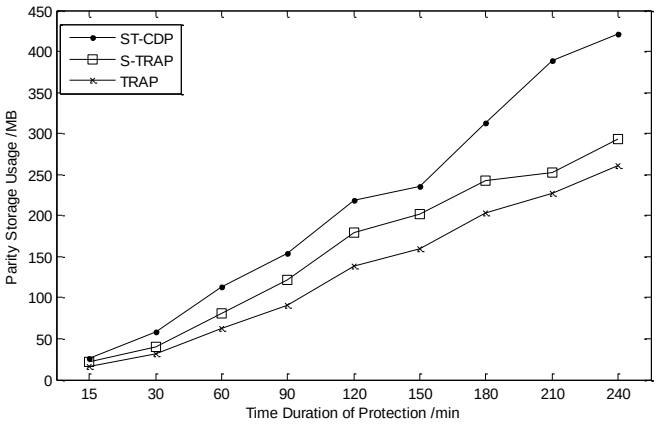
respectively. Then, we measured the parity storage space usage for different mechanisms. Fig. 10 shows the results of parity storage space usage comparison for TPC-C on the oracle database.

In Fig. 11, we also measure the space usage of TRAP, ST-CDP and S-TRAP for different protecting time durations with block size 4KB and 64KB.

It is shown in Fig. 10 that TRAP requires the least parity storage space while the ST-CDP the most. For example, for the block size of 16KB, the space usage of ST-CDP increases by 22.22% compared with TRAP, while S-TRAP is only 9.52%. We can get similar results from Fig. 11.



(a)



(b)

Fig. 11. Parity storage space usage over time duration: (a) Block Size=4KB; (b) Block Size=64KB

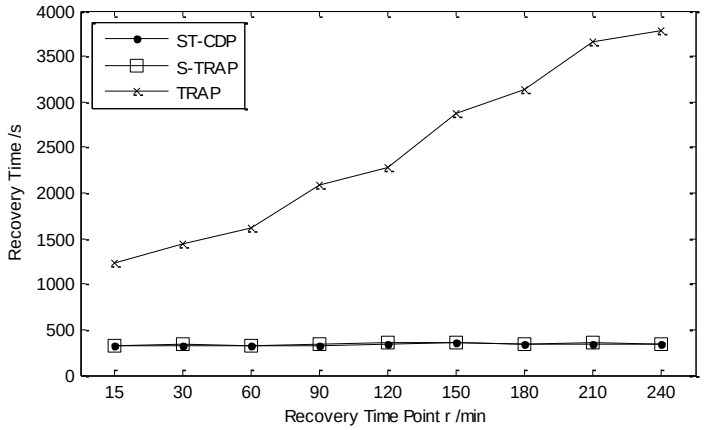
The reason is that S-TRAP performs the same parity algorithm as TRAP in most cases, only performs the special parity algorithm once for each L time points. It is noteworthy that S-TRAP does not increase the total number of

parities. On the other hand, ST-CDP supplely adds a snapshot for each d parities. Compared with parities with high compression ratio, snapshot takes up more storage space.

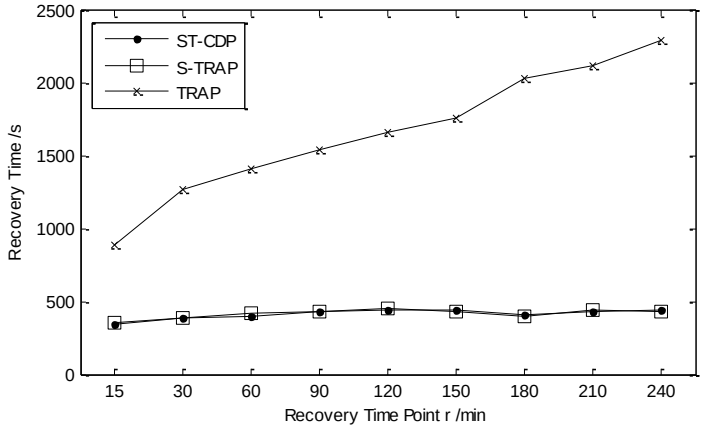
6.4. Recovery time

RTO (Recovery Time Objective) is a key indicator to evaluate a data protection mechanism, and it reflects the recovery duration of time after a disaster. In order to compare the recovery time among TRAP, ST-CDP and S-TRAP, we carried out this experiment. With block sizes of 4KB and 64KB, we run TPC-C benchmark for a sufficiently long time, more than five hours, on TRAP, ST-CDP and S-TRAP, separately. As the benchmark runs, the parity storage was filled with sufficient parities. Then, we perform recoveries for different time points in the past. Fig. 12 shows the measured results of recovery time for TRAP, ST-CDP and S-TRAP.

In Fig. 12, TRAP's recovery time increases obviously as RPO increases; while the recovery time of ST-CDP and S-TRAP keeps flat while RPO changes. This also illustrates the related results of Theorem (2). In addition, ST-CDP and S-TRAP perform almost the same in recovery ability. Thus it can be seem that, under the premise of further reducing parity storage usage, S-TRAP still maintains high recovery efficiency. For example, for block size 4KB, TRAP takes 1440 seconds to recover data to 30 minutes ago, about 4.3 times longer than ST-CDP or S-TRAP; while TRAP takes 3790 seconds to recover data to 240 minutes ago, about 11.3 times longer than ST-CDP or S-TRAP. Consequently, ST-CDP and S-TRAP have faster and more stable recovery ability.



(a)



(b)

Fig. 12. Recovery time comparison between TRAP, S-TRAP and ST-CDP: (a) Block Size=4KB; (b) Block Size=64KB

7. Conclusions

In this paper, we present a novel block-level CDP architecture, referred to as S-TRAP. In accordance with a certain time interval L, S-TRAP breaks down the parity chain of TRAP and generates new sub-chains. Furthermore, S-TRAP introduces PBC which can reduce the negative impact on primary storage system. We use a simple mathematical model to guide our design in such six aspects as: recovery time, parity storage space usage, reliability, impact on system performance, recovery direction and optimal L. Extensive experiments have been carried out to evaluate S-TRAP. Both mathematical analysis and experimental evaluation have shown that S-TRAP not only has the advantage of high recovery efficiency and reliability, but also further reduces the parity storage usage. Even more important, S-TRAP reduces the negative impacts on primary storage system performance to a large extent.

For future work, we plan to further improve the reliability and recoverability of S-TRAP by ECC (Error Correcting Codes) in sub-chains. We also plan to design parity storage architecture for S-TRAP in order to improve its recording and recovering ability.

Acknowledgment. This research is supported by the National High Technology Research and Development Program (863 Program) of China under the Grant No. 2009AA01A404, and the National Natural Science Foundation of China (NSFC) under the Grant No. 61033007.

References

1. Charles, B.M., Grunwald, D.: Peabody: The time travelling disk. In Proceedings of the 20th IEEE/11th NASA Goddard Conference on Mass Storage Systems and Technologies. San Diego, CA, USA, 241-253. (2003)
2. Chervenak, A., Vellanki, V., Kurmas, Z.: Protecting File Systems: A Survey of Backup Techniques. In Proceedings of the 6th NASA Goddard Conference on Mass Storage Systems and Technologies / 15th IEEE Symposium on Mass Storage Systems, College Park, MD, USA, 17-31. (1998)
3. Cox, L.P., Murray, C.D., Noble, B.D.: Pastiche: making backup cheap and easy. In Proceedings of the 5th Symposium on Operating Systems Design and Implementation, Boston, MA, USA, 285-298. (2002)
4. Damoulakis, J.: Continuous Protection. Storage, Vol. 3, No. 4, 33-39. (2004)
5. Deng, Y.B., Chen, S.G., Hu, W., Gao, F., Liu, C.Y.: A novel block-level continuous data protection system. In Proceedings of the 4th International Conference on New Trends in Information Science and Service Science, Gyeongju, Korea, 242-247. (2010)
6. Duzy, G.: Match Snaps to Apps. Storage, special issue on managing the information that drives the enterprise, 46-52. (2005)
7. Flouris, M.D., Bilas, A.: Clotho: Transparent Data Versioning at the Block I/O Level. In Proceedings of the 12th NASA Goddard / 21st IEEE Conference on Mass Storage Systems and Technologies, College Park, MD, USA, 101-114. (2004)
8. Gifford, D.K., Needham, R.M., Schroeder, M.D.: The Cedar File System. Communications of the ACM, Vol. 31, No. 3, 188-198. (1998)
9. Howard, J.H., Kazar, M.L., Menees, S.G., Nichols, D.A., Satyanarayanan, M., Sidebotham, R.N., West, M.J.: Scale and performance in a distributed file system. In Proceedings of the 11th ACM Symposium on Operating Systems Principles, New York, NY, USA, 1-2. (1987)
10. Hu, Y.M., Yang, Q.: DCD-disk caching disk: a new approach for boosting I/O performance. In Proceedings of the 23rd Annual International Conference on Computer Architecture, Philadelphia, PA, USA, 169-178. (1996)
11. Keeton, K., Santos, C., Beyer, D., Chase, J., Wilkes, J.: Designing for disasters. In Proceedings of the 3rd Usenix Conference on File and Storage Technologies, San Francisco, CA, USA, 59-72. (2004)
12. Kistler, J.J., Satyanarayanan, M.: Disconnected operation in the Coda File System. ACM Transactions on Computer Systems, Vol. 10, No. 1, 3-25. (1992)
13. Laden, G., Ta-Shma, P., Yaffe, E., Factor, M., Fienblit, S.: Architectures for controller based CDP. In Proceedings of the 5th Usenix Conference on File and Storage Technologies, San Jose, CA, USA, 107-121. (2007)
14. Lee, E.K., Thekkath, C.A.: Petal: distributed virtual disks. In Proceedings of the 7th International Conference on Architectural Support for Programming Languages and Operating Systems, Cambridge, MA, USA, 84-92. (1996)
15. Li, X., Xie, C.S., Yang, Q.: Optimal Implementation of Continuous Data Protection (CDP) in Linux Kernel. In Proceedings of the IEEE 2008 International Conference on Networking, Architecture, and Storage, Chongqing, China, 28-35. (2008)
16. Lu, M.H., Lin, S.B., Chiueh, T.: Efficient Logging and Replication Techniques for Comprehensive Data Protection. In Proceedings of the 24th IEEE Conference on Mass Storage Systems and Technologies, San Diego, CA, USA, 171-184. (2007)

