Proceedings of the

17th European Conference on Software Maintenance and Reengineering

CSMR 2013

Proceedings of the

17th European Conference on Software Maintenance and Reengineering

5-8 March 2013 / Genova, Italy

Sponsored by

Reengineering Forum industry association

and

Università degli Studi di Genova - DIBRIS

Technical Co-sponsor

IEEE Computer Society
Technical Council on Software Engineering (TCSE)

Los Alamitos, California

Washington • Tokyo

IEEE Computer Society Order Number P4948
BMS Part Number: CFP13102-ART
ISBN 978-0-7695-4948-4

Additional copies may be ordered from:

IEEE Computer Society
Customer Service Center
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314
Tel: + 1 800 272 6657
Fax: + 1 714 821 4641
http://computer.org/cspress
csbooks@computer.org

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Tel: + 1 732 981 0060
Fax: + 1 732 981 9667
http://shop.ieee.org/store/
customer-service@ieee.org

IEEE Computer Society
Asia/Pacific Office
Watanabe Bldg., 1-4-2
Minami-Aoyama
Minato-ku, Tokyo 107-0062
JAPAN
Tel: + 81 3 3408 3118
Fax: + 81 3 3408 3553
tokyo.ofc@computer.org

Individual paper REPRINTS may be ordered at: <reprints@computer.org>

Editorial production by Bob Werner
Cover art production by Mark Bartosik

IEEE Computer Society
Conference Publishing Services **(CPS)**
http://www.computer.org/cps

2013 17th European Conference on Software Maintenance and Reengineering

CSMR 2013

Table of Contents

Traceability & Impact Analysis

Reengineering

Languages & Programming

Architecture Analysis & Evolution

Early Research Achievements Track

Repository Mining

Software Maintenance

Software Analysis

European Projects Track

Welcome from the Conference Chairs

The Conference on Software Maintenance and Reengineering (CSMR) is the premier European conference on the theory and practice of maintenance, reengineering and evolution of software systems. CSMR promotes discussion and interaction among researchers and practitioners about the development of maintainable systems, and the evolution, migration and reengineering of existing ones.

The 17th edition of this conference takes place in Genova, Italy, known as the city of lighthouse. The conference is hosted by the Software Engineering Research Group at DIBRIS, University of Genova.

As in previous years, the main conference hosts a wide range of scientific events: keynote talks by two distinguished speakers : Michele Lanza and Ezio Armando; technical research presentations by scientists from all over the world; ERA track presentations discussing novel research ideas in early stages of development; industrial track presentations from practitioners in the field; tool demonstrations showing how scientific approaches are supported by working tools; a doctoral symposium for PhD students; and presentations of ongoing European projects in the field. In addition to this main conference programme, CSMR 2013 will also host some satellite events.

The technical research track of CSMR 2013 attracted 101 abstract submissions, that made it into 80 full paper submissions. Each of these submissions was reviewed by 3 members of an international program committee, chaired by Anthony Cleve and Filippo Ricca. Out of all submitted papers, only 29 could be accepted.

We extend our gratitude to the numerous organisations that helped in making CSMR 2013 a success. We thank our principal sponsors, the Reengineering Forum (REF) and the Dipartimento di Informatica, Bioingegneria, Robotica ed Ingegneria dei Sistemi (DIBRIS) of the University of Genova. We also thank our technical sponsors IEEE Computer Society and the Technical Council on Software Engineering (TCSE).

We are also grateful to the many individuals that contributed to the CSMR 2013 conference programme: the participants of the conference, the keynote presenters, the authors who submitted a contribution to one or more of the conference events, the programme committees of the different events for their excellent reviewing work and their help in promoting the conference, and the CSMR steering committee.

Special thanks are due to all the members of the organization committee, for the excellent work they achieved: Alberto (publicity chair), Alessandro & Tanja (European track chairs), Andy & Ali (ERA chairs), Elliot (advisor), Gianna (local chair), Jens (tutorial chair), Joost (industrial chair), Massimiliano & Radu (doctoral symposium chairs), Maurizio (web chair), Walter & Michel (workshop chairs), Xavier & Jacques (tool demonstration chairs).