17. McKnight, J., Asaro, T., Babineau, B.: Digital Archiving: End-User Survey and Market Forecast. The Enterprise Strategy Group (2006). [Online]. Available: http://www.enterprisestrategygroup.com/2006/03/digital-archiving-end-user-survey-market-forecast-2006-2010/ (current December 2011)
18. Morrey, C.B., III, Grunwald, D.: Peabody: the time travelling disk. In Proceedings of the 20th IEEE / 11th NASA Goddard Conference on Mass Storage Systems and Technologies, San Diego, CA, USA, 241-253. (2003)
19. Patterson H., Manley S., Federwisch M., Hitz D., Kleiman S., Owara S.: SnapMirror(R): file system based asynchronous mirroring for disaster recovery. In Proceedings of the 1st Usenix Conference on File and Storge Technologies, Monterey, CA, USA, 117-129. (2002)
20. Patterson, D.: Availability and Maintainability >> Performance: New Focus for a New Century. Keynote of the 1st Usenix Conference on File and Storge Technologies, Monterey, CA, USA (2002). [Online]. Available : http://www.cs.berkeley.edu/~pattrsn/talks/keynote.html (current December 2011)
21. Peterson, Z., Burns, R.: Ext3cow: a time-shifting file system for regulatory compliance. ACM Transactions on Storage, Vol. 1, No. 2, 190-212. (2005)
22. Rock, M., Poresky, P.: Shorten Your Backup Window. Storage, special issue on managing the information that drives the enterprise, 28-34. (2005)
23. Santry, D.S., Feeley, M.J., Hutchinson, N.C., Veitch, A.C., Carton, R.W., Ofir, J.: Deciding when to forget in the Elephant file system. In Proceedings of the 17th ACM symposium on Operating systems principles, New York, NY, USA, 110-123. (1999)
24. Shaw, S.: Hammerora: Load Testing Oracle Databases with Open Source Tools. Hammerora Project (2004). [Online]. Available: http://hammerora.sourceforge.net (current December 2011)
25. Sheng, Y.H., Wang, D.S., He, J.Y., Ju, D.P.: TH-CDP: An Efficient Block Level Continuous Data Protection System. In Proceedings of the 2009 IEEE International Conference on Networking, Architecture, and Storage, Zhangjiajie, China, 395-404. (2009)
26. The 451 Group: Total Recall: Challenges and Opportunities for the Data Protection Industry. The 451 Group (2006). [Online]. Available: http://www.the451group.com/reports/executive_summary.php?id=218 (current December 2011)
27. Wang, D., Xue, W., Shu, J.W., Shen, M.M.: Fault Tolerance with Virtual Disk Replicas in the Mass Storage Network. Journal of Computer Research and Development, Vol. 43, No. 10, 1849-1854. (2006)
28. Xiang, X.J., Shu J.W., Zheng, W.M.: An efficient fine granularity multi-version file system. Journal of Software, Vol. 20, No. 3, 754-765. (2009)
29. Xiang, X.J., Shu, J.W., Xue, W., Zheng, W.M.: Design and implementation of an efficient multi-version file system. In Proceedings of the 2007 International Conference on Networking, Architecture, and Storage, Guilin, China, 277-278. (2007)
30. Xiao, W.J., Liu, Y., Yang, Q., Ren, J., Xie, C.: Implementation and Performance Evaluation of Two Snapshot Methods on iSCSI Target Storages. In Proceedings of the 14th NASA Goddard / 23rd IEEE Conference on Mass Storage Systems and Technologies, College Park, MD, USA, 101-110. (2006)
31. Xiao, W.J., Yang, Q.: Can We Really Recover Data if Storage Subsystem Fails. In Proceedings of the 28th International Conference on Distributed Computing Systems, Beijing, China, 597-604. (2008)

32. Xiao, W.J., Yang, Q., Ren, J., Xie, C.S., Li, H.Y.: Design and Analysis of Block-Level Snapshots for Data Protection and Recovery. IEEE Transactions on Computers, Vol. 58, No. 12, 1615-1625. (2009)
33. Xiao, W.J., Ren, J., Yang, Q.: A Case for Continuous Data Protection at Block Level in Disk Array Storages. IEEE Transactions on Parallel and Distributed Systems, Vol. 20, No. 6, 898-911. (2009)
34. Yang, Q., Xiao, W.J., Ren, J.: TRAP-Array: A Disk Array Architecture Providing Timely Recovery to Any Point-in-time. In Proceedings of the 33rd International Symposium on Computer Architecture, Boston, MA, USA, 289-300. (2006)
35. Zhu, N.N., Chiueh, T.: Portable and Efficient Continuous Data Protection for Network File Servers. In Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, Edinburgh, UK, 687-697. (2007)

Chao Wang is currently a PhD candidate at the School of Computer Science, Northwestern Polytechnical University, Xi'an, China. His research interests include disaster tolerant, massive storage system and performance evaluation.

Zhanhuai Li is a professor at the School of Computer, Northwestern Polytechnical University, Xi'an, China. He is vice chairman of Database Technical Committee, China Computer Federation. He is a senior member of China Computer Federation. He received his Master's and Ph.D. degrees from Northwestern Polytechnical University in 1987 and 1996 respectively. His current research interests include RFID data management and multimedia real-time database.

Na Hu is currently a Master candidate at the School of Computer Science, Northwestern Polytechnical University, Xi'an, China. Her research interests include data protection and performance evaluation.

Yanming Nie is currently a PhD candidate at the School of Computer Science, Northwestern Polytechnical University, Xi'an, China. His research interests include data stream processing, uncertain data management, RFID data management and software engineering and process.

Received: April 13, 2011; Accepted: December 12, 2011.

DOI 10.2298/CSIS110316068P

A Comparative Evaluation of Effort Estimation Methods in the Software Life Cycle

Jovan Popović[1] and Dragan Bojić[1]

[1] Faculty of Electrical Engineering, University of Belgrade,
Bulevar Kralja Aleksandra 73,
11000 Belgrade, Serbia
{jovan.popovic, bojic}@etf.rs

Abstract. Even though there are a number of software size and effort measurement methods proposed in literature, they are not widely adopted in the practice. According to literature, only 30% of software companies use measurement, mostly as a method for additional validation. In order to determine whether the objective metric approach can give results of the same quality or better than the estimates relying on work breakdown and expert judgment, we have validated several standard functional measurement and analysis methods (IFPUG, NESMA, Mark II, COSMIC, and use case points), on the selected set of small and medium size real-world web based projects at CMMI level 2. Evaluation performed in this paper provides objective justification and guidance for the use of a measurement-based estimation in these kinds of projects.

Keywords: software measurement, effort estimation, comparative analysis, empirical evaluation.

1. Introduction

Estimating size and cost of a software system is one of the biggest challenges in software project management. It is one of the basic activities in the entire software development process, since a project budget, size of a development team, and schedule directly depend on the estimate of the project size and cost. Software teams use many different approaches for estimating effort required for implementing a given set of requirements. Approaches may rely on the experience (as in the expert judgment methods such as Wideband Delphi[1] or Planning Game [2]) or exploit requirement analysis by breaking requirements into elementary work items that can be easily estimated. This research focuses on algorithmic methods that try to quantify systems using certain measurement techniques, and apply algorithms and formulas in order to derive an estimate based on the measured size of the system.

Even though algorithmic methods provide more objective and accurate estimates, many software organizations are reluctant to apply them in practice. According to Hill [3], approximately 60% of the software organizations still use task-based estimates that rely on work breakdown structures (WBS) and expert judgment, 20% combines expert methods with the measurement methods as an additional validation, and only 10% rely solely on measurement methods. Goal of this research is to evaluate reliability of the methods that are most commonly used in practice, by applying them on preselected set of the real world projects.

Current studies [4] show that success rate of software project implementation is very low – only 30% to 35% of all software projects get finished within the planned time and within the budget. One of the most common reasons for such low success rate is an unsuccessful estimation (mostly based on the subjective judgment) and a lack of objectivity. As a result, there is an increasing pressure on software project teams to abandon the experience-based methods for estimating and planning, and to replace them with more objective approaches, such as measurement and analysis based methods. Measurement and analysis is a required process, even for the organizations with the lowest maturity level (level two) according to the CMMI standard [5], which supersedes the older SW-CMM.

Estimate accuracy varies with the phase of the project in which it was determined. Boehm [6, 7] presented this accuracy as so-called “Cone of uncertainty” shown on the Figure 1.

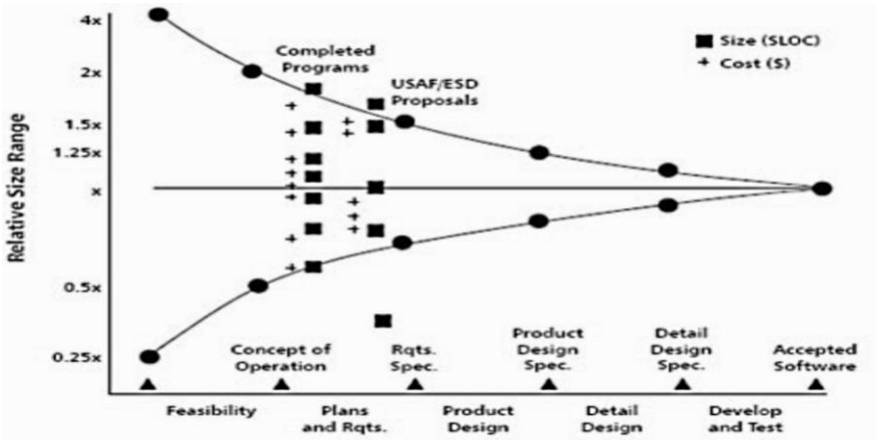


Fig. 1. Cone of uncertainty with effort estimate errors throughout project lifecycle (error varies from 400% at the start of the project and converges to the accurate effort at the end of the project)

Effort estimate determined in earlier phases of the project, such as definition of the initial concepts, might differ up to four times from the final one. This variation can be explained by the fact that initial concepts do not describe the final software system accurate enough. As more details get

defined, the effort estimate converges to the actual value (that can be accurately determined when project is completed). Project Management Institute (PMI) [8] presents similar results about uncertainty, except that their results introduce an asymmetric cone where initial estimates vary between +75% and –25%, budgetary estimate between +25% and -10%, and the final estimate between +10% and -5%. We are using these boundary values as an evaluation criterion for the measurement methods used in the paper. Deviation between estimated and actual effort of the measurement methods used in this research needs to be within Boehm's and PMI boundaries; otherwise, methods should be rejected as inaccurate.

The rest of the paper, presenting the results of our analysis, is organized as follows:

1. Second section contains the problem statement and explains importance of evaluating applicability of the measurement methods in the practice;
2. Third section describes projects used for the analysis, classified measurement methods found in the literature, and methodology of measurement and analysis used in the research;
3. Fourth section describes how the measurements found in the literature were applied on the projects in our data set, how they can be applied in different phases of the project lifecycle, and what deviations exist between the effort estimated using the measurement methods and the real effort values;
4. The last section presents results of evaluation and shows that estimating effort using the measurement methods proposed in the literature has better accuracy than the one reported in practice.