We hope you will find the conference program enriching and stimulating, and we hope you will enjoy reading these conference proceedings!

Maura Cerioli
University of Genova
General chair

Anthony Cleve
University of Namur
Program chair

Filippo Ricca
University of Genova
Program chair

Welcome from the Doctoral Symposium Chairs

It is our pleasure to welcome you to the CSMR 2013 Doctoral Symposium. The Doctoral Symposium provides an opportunity to PhD students to receive early feedback about their research accomplishments and plans. Specifically, during the conference, students participating in the Symposium will present their work to a panel of experts, which will provide them with advices on how to improve their research and how to better plan their future research activities, in order to increase the likelihood of accomplishing a successful PhD.

This year the CSMR 2013 Doctoral Symposium received seven submissions. After a rigorous reviewing process, we accepted three papers. The first one, by John Businge from the Eindhoven University of Technology, studies the co-evolution of the Eclipse SDK framework and third-party plugins, analyzing various aspects such as the survival of these plugins and their compatibility with the Eclipse SDK framework. The second one, by Veronika Bauer from the Technical University of Munich, investigates, by means of an industrial survey, the issues and needs of industrial developers regarding software reuse. The third one, by Csaba Nagy from University of Szeged, is about a tool and an approach for the analysis of data intensive applications, and about his experience of applying such a tool in the software industry.

In summary, the three PhD proposals that will be presented at CMSR 2013 deal with three complementary aspects of software maintenance and reengineering. We are sure that the audience will enjoy the presentations and students will receive a useful feedback.

Massimiliano Di Penta
University of Sannio
Italy

Radu Marinescu
University of Timisoara
Romania

Welcome from the Early Research Achievements Chairs

The Early Research Achievements (ERA) Track aims at providing researchers with a forum for discussing novel research ideas in early stages of development. The topics of interest for this track are the same as the main research track, but we aim at creating a stimulating atmosphere where researchers can present and get early feedback on promising work that has not yet been fully evaluated.

This year's Early Research Achievements track attracted 40 submissions written by authors from 22 countries on 4 continents. After an intense reviewing process in which each paper was reviewed by at least 3 members of the ERA program committee, 15 papers were accepted by the 20 members of the ERA program committee. As expected, the majority of the papers covers traditional CSMR topics such as reengineering and software analysis. However, the ERA track is proud to also provide a platform for CSMR-related topics that have not been discussed before. The following paragraphs summarize the papers of the ERA track and thereby provide a guide for interested readers.

1) Repository mining
Tóth et al. use the version control history to follow the changes of source code elements. Thung et al. study the network structure of social coding in GitHub. Xia et al. present a comparative study of supervised learning algorithms for re-opened bug prediction. Stevens et al. have developed a history querying tool and applied it to detect multi-version refactorings. Mihancea and Marinescu study whether there is a correlation between changes, defects and polymorphism.

2) Software maintenance
Csaba et al. relate clustering measures to software quality. Polychniatis et al. present a technique to detect cross-language dependencies generically. Kochar et al. perform a preliminary study of 50,000 open source projects to see how they test their software. Molitorisz presents a pattern-based approach for refactorization of sequential source code. Sasaki et al. investigate the reordering of program statements to improve source code readability.

3) Software analysis
Gravino et al. present an early investigation into the contribution of class and sequence diagrams in source code on program comprehension tasks. Kazato et al. investigate Incremental feature location and identification in source code. Jezek et al. explain an approach to supply a compiler with static compatibility checks through the analysis of third-party libraries. Ghaith presents a profile-based, load-independent bug detection and analysis mechanism in regression testing of software systems. Scaniello et al. explore how to use the GPU to green an intensive and massive computation system.

The track chairs would like to thank the members of the CSMR ERA 2012 program committee and the additional reviewers for their effort!

Andy Zaidman
Delft University of Technology
The Netherlands

Ali Mesbah
University of British Columbia
Canada

Welcome from the European Projects Track Chairs

Welcome to the European Projects Track (EU Projects track) of the 17th European Conference on Software Maintenance and Reengineering (CSMR 2013), to be held in Genova, Italy, March 2013.

Today's European projects, try to bridge the famous gap between research (e.g., academia) and practitioners (e.g., industries). These projects, in fact, can allow the research community to practice with their research ideas in real industrial environment while, at the same time, can raise the need for new and different research inspired by the needs of industry.

The EU Projects track of CSMR 2013 provides an opportunity for researchers involved in ongoing and recently completed European research projects (both national and international) related to the themes of the conference to present their projects and disseminate the objectives, deliverables, or outcome of these projects. Hence, on one side, the EU Projects track offers the possibility to project participants to share and disseminate their project results. On the other side, it allows the CSMR participants to get a better insight in which research projects are currently going on in Europe within the field of software maintenance and reengineering. The EU Projects track of CSMR represents a consolidated forum where projects, researchers and industrial practitioners can share experience, ideas and knowledge.

In this edition of the EU Projects track, we overall received 8 submissions. Each submission has been reviewed by three up to four members of the program committee of the track, composed of 14 international scientists, working in academia, research centers and industries. The reviewers assessed the completeness of the information reported in the submission about the presented project, as well as the quality of and the relevance of the submission for the CSMR audience. Of all submissions, 7 have been accepted to be published in the CSMR conference proceedings. The received submissions involved 42 authors working in 20 different institutions (10 universities, 3 research centers, and 7 industries) of 10 different countries. These numbers and the high quality of the received submissions confirm that European projects can play a key role to fill the gap between research (academia) and practitioners (industries). The final program of the EU Projects track 2013 is composed of a set of interesting works reporting promising projects investigating different fields and funded by different institutions, for instance, the European Union, national organizations, and even by individual industries.

We would express our gratitude to all the authors, the members of the program committee of the EU Projects track 2013 involved in the paper evaluation and selection activity. We would finally thank also the organizing committee and the staff of the CSMR 2013 conference and, in particular, to Maura Cerioli, Filippo Ricca and Maurizio Leotta (University of Genova, Italy), and Anthony Cleve (University of Namur, Belgium) for their important support.

We hope you enjoy the track and the reading of the track proceedings.

Alessandro Marchetto
Fondazione Bruno Kessler
Trento, Italy

Tanja E.J. Vos
Universidad Politecnica de Valencia
Spain

Welcome from the Tool Demonstrations Chairs

Welcome to the Tool Demonstrations Track of the 17th European Conference on Software Maintenance and Reengineering!

This year's Tool Demonstrations Track attracted 7 submissions from a tremendously diverse group of authors. Each paper has been reviewed by three program committee members. After review and follow-up discussions, the program committee selected 5 demonstrations for presentation in the track.

We thank all of the authors who submitted their work to the Tool Track. The quality of submissions this year was extremely high, resulting in very difficult decision making, but we feel gratified by the quality of the resulting program.

We are very grateful to the members of the program committee. Their timely completion of the reviews and their active participation in discussions are key to a fair, highly effective, selection process.

We also want to thank the CSMR steering committee and the CSMR 2013 Organizing Committee for their help in putting together such an exciting program.

Finally, we thank all of you who come to the conference. We hope you find the conference both stimulating and enjoyable!