2. Problem Statement

Avoiding the use of measurement and analysis in practice opens an important question – whether the measurement process is applicable in the software projects at all. In other engineering branches, scientific results and laws based on measurement are successfully applied in practice, giving engineering teams a valuable help in managing their own projects. Unfortunately, in software engineering, measurement is still not a preferred approach. According to Hill [3], only 30% of software companies use measurement, mostly as a method for additional validation. In order to determine whether the objective scientific approach can give better results than experience, we have evaluated several well-known measurement and analysis methods, applied them on a set of real-world projects and determined whether they are applicable or not. By obtaining a satisfactory evaluation results, we would prove that measurement techniques could be used for software project estimating, and it would allow us to define a methodology for applying measurement and analysis in practice.

Another important question is whether the software teams should use a single measurement method in the analysis, or they should combine multiple

methods together. In practice, a software team is obliged to give various estimates throughout a project life cycle. At the beginning of the project, some ballpark estimates are needed with minimal engagement and analysis. In the later phases of the project, more accurate and more obliging estimates are needed. Various estimation methods are proposed in the open literature, of varying complexities. Hence, they provide different accuracies. We believe that the most efficient way for estimating a required effort is to use the proper combination of several different measurement methods. By combining the measures together, and applying them in the correct points of time in the project lifecycle, we get a continual estimating process, where the accuracy of an estimate converges to the right value as the project matures. The goal of this evaluation is to derive a methodology for applying the most appropriate measurement techniques during the software project lifecycle, including best practices in using measurement activities.

The evaluation performed in this paper is to provide objective justification and guidelines for using measurement-based estimates in software projects. Research results should prove that software companies could benefit from this methodology and increase the success rate of their projects over the value that can be expected from the current statistical results [4].

3. Measurement Approach

A measurement approach defines what kind of information, techniques, and tools will be used in the analysis. Our measurement approach is specified with the following elements:

1. A project data set gives details about a data collection used in the research. A data collection is a set of the software projects used for applying measurement methods and estimating the effort that is required for the projects to be completed;
2. Description and classification of the existing measurement methods found in the literature, with the detailed analysis of methods that might be applied on the project data set used in the research;
3. An analysis model describes the mathematical models and tools used in the research. A measurement model defines a structure for storing data, as well as techniques that are used in the analysis.

Once the measurement approach is defined, the measurement methods can be applied on the project data set, and the results can be evaluated and compared. The following sections describe concepts important for the measurement approach in more details.

3.1. Project Data Set

In this research, we have used a set of real-world software projects implemented from 2004 to 2010. All projects in the dataset are implemented

by a single company for various clients located in the United Kingdom. Historical company database contains information about 94 projects, categorized by technology and application domain as shown in the figure 2.

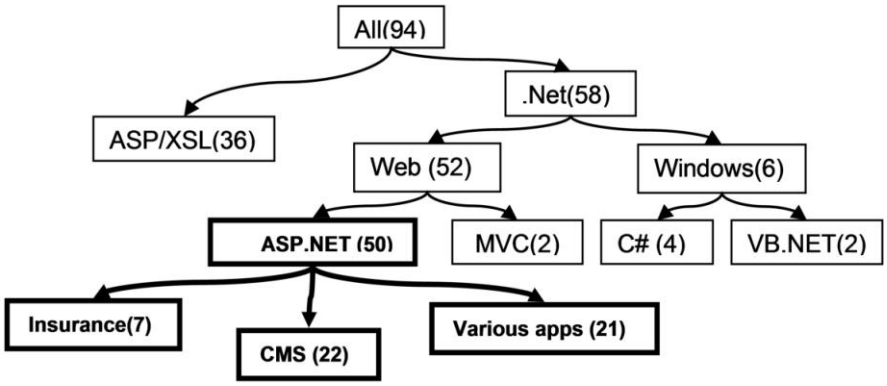


Fig. 2. Classification of the projects from the project dataset by used technology and by business domains. Highlighted projects are used in the analysis

In order to homogenize project dataset used in the analysis, we have excluded 36 ASP projects (OO approach is not applied in these projects and they are poorly documented), and 8 experimental Windows/MVC projects. The remaining 50 ASP.NET applications in the various domains represent set of the project used in the analysis.

Programming languages used in the ASP.NET projects are C#, T-SQL, HTML, and JavaScript. Consistent object-oriented analysis and design methodology was used in all projects in the dataset, and general software development [49] and project management practices [8] were applied. High-level domain models and use cases models were created in the earlier phases of the projects. These models were refined into the more detailed use case scenarios, sequence and entity-relationship diagrams. The core architectural elements were created according to the analysis models in the design phase. In addition, physical design of the system integration was described using the deployment diagrams. Most of the projects in the data set contain only partial technical design documentation. Detailed design documentation is created only for integration of third party components or legacy systems, where off-site teams have to review them before the project start. Only basic technical documents are created for detailed functionalities implemented in the projects, and in most of the cases, they are automatically generated using documentation generation tools. Hence, they do not contain enough information for applying the measurements based on detailed technical documentation.

A company that developed projects in observed data set implemented all practices at CMMI maturity level two. Hence, project management activities (e.g. planning, monitoring and control) were consistently performed across all projects in the data set providing enough information about the effort required

for completing the project. Values of the effort spent on projects are recorded in the project plans, as well as in the invoices sent to customers; therefore, those values can be considered as valid. Various types of documents and specifications describe project parameters, functionalities, lifecycle, and team structure. Table 1 shows how many projects in the data set contain a particular type of documentation.

Table 1. Documentation and lifecycle models available in the projects

Project items/characteristics	Number of projects
Project management documents	
Project plan	46
Estimate/Budget	50
Specification documents	
Vision scope and domain models	24
Requirement document	9
Use case models/scenarios	27
Database models	30
Functional specification	23
Analysis model	14
Software architecture document	7
Detailed UML design	3
Lifecycle model	
Phased(UP/MSF)	23
Agile(XP/Scrum)	7
Project characteristics/constraints	
Team structure description	4
Post mortem analysis	5

Total amount of 30 projects, with enough technical documentation aligned with project plans where we can apply several different measurement methods, were selected for the final dataset. Statistical information about the effort, duration, team size, and experience of the project team members are shown in the table 2. The final set of the projects that is used in the analysis is homogenous by the team structure, with identical technical parameters (e.g. platform, programing languages, and complexity).

Table 2. Project statistical parameters

	Effort (Person-month)	Duration (Months)	Team size (Person)	Experience (Years)
Minimum	2	2	4	1
First quartile	4	4	6	4
Median	7	5	7	5
Third quartile	12	7	8	6
Maximum	15	12	10	8

3.2. Metrics Used in Research

One of the important tasks in the research was to select measures and metrics to be used for defining type of information that should be collected, as well as shaping the entire analysis process. There are many software size measurement approaches present in the literature. Metrics can be categorized as:

1. Metrics based on the source code such as a number of source lines of code [9], Halstead's [14] or McCabe's [15] complexity metrics;
2. Functional metrics such as IFPUG [10], NESMA [16], Mark II [17] or COSMIC [12] methods. These metrics are widely used in the software industry and they are standardized as ISO/IEC standards [13].
3. Object oriented metrics such as use case points [11], class points [18] or object oriented design function points [19].
4. Web metrics – a set of methods specialized for web application development. The first papers in this field were Reifer's web objects [21], followed by the work of Mendes[34, 35], and Ferrucci [36],
5. Other techniques and enhancements such as Galea's 3D function points [20], Fetche's [22], Uemura's [23], or Buglione [40] approaches.

We examined project documentation shown in the table 1 in order to determine which measurement methods can be applied in the project data set. Results are shown in the table 3.

Table 3. Number of projects that can be used for the measurement methods

Measurement methods	Number of projects
Source code based	0
Functional methods	
FPA/NESMA	23
NESMA Estimated/Indicative	30
Mark II	30
COSMIC	21
Object oriented methods	
Use case point	27
Class points/OOAD points	6
Web methods	N/A
Other methods	N/A

Source code based metrics are not applied in the project data set. There are at least four different languages applied in each project, without the information what effort was required to implement pieces of code written in different language. Hence, it is questionable whether these metrics can be successfully applied in the project data set. In addition, as a significant (but unknown) amount of source code is automatically generated, it is impossible to isolate the code that is directly created by the development team and to measure just a size of that code.

All projects have enough functional documentation; therefore, it was suitable to apply measures in the functional group (IFPUG, NESMA, Mark II, and COSMIC). These measurements are widely used in the practice and certified as ISO/IEC standards [13] indicating that they should give satisfactory results.

The only object-oriented measure that is applied is a use case point metric. Functional specification has defined use case scenarios and models; hence, there is enough information for applying this metric successfully. The other object-oriented metrics are not applied in this research, as there is no sufficient information in the technical documentation. Therefore, applying these techniques on inadequate data will cause too many errors.

We have not applied web metrics in the research, although the projects in our dataset are web applications for several reasons. Application framework used in the projects (ASP.NET) encapsulates most of the web related code, and generates most of the HTML/JavaScript code automatically. Hence, there are many parameters used in the web methods that hidden from the development team in the framework, and therefore these parameters do not affects the effort. There are only few projects in the data set where web aspects of the application development are not completely encapsulated; however, there are not enough projects for the analysis.

The other nonstandard methods [20], [22], [23] were not applied, since each of them requires some specific information. In the existing documentation, we have not found enough information required for these methods; and our decision was to avoid any change or producing new documentation.

Most of these metric use both functional and nonfunctional parameters of the system when a final size is determined. Usually, two different sizes are created, the one that depends on the functional parameters, which we call unadjusted size; and another that is determined by adjusting an unadjusted size using nonfunctional parameters, which we call adjusted size.

As an example, there are 17 cost drivers and 5 scale factors used in COCOMO II [2] in order to adjust the functional size of the system. Each factor takes one of the rates (very low, low, nominal, high, very high, and extra high) and appropriate numeric weight. We have evaluated applicability of COCOMO II drivers in the projects from our data set, and we have found that most of them are too unreliable or undocumented. COCOMO II factors evaluated in our project data set are:

1. Product factors – required reliability (RELY), database size (DATA), and complexity (CPLX) might be considered as nominal according to the subjective judgment of team members. Documentation (DOCU) and reusability (RUSE) depends on the budget and project timelines; however, they are not documented in most of the projects.
2. Platform factors such as execution time (EXEC), storage (STOR), and platform volatility (PVOL) might be considered as nominal across all projects.

3. Personnel factors – there are no records about the team capability (PCAP and ACAP factors), experience (APEX, PLEX, and LTEX factors), nor about the personnel continuity (PCON factor).
4. Project factors – usage of software tools (TOOL), multisite development (SITE) and schedule (SCED) varies on the project but there are no organization-wide standard for assessing impact of these factors.
5. Scale factors process maturity (PMAT), team cohesion (TEAM) and development flexibility (FLEX) are nominal; however, there are no information about precedentedness (PREC) and architecture/risk resolution (RESL) factors.

Due to the fact that most of the non-functional parameters are either undocumented or depend on the subjective opinion of team members, only functional measurements were considered, without using any non-functional parameters of the system for adjustments. We are confident in the validity of the functional size metrics defined in this section because they are derived from the objective sources (e.g. documentation and prototypes). Combining these objective metrics with the factors that are either undocumented or subjective would affect objectivity of the results due to the high uncertainty of the non-functional parameters.

In this paper, the term “size” is equivalent to the term “unadjusted size” in the original measurement techniques.

3.3. Analysis Model

This section describes the model used for analysis of the projects in our dataset. The analysis model is represented through following components:

1. Data model that is used in the research where we have defined data structure used to store project data, and mathematical models used for analysis,
2. Prediction model where we have defined functional dependency between the size and effort,
3. Validation model where we have described methodology used to validate measurement/prediction methods used in the research.

Data Model

We have complete access to project characteristics and documentation for all projects in the dataset; however, these characteristics have to be organized so that we can easily use them in various measurement methods. Project characteristics required for all measurement techniques are collected from project documentation and grouped in tuples containing information about project name, effort, and all project characteristics required by selected measurement techniques. These characteristics are used to establish a uniform tuple space that represents a repository for all project measurements, and it is used as a base source of information for all selected measurement techniques in our research, as shown on the figure 3.

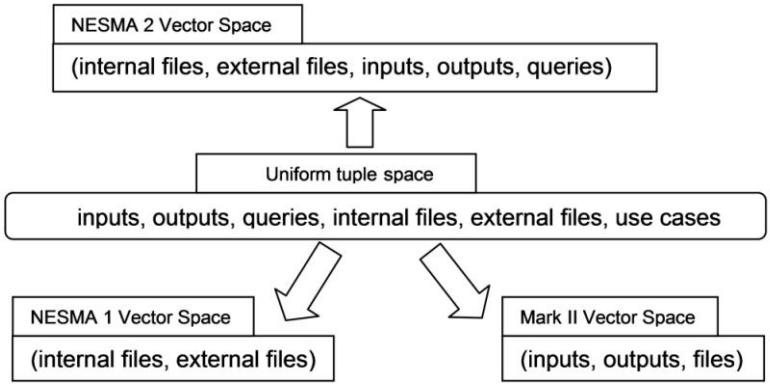


Fig. 3. Uniform tuple space with derived vector spaces specific for particular measurement techniques. Uniform tuple space contains information about all project characteristics. When vector spaces for the specific measurement techniques are needed, only attributes required for particular measurement are derived and projected into a new vector space that is used in measurement

In order to apply a particular measurement method to the given data set, specific information must be extracted from the uniform tuple space. Extraction of required project characteristics is done by restricting attributes in the tuple space, which creates a vector derived for a particular measurement technique. Example of deriving vector spaces for Mark II, NESMA indicative and NESMA estimated measurement techniques from the uniform tuple space is shown on the figure 3.

The uniform tuple space and derived vector spaces described in this section represent a flexible structure that can be used for applying a uniform measurement approach on various metrics. The model of the data structure is shown on the figure 4.

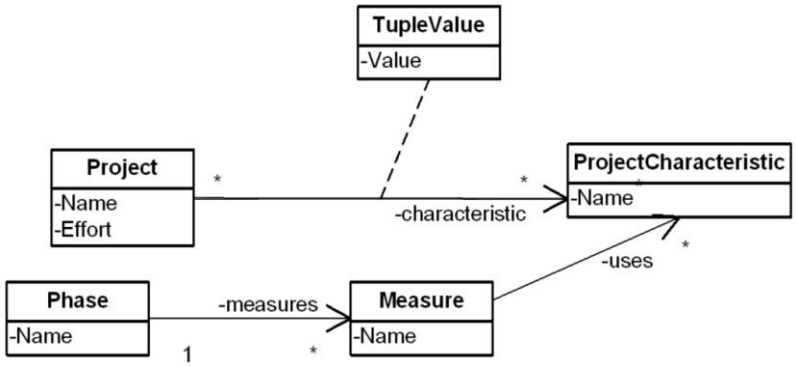


Fig. 4. Data model for the structure of the projects and their measured data used in the research

Each project in the data set has a name, effort, and set of project characteristics (e.g. number of use cases, inputs, or files). Each characteristic has an associated value in the tuple. Measures use some of the project characteristics to calculate the size of the system. Measures are grouped by phases of the project (this is described in more details in the section 4). This data model can be converted to any data structure for storing data e.g. in database relational model, or even in the flat MS Excel spreadsheets.

All measurement methods used in the research calculate size as a weighting sum of project characteristics, as shown in the formula 1.

$$Size = \sum_{i=0}^{N} w_i * x_i \quad (1)$$

In the formula 1, N is a number of characteristics that are used for some particular measurement, w_i are constants, and x_i represents tuple values of interest for some particular measurement method. For example, NESMA indicative method uses two characteristics (N=2) where x_0 is a number of internal files, and x_1 is a number of external files. Values of the weighting factors w_i are specific for the particular measurement methods – reader can find more details about the specific rules and values of weighting factors in the section four, which describes ways of applying these measurement methods.

Formally, we are using different form of equation 1, represented as a scalar product shown in the formula 2.

$$||x|| = x * w . \quad (2)$$

In the formula 2, x is a vector that represents the project characteristics, and w is a weighting vector with the same number of dimensions as other vectors in the vector space. Vector w has constant value for each of the measurement technique. Scalar value ||x|| is the norm of the vector, and it represents measured size of the project from the formula 1. Applying formula 2 on the vectors derived from the tuple space, we are defining a normed vector space, where a norm represents the actual size of the project.

Prediction Model

In order to evaluate a correlation between measured size and actual effort, it is necessary to hypothesize a functional model that describes the dependency between these two values. Such functional model would help us to predict an effort using the size, and compare predicted value with actual.

The most commonly used functional model is a linear dependency assuming that an effort needed to complete a project is directly proportional to the project size as it is shown in the formula 3.

$$Effort = PDR * Size . \quad (3)$$

Effort is a predicted value, Size is measured size for the project calculated using formula 1 or 2, and PDR is a Project Delivery Rate expressed as hours

per unit of size [3]. Linear models are one of the earliest functional dependencies developed in the theory of effort estimation. They were used by Albrecht [24], then by Nelson [25], and finally Briand in COBRA [26] model.

The other group of models is a group of nonlinear models (an effort required for implementation is not directly proportional to the size). One of the first nonlinear models was created by Kemerer [27] who used polynomial dependency between the size and effort, and it has better accuracy than linear Albrecht's model [24]. Currently, the most common form of nonlinear models is a power function model defined in the formula 4.

$$\text{Effort} = K\ \text{Size}^{E}. \tag{4}$$

Linear coefficient K and exponent E are either constants or being calculated using the project constraints such as product complexity, delivery time, or team capability. Currently, the most common power function models are Boehm's COCOMO II and Putnam's SLIM model [28]. Other nonlinear models such as Walston-Felix [29] or Beily-Basily [30] models are not used in the practice. In the power models, a logarithm of effort (instead of effort) is directly proportional to the size of the system.

Different methods can be used to build a model for predicting the project effort based on the size. The most commonly used method is a regression model; however, there are lot of case studies where prediction models are significantly improved using the Fuzzy logic [37], Bayesian networks[38], Neural networks[39] etc. These methods are very effective when there are a large number of different parameters in the model that should be analyzed, or when models are too complex to be represented as a regular function.

In the projects in our dataset, we have only one independent variable – the size of the project that should be used to predict an effort. Organization that has developed projects in the dataset has CMMI maturity level two, where are implemented "Measurement and analysis" and "Project planning" practices. Therefore, we are convinced that size of the projects and efforts spent on the projects are valid parameters. Non-functional parameters, such as product complexity, analyst capability, or data complexity, are either undocumented, or we are not convinced that they are valid. Therefore, organization and process parameters are not used in the research.

The method we have chosen to build our prediction model is a linear regression technique. Due to the fact that we have only one independent value, and that functional dependencies are defined using the formulas 2 and 3 this is the most appropriate method in our data sets. In the linear regression model is the most common regression model, where estimated effort can be expressed as a linear dependency of the calibration constants and measured size, as shown in the formula 5.

$$Effort^{*}(Size) = k * Size + n. \tag{5}$$

In the formula, $Effort^*$ is a predicted value of the effort, parameter k and parameter n are constants. Ideally, k should be close to the PDR value given in formula 3.

As neither linear nor power model is favored in the analysis, the same regression formula is applied in both models. During the evaluation of linear model, effort was used as a predicted value, while in power models, logarithm of project effort is used instead of the effort.

The linear regression is just a basic model for a prediction. Many other techniques such as neural networks or case-based reasoning can be used instead of the regression [41]. However, most of these methods are more suitable in the cases where we have many different independent values that should be analyzed. As described above, we cannot be convinced that organizational parameters are valid in the observed project data set. Therefore, as we do not have a broad range of available independent variables, advanced methods are not applied in our data set. In the paper will be shown that companies on the maturity level two are limited to the basic prediction methods, and that if they want to use more accurate and advanced models, they will need to improve their maturity level to the CMMI level three at least.

Validation Model

In order to find metrics that are most suitable for the effort estimation, we have evaluated correlation between measured sizes and effort values. Pearson's product-moment correlation factor is used as a criterion for evaluation of applicability of measures. If projects in the data set are represented as pairs of size and effort such as ($Size_i$, $Effort_i$), correlation between two sets of data is given in the formula 6.

$$R = \frac{\sum_{i=1}^{N}(Size_i - \overline{Size})(Effort_i - \overline{Effort})}{\sqrt{\sum_{i=1}^{N}(Size_i - \overline{Size})^2}\sqrt{\sum_{i=1}^{N}(Effort_i - \overline{Effort})^2}} \quad . \tag{6}$$

The value of the correlation coefficient R varies from 0 to 1, and values closer to 1 indicate that there is a better correlation between the sizes and efforts. Sizes that are highly correlated to the effort can be applied as valid measures of the system.

In addition, we have evaluated how well the predicted effort, calculated using the formula 5, is close to the actual effort. The actual effort is the sum of predicted value and prediction error as shown in formula 7.

$$Effort(Size) = Effort^*(Size) + \varepsilon \quad . \tag{7}$$

Estimation function $Effort^*(Size)$ (calculated using the formula 5) is determined so that the prediction error ε is minimized. In that case, estimated value of the effort is very close to the real values. Instead of the prediction

error, we are using a mean relative error (MRE) [43] for evaluating the quality of the estimation model.