Xavier Blanc
University of Bordeaux
France

Jacques Klein
University of Luxembourg
Luxembourg

MIMOS. System Model-Driven Migration Project

Ricardo Pérez-Castillo, Ignacio García-Rodríguez de Guzmán and Mario Piattini
Instituto de Tecnologías y Sistemas de Información (ITSI)
University of Castilla-La Mancha,
Paseo de la Universidad 4, 13071, Ciudad Real, Spain
[ricardo.pdelcastillo | ignacio.grodriguez | mario.piattini]@uclm.es

***Abstract*—The volatile IT industry often tempts companies to replace legacy information systems with new ones. However, legacy systems cannot always be completely discarded because they gradually store a significant amount of valuable business knowledge as a result of progressive maintenance over time. Most migration techniques are proposed and applied in an ad hoc way. As a result, most migration techniques have a lack of automation and formalization, which makes it difficult to reuse such techniques to large, complex legacy information systems. This paper introduces MIMOS, a third-year project aimed at developing a methodological and technological modernization framework to facilitate the migration of legacy systems based on high-level design models. The work in progress during the first year mainly focused on the definition of a business process mining technique to retrieve the business knowledge embedded in source code so that it can be reused in the target system.**

***Keywords*—Migration, architecture-driven modernization, business process, legacy systems.**

I. Motivation

Although software is an intangible object, the quality of software diminishes over time in a similar way to that of material objects. Lehman's first law states that an information system must continually evolve or it will become progressively less suitable in a real-world environment [7]. Companies currently have an enormous amount of large legacy systems which undergo the phenomenon of software erosion and software ageing. This means that existing information systems become progressively less maintainable [12]. The negative effects of software erosion can be dead code, clone programs, missing capacities, inconsistent data and control data (coupling), among others [15].

On the one hand, software maintenance is part of the software erosion problem, since software erosion is due to maintenance itself and to the uncontrolled evolution of the system over time. On the other hand, software maintenance is also part of the solution to software erosion. The successive changes in information systems transform them into Legacy Information Systems (LIS), and a new and improved system must therefore replace the previous one when the maintainability levels diminish below acceptable limits [9]. Nevertheless, the wide replacement of these systems from scratch is a key challenge since it makes a great impact on the technological, human and economic aspects of companies [14]. Firstly, the entire replacement of LISs affects technological and human aspects, since it usually involves retraining all the users in order for them to understand the new system and/or the new technology. Secondly, the new system may have a lack of specific functionalities that are missing as a result of technological changes. Thirdly, the economic aspect of companies is also affected, since the replacement of an entire LIS, by implementing a new system from scratch, implies a low Return of Investment (ROI) with regard to the old system. In addition, the development or purchase of the new system might exceed a company's budget.

Software migration is a particular type of maintenance that focuses on adaptive and perfective modifications. Indeed, according to [5], 78% of maintenance changes are corrective or behavior-preserving. Over the last two decades, reengineering has been the principal technique used to address the migration of legacy systems [2]. Reengineering is the examination and alteration of a subject system to reconstitute it in a new form and the subsequent implementation of the new form. This form may include modifications with respect to new requirements not met by the original system [4]. The main advantages is that reengineering preserves the systems' legacy knowledge and makes it possible to change software easily, reliably and quickly, resulting in a maintenance cost that is also tolerable [1].

Several reengineering proposals consist of the analysis and inspection of different software artifacts. For example, *Zou et al.* [16], statically analyze the source code and apply a set of heuristic rules to discover embedded business processes. Other methods using dynamic analysis are also proposed to preserve the embedded business knowledge, e.g., Marchetto et al. [8] discover business processes through the execution of graphical user interfaces in Web applications.

The main challenge of migration, and in general of reengineering, is that most efforts are *ad hoc* proposals, which are developed for particular platforms, technologies and specific contexts. This lack of formalization and standardization leads to another challenge related to the automation of such techniques, and the repeatability of migration techniques in large-scale projects is therefore in doubt [3]. In fact, a 2005 study [14] states that 50% of reengineering projects fail owing to the lack of standardization and automation, which often leads to overruns in costs.