Mean relative error, or magnitude of relative error represents the ratio of the difference between the estimated and the real value ε, and the actual value itself, as shown in the formula 8.

$$MRE_i = \frac{\left| Effort_i^*(Size) - Effort_i(Size) \right|}{Effort_i(Size)} \ . \tag{8}$$

In order to evaluate the accuracy of the estimating model, mean and maximum values of magnitude of relative error are determined as well. Mean magnitude of relative error (MMRE) and maximum magnitude of relative error (MRE_{max}) are derived from the individual MRE values using formulas 9 and 10.

$$MMRE = \frac{1}{N}\sum_{i-1}^{N} MRE_i \ . \tag{9}$$

$$MRE_{\max} = \max_{i\in[1..N]}(MRE_i) \ . \tag{10}$$

MMRE indicates accuracy of the estimating model, while MRE_{max} parameter reveals extreme cases. Values within the error range reported by Boehm and PMI prove that measurement technique can be successfully applied in the practice.

4. Applying Measurement During Project Lifecycle

Once the preparation for measurement is finished i.e. data set, measures, and model are defined, selected measurement techniques can be applied in order to evaluate their accuracy.

One of the goals of this research was to evaluate whether the measurement techniques selected from the literature can be used for estimating an effort during the project life cycle. We use the standard phases from Unified Process (UP) [31], as a lifecycle model of the software development process:

1. Inception – where a scope of the system is agreed upon, and requirements are described at the highest possible level;
2. Elaboration – where major analysis activities are done, and requirements, architecture and a general design are specified;
3. Construction – where detailed design is completed, code is created and tested according to the requirements and design;
4. Transition – where a solution is transferred to end users.

This phased model is applicable to all projects in the data set, even if some of them were not implemented according to the strict UP model (e.g.

waterfall or MFS [32]). UP phase model can be applied even on the iterative models such as Extreme Programming [2] or Scrum [33], where iterations can be divided into smaller phases mentioned above.

Documentation available in the projects within the project data set, can be categorized by the phase in which they were created. For example, a vision/scope document, statement of work, and project brief document, which are created at the start of a project, are categorized as inception phase documents. List of the available documents and creation phases is shown in the table 4.

Table 4. Documents and models available in various phases of the software development process

Project Phase	Available Documents
Inception	Project Brief/Proposal
	Vision/Scope Document
	Conceptual Domain Model
Elaboration	Use Case Model and Scenarios
	User Interface Specification
	Logical Model
	Functional Specification
	Physical Model
Construction	UML Class Diagrams
	System Sequence Diagrams
Transition	Training Material

4.1. Measurement During Inception Phase

In the earlier phases of projects such as Inception phase in the UP [31] model, or Envisioning phase in the MSF [32] model, only high-level requirements are defined. Table 4 shows that vision/scope document, statement of work, and project brief document are created during the inception phase. These documents contain information about scope of the project, users who will use the system, their needs, high-level features of the system, and domain models. From the set of available metrics, the following ones can be applied in the inception phase:

1. NESMA indicative metric – domain model which contains business concepts and their description is available, metric of the system based on the number of internal and external objects can be determined for each project;
2. Number of use cases – use cases can be identified from the vision/scope document, and their count can be used as a measure of the system size. Farahneh has used the same metric in his research [42].

NESMA indicative method approximates the standard NESMA functional point method, where size of the system is determined by using just basic

functional information. The logical groups of pieces of information (files in the functional point measurement terminology) that represent data structures are identified in the system. Structure of files (relationships between files and information they contain) are not relevant. The only analysis that should be done is categorization of files as internal and external, depending on the fact whether they are maintained within the system or just referenced from the other systems. In this case, NESMA indicative metric uses vectors containing the number of internal/external files in the form (ILF, EIF). Size of the system is determined by deriving norm of the vector using a scalar product between weighting vector (35, 15) and the size vector.

Number of use cases approximates the standard use case point metric, since the use cases are just enumerated without any deeper analysis related to their complexity. The vision/scope document, which is created during inception phase, contains high-level features of the system without their detailed descriptions. Hence, use cases can be identified upon these features, although detailed scenarios are still not defined. The number of total use cases to be implemented in the system can be used as a measure of the system functionalities. This number is pure scalar metric; it has only one dimension that represents size of the system.

Table 5. Measurement applied during inception phase with their correlation with effort (R), Mean magnitude of relative error, and Maximal magnitude of relative error for both linear and nonlinear (power) models

	Linear model			Nonlinear model		
Metric	R	MMRE	MRE_{max}	R	MMRE	MRE_{max}
NESMA Indicative	93%	16%	46%	85%	20%	69%
Number of Use Cases	58%	46%	126%	59%	40%	94%

These two metrics are applied on the projects in our data set to determine the size of each project expressed in NESMA function points and number of use cases. Correlation between the size and actual effort was analyzed and results of the analysis are shown in the table 5.

The NESMA indicative method shows significantly better results than a number of use cases as a size metric. Number of use cases does not show a true nature of the system, since beneath each use case frequently lie several hidden functionalities that might significantly increase the effort required for implementation of the use case. Hence, there is a significant variation between the effort and number of use cases. Farahneh [42] got similar results when he applied the number of use cases as a measurement. Accuracy in his case study was between 43% and 53%; however, this is still not even close to the accuracy of the NESMA indicative method.

The major issue with counting use cases is the absence of any standard that precisely defines what a use case is. Use cases in our dataset differ from project to project; they might be either merged or decomposed to several use

cases. There is no strict rule defining when a set of user actions should be placed within one use case scenario, nor when they should be decomposed into set of different use cases (e.g. included or extended sub use cases). Granularity of use cases significantly affects a number of use cases, and therefore size of the system. We believe that inconsistency in the style causes large deviations in the number of use cases as a size metric.

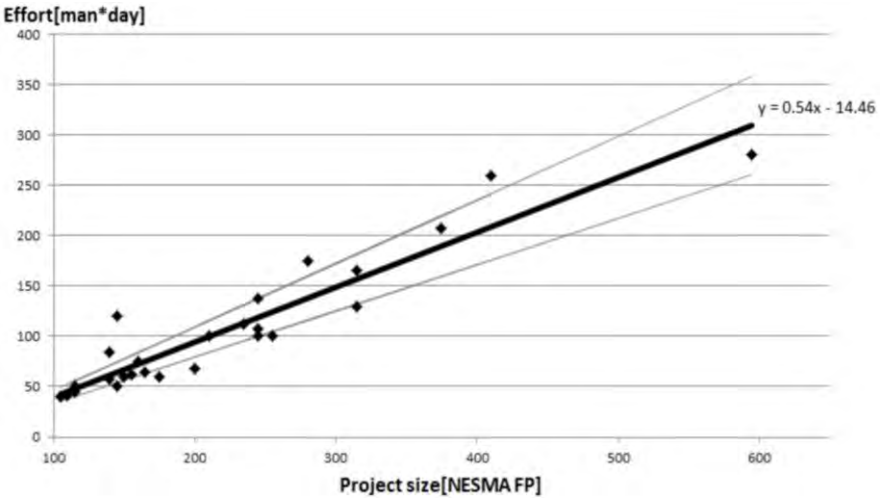


Fig. 5. Dependency between the effort and size measured in functional points using NESMA indicative method. There is a good linear dependency where most of the real effort values are near to the estimated values

Having in mind an absence of detailed information about the system functionalities, NESMA Indicative metric has a good accuracy that is acceptable in the practice. The inception phase milestone (Scope/Vision completed) is equivalent to the "Concept of Operation" milestone in Boehm's report, or to "Project initiation phase" in the PMI terminology, because in these phases we have just a rough description of system features. Comparing these results with the results reported by Boehm's and PMI it might be noticed that estimating software size using NESMA indicative method gives better results than the ones that can be found in practice. Using Boehm's results, estimating error in Concept of Operation milestone can be up to 100%, and according to PMI accuracy varies from -25% to +75%. Therefore, the mean relative error of 16% is good, and even the maximal error of 46% is satisfactory.

Dependency between sizes of projects measured as NESMA Indicative function points and actual effort applied on the data set used in the research is shown on the figure 5.

Applying NESMA indicative functional point method in the inception phase of the project enables project team to estimate the effort with accuracy that is significantly better than the results found in the practice. Therefore, using

measurement method in this phase can increase accuracy of project estimate.

4.2. Measurement During Elaboration Phase

During the elaboration phase of the project analysis and design are performed. Those activities are necessary for translating user requirements into the detailed specification. Elaboration phase can be divided into several separate iterations. In the earlier iterations, focus is still on the user needs, requirements and analysis, while in the later iterations focus is on the detailed design, technical solution and full functional specification.

In earlier iterations of the elaboration phase, detailed user requirements are collected; scenarios of usage are documented via use case scenarios; and user interface and logical model of data are created.

As elaboration phase continues, more documentation and design with details and technical requirements are created. Detailed design allows project team to apply more detailed and more accurate metrics. In later iterations of elaboration phase, detailed functional requirements with a description of all fields, data structures, rules, and processes that should be implemented in the system are created. These iterations may include creation of detailed database models with all necessary fields and relationships so that the project team can start with development. Several metrics can be used in elaboration phase:

1. Use case points – a method that is used to measure size of the system by analyzing a complexity of use case scenarios;
2. NESMA Estimated method – an approximation of the standard measurement technique based on functionalities of the system;
3. Mark II method – a method used to measure a functional size of the system, mostly applied in United Kingdom;
4. Functional point analysis (FPA) method – a method where detailed functionalities of the system are measured.

First three metrics can be applied in earlier iterations, while the fourth metric can be used when elaboration is near to end. In the rest of this section, we will separately discuss metrics that can be applied in earlier and later iterations of the elaboration phase, as well as the accuracy of these methods.

Metrics Used in the Earlier Iterations of Elaboration Phase

Detailed use case scenarios are described in earlier iterations of elaboration; hence, there is enough data to determine the exact measure of use case complexity expressed in use case points. The use cases are categorized as low, average, and high, depending on the number of transactions within the use case scenarios. They are combined with three classes of user complexity (low, average, high) forming six-dimensional vector (UC_H, UC_A, UC_L, A_H, A_A, A_L) that is used to measure size of the system. The norm of the vector is

determined using a scalar product between the use case point size vector and the weighting vector (15, 10, 5, 3, 2, 1).

Functionalities and basic concepts of the system are also defined. Hence, some estimated metrics based on functionalities such as NESMA Estimated and Mark II method can be applied.

NESMA Estimated method approximates the standard NESMA method for measuring the functional size. Size of the system is determined by identifying objects and processes in the documentation. Objects in the system are classified as internal or external files, depending on the fact whether they are maintained within the system or just referenced from other systems. Functional processes are classified as inputs, outputs, and queries. Total number of internal files, external files, inputs, outputs, and queries forms a five dimensional vector (ILF, EIF, EI, EO, EQ), which is used to determine size of the system. The norm of the vector is determined as a scalar product of the size vector and the weighting vector (7, 5, 4, 5, 4).

Mark II method is a modification of standard functional point method where information about the number of input processes, output processes and objects within the system are taken into consideration. In the Mark II method, a three-dimensional vector of inputs, outputs (exists), and objects (N_i, N_e, N_o) is used for determining size of the system. The norm of the vector is determined by using a scalar product between the weighting vector (0.58, 1.66, 0.29) and the vector itself.

Metrics Used in the Later Iterations of Elaboration Phase

Detailed functional point metric (either standard IFPUG or very similar detailed NESMA method) can be applied in the later iterations of elaboration phase. Functional point analysis is done using detailed description of user interface such as UI design or process flows, and informational models such as Entity-Relationship diagrams or class diagrams.

In the functional point analysis five system parameters are analyzed – internal logical files (ILF), external interface files (EIF), inputs in the system (EI), outputs from the system (EO), and queries (EQ). However, these elements are not simply counted – they are also categorized as low, average, and high, depending on their complexity. Rules for classification depend on the number of data elements and files that are affected by the transaction for the inputs, outputs, and queries; and number of data and composite structures for files [10]. The number of elements in each class represents a separate dimension of the system, forming a vector with fifteen dimensions**.** The norm of the vector is computed using a scalar product between the vector that represents system and the weighting vector.

Accuracy of Measurements Used in the Elaboration Phase

Results of applying the measurement methods described in the previous sections on the project data set are shown in the table 6.

Early iterations in the elaboration phase are equivalent to the "Requirement specification" phase in the Boehm's cone of uncertainty, or "Budgetary estimate" point in the PMI model, since in this phase there is enough information to create a valid budget of the project. Estimating error is up to 50% in the requirement specification phase by Boehm, and between -10% and +25% by PMI results. Therefore, mean error of 10-13%, achieved using NESMA estimated or Mark II methods is more than acceptable. Even the maximal error of 30-65% is within the acceptable range according to the Boehm's results.

Table 6. Accuracy of measurements applied during elaboration phase

	Linear model			Nonlinear model		
Metric	R	MMRE	MRE_{max}	R	MMRE	MRE_{max}
Use Case Points	80%	31%	94%	74%	29%	74%
NESMA estimated	92%	13%	65%	93%	16%	34%
Mark II	92%	10%	30%	93%	15%	35%
FPA	92%	10%	22%	96%	12%	29%

Correlation between the project effort and a use case point size is significantly lower, and mean relative error of 31% is outside of the acceptable range of errors. The accuracy of the UCP method varies in the literature. In three case studies [45, 46, 47] Anda has reported that MMRE is between 17% and 30%, and Lavazza [49] got MMRE of 29.5%. Hence, this metric is not good enough, at least not in the data set used in the research. In the projects from our data set, we have found a variety of different styles of use cases, which might cause a high prediction error. Two major issues we have noticed in documentation (that affect use case point metric accuracy) are:

1. Decomposition of use cases – there is no strict rule that defines when a set of user actions should be placed within one use case scenario, and when they should be decomposed into set of different use cases (e.g. included or extended sub use cases). Granularity of use cases significantly affects a number of use cases, and therefore a measured size;
2. Level of details – some use cases contain just an essential functional description, while the other have many details. Number of details placed in the use case scenario affects a number of transactions and measured complexity of the particular use cases. Hence, size expressed in the use case points varies although functionality is the same.

We believe that these issues cause a low accuracy of the UCP method in our data set. Different data sets/studies might give better results if use cases styles are standardized. However, we have decided not to alter, standardize or "improve" quality of documentation in order to get real values of accuracy.

Later iterations of the elaboration phase in the Unified Process are equivalent to the Boehm's "Product design" phase where estimating error is

around 25%, or final estimate in the PMI model where the error is from -5% to +10%. Complete functional point analysis metric with mean error of 10% and maximal error of 22% is within the Boehm's range, but not better than the error ranges reported by PMI, especially if a maximal error is considered. However, the FPA method cannot be held responsible for a lack of accuracy when it is compared with PMI results. This research analyzes dependency between the functional sizes and efforts without considering nontechnical factors. Therefore, it is reasonable to assume that adjusting the size examined in this research using the same nontechnical factors would increase the accuracy of estimate and align it with the results found in the practice. A result of applying FPA metric on the projects in the dataset is shown on the figure 6.

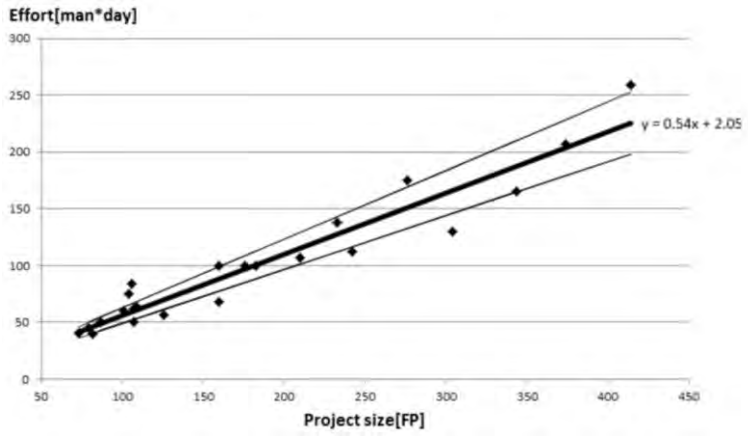


Fig. 6. Project sizes (expressed in FPA functional points) versus actual project effort. Most of the projects are placed around the regression line and within the MRE area bounded with the two thin lines

4.3. Measurement During Construction Phase

Focus of the measurement and analysis in the construction phase is not on the estimating size of the system. The main objective of the project management activities in the construction phase is project monitoring and control. Therefore, the focus of the measurement process is moved from the system as a whole to the individual pieces of the system (project tasks). Size of the entire system is measured in the previous phases; effort required to implement the software system is already determined; and budgets/plans for construction are developed according to the estimates. Development team needs to control a development process, compare a progress with the plan, and take the corrective actions if there are any significant deviations between the actual work and planned work. Even in the agile project approaches

(where planning is done in the iterative manner) the most important task is to control performances of a software team. Therefore, a main goal of the measurement is estimating a size of the tasks that were planned in the iterations. In order to determine size of the particular functionalities we can apply two measurement methods in the construction phase:

1. Functional point analysis method – same technique used to determine size of the system method can be applied on the particular tasks as well;
2. COSMIC Full Functional Point method – latest method in functional metric family can be applied on all functionalities that have defined communication protocols.

The other methods presented in the paper are not suitable for measuring particular tasks. NESMA indicative and estimated methods are just approximations of the standard functional point method. Hence, standard NESMA or IFPUG methods have better results. Mark II measure is too robust because it only counts functional elements. Use case point method might be applied on the particular tasks under the assumption that use cases match project tasks. However, for projects in our data set, most of the project tasks represent parts of the use cases. Hence, use case point method cannot be applied on most of the projects in the dataset.

Standard functional point analysis can be applied on the particular tasks in the same way as on the entire system. The only difference is that functional sizes are not summed up in order to determine the size of the system. Measured sizes are associated to the particular tasks that are classified as inputs, outputs or enquiries and rated as low, average or high complexity tasks (based on the specification). The size is expressed as a scalar measure representing complexity of the particular functionalities.

Functional dependency between the effort required to complete tasks and corresponding sizes is shown in figure 7.

Size of transactions is between three and seven functional points (FP), depending on the type and complexity of the functionality. In our dataset, efforts of particular tasks are in the range from one hour to 2.5 days.

For medium sized tasks, one can establish some dependency between size and time. However, this method is not applicable on the tasks that are either too small or extremely long. Projects in our data set contain tasks in broad ranges of tasks from 1-2 hours, including minor queries or service calls, to the extremely complex reports that fell out of the medium range of tasks.

COSMIC method is the latest method from the functional point family, which should overcome drawbacks of the existing functional point methods. While using this method, we measured all communication messages that are exchanged between user of the system and system components.

The prerequisite for applying COSMIC method is an existence of detailed communication specification or sequence diagrams that define how messages between users, system components, and data modules are exchanged. In the COSMIC method, there are four types of messages that are identified – entry messages (E) where information is entered in the system, exit messages (X) where information is taken from system

components, write messages (W) used to store information in the persistent storage, and read messages (R) used for taking information back from the persistent storage. Size of the functionality is calculated as a total amount of all messages exchanged between user and system components within that functionality. Maximum size of functionalities is not limited, since there is no upper boundary regarding the number of messages that can be exchanged within the functionality. Dependency between the task sizes and efforts is shown in the figure 8.

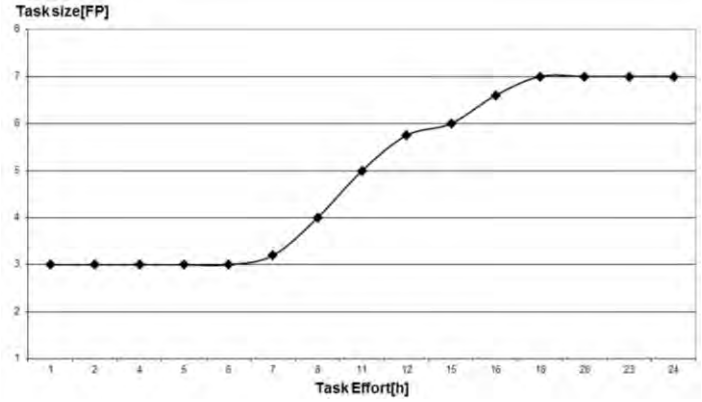


Fig. 7. Relationship between efforts required to implement particular tasks and task size expressed in functional points. For tasks that need between 7 and 17 hours to complete, a dependency is monotone; however, for tasks outside this range there are two saturation areas

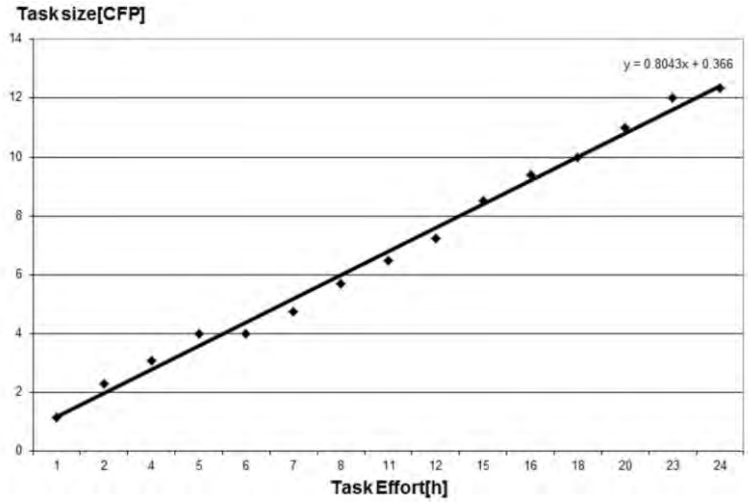


Fig. 8. Relationship between size of functionalities expressed in COSMIC points (CFP) and required effort recorded in hours. Applied linear regression returns good estimating function where actual values are close to the real ones

COSMIC metric does not have the same kind of constraints as FPA, since the number of messages that are exchanged between user and system components can vary from one to infinity. Hence, there is no saturation area in the figure 8. Deviation between the real task effort and the estimated one is less than a half of an hour, representing a very good accuracy with a mean error of 8%. Applying linear regression on the data set, it can be determined that there is a good functional dependency between the mean development time for tasks and size expressed in COSMIC functional points (CFP). Knowing that Boehm's results during the "detailed specification definition" milestone give error equal to 10%, this result is also acceptable.

Accuracies of the FPA and COSMIC methods are shown in the table 7. Correlation and error of the FPA method are determined only in the linear area.

Table 7. Compared accuracies of the FPA and COSMIC methods

	R	MMRE	MRE_{max}
FPA	95%	10%	14%
COSMIC	97%	8%	11%

COSMIC method is highly correlated with the actual effort of the tasks with a very good accuracy. FPA method also has a good correlation if it is applied on medium sized tasks. However, this method cannot be successfully applied on the tasks outside the linear area.

5. Conclusion

Results presented in the paper are empirical evidence that measurement and analysis can be successfully applied in the practice. Our evaluation of measurement methods on the real projects shows that there is no objective reason for using a subjective judgment in the process of project effort estimation, because the measurement based methods are accurate enough. These results are explained in more details in the following section.

5.1. Results

We have evaluated the most common measurement techniques and applied them on the real projects to estimate the effort required for the project. Estimated values of efforts are determined using regression techniques based on the measured sizes. We have compared estimated effort with the real effort and determined the prediction error. Furthermore, correlation between the measured size and the actual effort are determined in order to evaluate whether the measures can be applied in the effort estimation.

We have determined a mean and maximum magnitude of relative prediction error for all measurement methods used in this research. These criteria are used to compare and evaluate accuracies of the estimating models. Evaluation methods that use the pure MMRE criterion were criticized in [44]. Therefore, we have determined distribution of the mean error for all methods including minimum value, median value, and the range where are placed 50% of the errors. Figure 9 shows compared distributions of the method accuracies applied in the UP project phases.

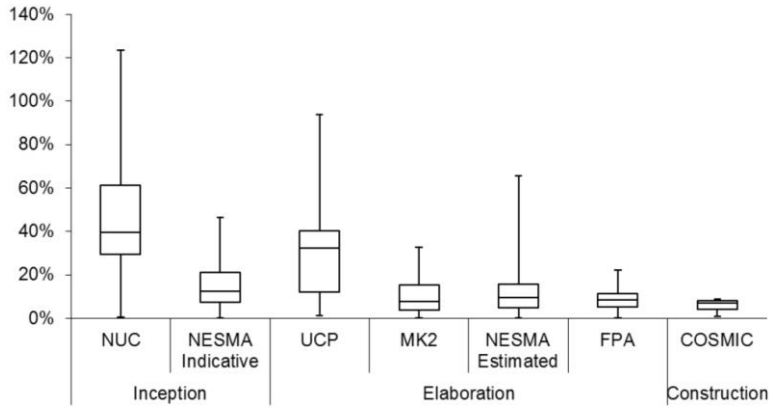


Fig. 9. Compared accuracies of the measurement methods applied in the research. Methods are grouped by UP project phases

The half of the errors displayed in the boxes surrounding median value of the error is in the acceptable ranges below 60%. The functional methods in most cases have good accuracy, although there are potential outliers with more than 60% for NESMA estimated method. Measurement methods based on the use cases are not accurate as the functional methods. We believe that cause of this inaccuracy is lack of the specification standards for definition of use cases.

Diagram shows the best practice methods that can be applied during the project lifecycle:

1. NESMA indicative method that has the best accuracy at the beginning of the project;
2. NESMA estimated and Mark II that have the best accuracy in the early iterations of the elaboration phase. Standard FPA method that has even better accuracy; however, it can be applied only in the later iterations of the elaboration phase;
3. COSMIC method has the best correlation with the effort at the beginning of the construction phase.

Accuracy of the best practice measurement techniques that are applied on the project in our data set is shown in figure 10. Results are divided per project phases and compared with Boehm's and PMI results.

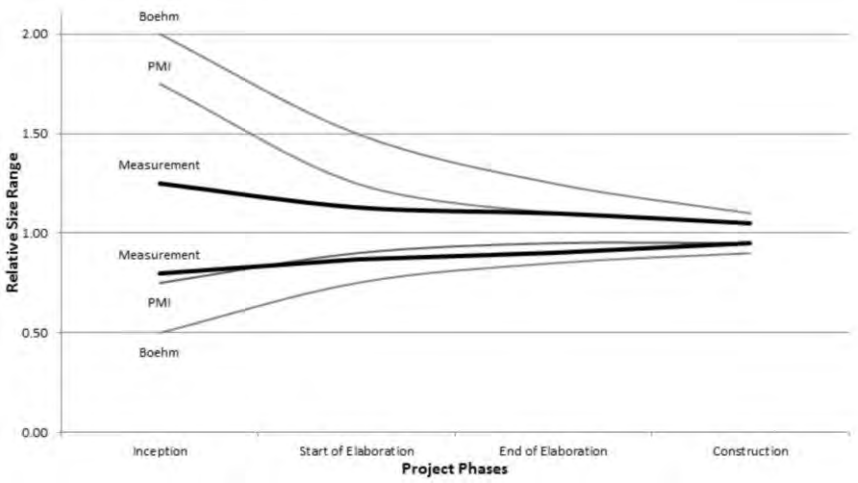


Fig. 10. Results of the estimate accuracy based on measurement compared with the Boehm's and PMI results

Results of the estimation using the measurement techniques and applying linear regression have better results than the results found in practice. Estimating errors applied on the data set are within Boehm's and PMI boundaries (except the lower boundary of the PMI where we have minor difference). In the earlier phases of the project, measurement methods have significantly better results than any results reported in the practice.

This is a concrete proof that there is no reason for assuming that the algorithmic models based on the measurement and analysis would not give good results. If applied correctly, estimating methods based on the metric can be used as quick and accurate methods for estimating the project effort.

Research shows that for small and medium projects, there is no significant difference between linear estimating models such as COBRA and nonlinear models such as COCOMO II or SLIM. Although it is reasonable to assume that many factors include nonlinear dependency between the size and the effort, it is shown that the effects of nonlinear behavior can be found only on the larger projects. A current trend in the software development is dividing large projects in smaller subprojects where a system is implemented in the iterative manner. Hence, this is a valuable conclusion. According to the results in the research, the linear models are applicable on the smaller projects; hence, project teams that practice an agile or iterative development can use simpler linear methods for determining functionalities between the size and the effort.

The fact that linear models can be applied is important because beside the simplicity, linear models enable project teams to track comprehensible performance indicators such as productivity. Measuring and tracking productivity is important in the software organizations, since one of the most important tasks is optimizing process performances.

A very important result in the paper is the fact that good accuracy of the estimating model is achieved although non-technical characteristics of the system are not used. Expert judgment methods consider both features that should be implemented and non-functional factors such as programmer/analyst capability or platform/technology complexity. Accuracy of estimates based on the bare functional size evaluated in this paper is close or even better than accuracies found in the practice, even if the non-functional parameters are not included in the research. This emphasis the great potential of the functional measurement presented in the paper. If the non-functional parameters that are used in the expert judgment methods are applied on metric presented in this paper, accuracy will be significantly improved and probably give far better results than those reported by Boehm and PMI.

5.2. Future Work

Our study shows that there is a good correlation between the pure functional size and the actual project effort; however, the influence of the nontechnical factors on the effort is not evaluated. Current estimation models, such as COCOMO II or SLIM, use several nontechnical parameters to adjust the functional size and improve accuracy of the predicted estimates. The inclusion of these parameters in the measurement model presented in this paper would significantly improve the accuracy of estimates. Note that our study shows that the accuracy of estimates in currently analyzed bare functional models is, in most of the cases, better than accuracy expected in the practice.

In the present data set, information about the nonfunctional parameters of the system is either incomplete, or it is arguable whether they are objective. The organization used as a source of information does not have implemented some crucial processes on the CMMI maturity level 3, such as organizational process definition (OPD) or decision analysis and resolution (DAR). Therefore, no organizational standards exist to assess the ability of workers who worked on the project. Lack of the objective organizational standards might cause significant variation in a team capability or product complexity assessment as it will rely on the subjective evaluation of the particular team members. As data about the nontechnical characteristics are too subjective, they have been discarded from the research. Next step in the research will be a creation of an organization-wide model for evaluation of nonfunctional parameters and including these values as parameters in the uniform tuple space. Once the nonfunctional parameters are collected, estimating models such as COCOMO II, SLIM, or COBRA can be directly applied, and we would be able to evaluate which of them corresponds to the project data.

In addition, current maturity level does not guarantee that processes are optimal; only that their cost can be consistently predicted using the measurement data. One of our future goals is to evaluate how measurement

and analysis can be applied on the higher maturity levels where measurement is used both for prediction and process optimization.

Acknowledgement. This work has been partially funded by the Ministry of Education and Science of the Republic of Serbia, grant. no. TR32047.

References

1. Stellman A., Greene J.: Applied Software Project Management, O'Reilly Media, Inc., 1 edition (November 2005), ISBN: 0-596-00948-8
2. Beck K., Extreme Programming Explained: Embrace The Change, Addison Wesley, 2004
3. Hill P.: Practical Software Project Estimation: A Toolkit for Estimating Software Development & Duration, 3rd Edition, McGraw - Hill, 2010
4. CHAOS Report: The Standish Group, 2008.
5. CMMI Product Team: CMMI® for Development, Version 1.3, Carnegie Mellon, Software Engineering Institute, November 2010
6. Boehm B.: Software Engineering Economics, Englewood Cliffs, N.J: Prentice-Hall, 1981.
7. Boehm B.: Software Cost Estimation with COCOMO II, Prentice-Hall, New Jersey, USA, 2000
8. A Guide to the project Management Body of Knowledge (PMBOK Guide), 4th Edition, The Project Management Institute PMI. ISBN 978-1-933890-51-7
9. Park. R.: Software size measurement: a framework for counting source statements, CMU/SEI-92-TR-020.
10. Garmis D., Herron D.: Functional Point Analysis, Addison-Wesley, Boston USA, 2001
11. Clemmons R.: Project Estimation With Use Case Points, The Journal of Defense Software Engineering, 2006
12. Abran A., Meli R., Simons C.: COSMIC Full Function Point Method: 2004 – State of Art, SMEF04, Rome, Italy, Jan. 2004.
13. ISO/IEC 14143-1:2007 Information technology – Software measurement – Functional size measurement
14. Halstead M.: Elements of Software Science, Elsevier North-Holland, New York, USA, 1997
15. McCabe T.: A Complexity Measure, IEEE Transactions on Software Engineering: 308–320. December 1976
16. Netherlands Software Metrics Association (NESMA): Definition and Counting Guidelines for the Application of Function Point Analysis, Version 2.0, Amsterdam NESMA 1997.
17. Symons C.: Software Sizing and Estimating, Mk II Function Point Analysis, John Wiley & Sons, Chichester, England, 1991
18. Costagliola G., Ferrucci F., Tortora G., Vitiello G.: Class Point: An Approach for the Size Estimation of Object-Oriented Systems, IEEE Transactions on Software Engineering, Vol. 31, No.1 January 2005
19. Janaki Ram D., Raju S.V.G.K.: Object Oriented Design Function Points, apaqs, pp.121, The First Asia-Pacific Conference on Quality Software (APAQS'00), 2000

20. Galea S.: The Boeing Company: 3D function point extensions, v. 2.0, release 1.0, Boeing Information and Support Services, Research and Technology Software Engineering, June 1995.
21. Reifer D.: Web Development: Estimating Quick-To-Market Software, IEEE Software, vol. 17, pp. 57-64, 2000.
22. Fetcke T., Abran A., Nguyen T.H.: Mapping the OO-Jacobson Approach into Function Point Analysis, Proc. TOOLS-23"97, IEEE Press, Piscataway, N.J., 1998.
23. Uemura T., Kusumoto S., Inoue K.: Function Point Measurement Tool for UML Design Specification, Proceedings of the 6th International Symposium on Software Metrics, p.62, November 04-06, 1999
24. Albrecht A., Gaffney J.: Software Function, Source lines of Code, and Development Effort Prediction: A Software Science Validation, IEEE Transactions on Software Engineering, SE-9 (6): 639-648, (1983).
25. Nelson R.: Management Hand Book for the Estimation of Computer Programming Costs, Systems Development Corp., 1966.
26. Briand L., El Emam K., Bomarius F.: COBRA: A Hybrid Method for Software Cost Estimation, Benchmarking, and Risk Assessment, International Software Engineering Research Network Technical Report ISERN-97-24, 1997
27. Kemerer C.: An empirical validation of software cost estimation models, Communications of the ACM, v.30 n.5, p.416-429, May 1987.
28. Putnam L.: A General Empirical Solution to the Macro Software Sizing and Estimating Problem, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. SE-4, NO. 4, JULY 1978
29. Walston C., Felix C.: A method of programming measurement and estimation, IBM Systems Journal, vol. 16, no. 1, pp. 54-73, 1977.
30. Bailey J., Basili V.: A meta model for software development resource expenditure, in Proceedings of the International Conference on Software Engineering, pp. 107–115, 1981.
31. Kruchten R.: The Rational Unified Process – An Introduction, Addison-Wesley, Boston, USA, 2000
32. Microsoft Solutions Framework version 3.0 Overview, White paper, published 2003, Microsoft Press
33. Schwaber K., Beedly M.: Agile Software Development with Scrum, Prentice Hall, 2001.
34. Mendes E., Watson I., Triggs C., Mosley N., Counsell S.: A comparison of development effort estimation techniques for Web hypermedia applications, In: Proceedings of the Eighth IEEE Symposium on Software Metrics, 2002
35. Mendes E.: Cost Estimation Techniques for Web Projects, IGI Publishing Hershey, PA, USA 2007, ISBN:1599041359 9781599041353
36. Ferrucci F., Gravino C., Di Martino S.: Estimating Web Application Development Effort Using Web-COBRA and COSMIC: An Empirical Study, Proceedings of the 35th Euromicro Conference on Software Engineering and Advanced Applications, 2009
37. Attarzadeh I., and Ow S. H.: Improving the Accuracy of Software Cost Estimation Model Based on a New Fuzzy Logic Model. World Applied Sciences Journal, 8 (2): 177-184, January 2010. ISSN 1818-4952.
38. Mendes E., Mosley N.: Bayesian Network Models for Web Effort Prediction: A Comparative Study, IEEE Transactions on Software Engineering, v.34 n.6, p.723-737, November 2008

39. Ajitha, S. Kumar, T.V.S. Geetha, D.E. Kanth, K.R.: Neural Network model for software size estimation using Use Case Point approach,
40. Buglione L., Cuadrado - Gallego J., Gutiérrez de Mesa A.: Project Sizing and Estimating: A Case Study Using PSU, IFPUG and COSMIC, In: IWSM/Metrikon/Mensura '08 Proceedings of the International Conferences on Software Process and Product Measurement, 2008
41. Finnie G., Wittig G., Desharnais J.: A Comparison of Software Effort Estimation Techniques: Using Function Points with Neural Networks, Case - Based Reasoning and Regression Models, Journal of Systems and Software, vol. 39, no. 3, pp. 281 - 289, 1997
42. Farahneh H., Issa A.: A Linear Use Case Based Software Cost Estimation Model, World Academy of Science Engineering and Technology, no. 32, 2011, pp.504 - 508
43. Conte S.D., Dunsmore H.E., Shen V. Y.: Software engineering metrics and models. The Benjamin/Cummings Publishing Company, Inc., 1986.
44. Foss T., Stensrud E., Kitchenham B., Myrtveit I.: A Simulation Study of the Model Evaluation Criterion MMRE, IEEE Transactions on Software Engineering, 29(11), 895 - 995, 2003
45. Anda, B.: Comparing effort estimates based on use cases with expert estimates. In Proceedings of Empirical Assessment in Software Engineering (EASE 2002) (Keele, UK, April 8-10, 2002)
46. Anda, B., Dreiem, D., Sjøberg, D.I.K., and Jørgensen, M.: Estimating Software Development Effort Based on Use Cases - Experiences from Industry. In M. Gogolla, C. Kobryn (Eds.): UML 2001 - The Unified Modeling Language. Modeling Languages, Concepts, and Tools, 4th International Conference, Toronto, Canada, October 1-5, 2001, LNCS 2185, Springer-Verlag, pp. 487-502.
47. Anda B., et al.: Effort Estimation of Use Cases for Incremental Large Scale Software Development, 27th International Conference on Software Engineering, St Louis, MO, 15-21 May 2005: 303-311.
48. Lavazza L., Robiolo G.: The role of the measure of functional complexity in effort estimation, In: PROMISE '10 Proceedings of the 6th International Conference on Predictive Models in Software Engineering, 2010
49. Abran A., Moore J.: Guide to the Software Engineering Body of Knowledge, IEEE Computer Society, 2004. ISBN: 0769523307

Jovan Popović received a MsC degree in electrical engineering and computer science from the University of Belgrade in 2010. He is a software practicioner specialized in project management and web application development. The focus of his research is application of software metrics and business inteligence in the software development process.

Dragan Bojić received a PhD degree in electrical engineering and computer science from the University of Belgrade in 2001. He is an assistant professor at the Faculty of Electrical Engineering, University of Belgrade. His research interests focus on software engineering techniques and tools, and e-learning.

Received: March 16, 2011; Accepted: December 16, 2011.

CIP – Каталогизација у публикацији
Народна библиотека Србије, Београд

004

COMPUTER Science and Information Systems : the International journal / Editor-in-Chief Mirjana Ivanović. – Vol. 9, No 1 (2012) - . – Novi Sad (Trg D. Obradovića 3): ComSIS Consortium, 2011 - (Belgrade : Sigra star). –30 cm

Polugodišnje. – Tekst na engleskom jeziku

ISSN 1820-0214 = Computer Science and Information Systems
COBISS.SR-ID 112261644

Cover design: V. Štavljanin
Library administration: M. Petković
Printed by: Sigra star, Beograd

ComSIS Computer Science and Information Systems
Volume 9, Number 1, January 2012

Contents

Editorial

Papers

1 A Personalized Multimedia Contents Recommendation Using a Psychological Model
Won-Ik Park, Sanggil Kang, Young-Kuk Kim

23 Representation of Texts in Structured Form
Mladen Stanojević, Sanja Vraneš

49 A Genetic Algorithm for the Routing and Carrier Selection Problem
Jozef Kratica, Tijana Kostić, Dušan Tošić, Djordje Dugošija, Vladimir Filipović

63 An XML-algebra for Efficient Set-at-a-time Execution
Maxim Lukichev, Boris Novikov, Pankaj Mehra

81 A Decision Support Method for Evaluating Database Designs
Erki Eessaar, Marek Soobik

107 An Enterprise Information System Agility Assessment Model
Rabah Imache, Said Izza, Mohamed Ahmed-Nacer

135 Selecting a Methodology for Business Information Systems Development: Decision Model and Tool Support
Damjan Vavpotič, Olegas Vasilecas

165 Improving the Evaluation of Software Development Methodology Adoption and its Impact on Enterprise Performance
Damjan Vavpotič, Tomaž Hovelja

189 A Framework for Developing and Implementing the Enterprise Technical Architecture
Tiko Iyamu

207 Avoiding Unstructured Workflows in Prerequisites Modeling
Boris Milašinović, Krešimir Fertalj

235 Improving Program Comprehension by Automatic Metamodel Abstraction
Michal Vagač, Ján Kollár

249 An Approach to Automated Conceptual Database Design Based on the UML Activity Diagram
Drazen Brđanin, Slavko Marić

285 Usage of Technology Enhanced Learning Tools and Organizational Change Perception
Mladen Čudanov, Gheorghe Savoiu, Ondrej Jaško

303 Effective Hierarchical Vector-based News Representation for Personalized Recommendation
Mária Bieliková, Michal Kompan, Dušan Zeleník

323 Concept Extraction and Clustering for Search Result Organization and Virtual Community Construction
Shihn-Yuarn Chen, Chia-Ning Chang, Yi-Hsiang Nien, Hao-Ren Ke

357 A Method for Decision Making with the OWA Operator
José M. Merigó, Anna M. Gil-Lafuente

381 A Generic Parser for Strings and Trees
Riad S Jabri

411 Using Lightweight Formal Methods to Model Class and Object Diagrams
Fernando Valles-Barajas

431 S-TRAP: Optimization and Evaluation of Timely Recovery to Any Point-in-time (TRAP)
Chao Wang, Zhanhuai Li, Na Hu, Yanming Nie

455 A Comparative Evaluation of Effort Estimation Methods in the Software Life Cycle
Jovan Popović, Dragan Bojić

ISSN: 1820-0214