Standardization and automation challenges limit the applicability of migration techniques to large and complex

1534-5351/13 $26.00 © 2013 IEEE
DOI 10.1109/CSMR.2013.68

legacy information systems. These challenges can be addressed by Model-Driven Development (MDD) principles, i.e., (i) considering and treating all software artifacts as models which conform to specific metamodels, and (ii) establishing automatic transformations between models at different abstraction levels. The Architecture Driven-Modernization (ADM) initiative (also known as software modernization) launched by the OMG, particularly advocates carrying out a reengineering process by following model-driven development principles.

ADM solves the formalization problem since it represents all the artifacts involved in the reengineering process as models, which are represented in accordance with specific metamodels. ADM therefore treats all software artifacts homogenously, i.e., as models that can be transformed into other models by using deterministic transformations. The model transformations can consequently be automated through their formalization. Furthermore, the model-driven development principles make it possible to reuse models used in different modernization projects, since a computational independent model (CIM) can be transformed into several platform independent models (PIM), and each PIM model can in turn be transformed into several platform specific models (PSM).

The remaining of this paper is structured as follows: Section II introduces the goals of the project. Section III explains in detail work packages and the schedule. Section IV provides the work in progress. Finally, Section V presents preliminary results and open issues.

II. PROJECT GOALS

The main research goal of MIMOS is *to develop a methodological and technological ADM-based framework to facilitate the migration of legacy systems based on high-level design models*. MIMOS carries out applied research, and divides the main goal into 9 sub-goals:

O1. To develop a platform-independent process supporting the reverse engineering of legacy information systems.
O2. To define a strategy for representing the retrieved knowledge in a platform-independent way.
O3. To build a mechanism for obtaining business rule / process views from legacy systems.
O4. To develop a set of refactoring techniques to be applied to high-level design models.
O5. To obtain a measuring system for assessing the gain of the migration process.
O6. To propose a set of techniques to automatically generate analysis and design models for the modernized system.
O7. To develop an incremental and iterative methodology for model-driven migration.
O8. To implement a technological framework by automating all the proposed methods and techniques.
O9. To conduct a real-life case study with a large, industrial legacy information system.

III. TECHNICAL DESCRIPTION

Table I provides the original name and code, duration, number of participants as well as the funding source and amount of the MIMOS project.

TABLE I. PROJECT DATASHET

Full name	Proyecto MIMOS. MIgración dirigida por MOdelos de Sistemas de información
Project Code	IDI-20120260
Duration	3 years
Date	From 01/01/2012 to 31/12/2014
Status	Milestone 1
Participants	12 full-time researchers
Source of funding	CDTI (Spanish Industrial Technology Development Center) with FEDER funds
Amount of funding	850,000 €

TABLE II. PROJECT MILESTONES AND WORK PACKAGES

Milestone	Work Packages
M1	WP0. Project coordination, technical and financial management WP1. Development of a reverse engineering technique for retrieving and representing embedded knowledge
M2	WP2. Obtaining business process views from legacy source code WP3. Definition of model-driven refactoring
M3	WP4. Development of a technological framework and tools. WP5. Industrial validation

TABLE III. PROJECT OUTCOMES AND DELIVERABLES

WPx	Outcomes	Month
WP0	O.0.1. Project website development	3
	O.0.2. Monitoring reports	6, 12, 18, 24, 30
	O.0.3. Project Dissemination	Every time
	O.0.4. Final Report	36
WP1	O.1.1. Generic reverse engineering process	6
	O.1.2. Generic Migration Process	12
	O.1.3. Catalog of legacy software assets	6
	O.1.4. KDM Extension (ISO 19506)	10
WP2	O.2.1. Model Transformations	24
WP3	O.3.1. Set of refactoring rules	14
	O.3.2. Metamodels to represent refactoring rules	18
	O.3.3. Implementation of refactoring rules	24
	O.3.4. Measurement mechanism for the refactoring rules	24
WP4	O.4.1. Model repositories for migration models	24
	O.4.2. QVT/ATL model transformations	30
	O.4.3. Supporting tools	34
	O.4.4. Integrated migration environment	36
WP5	O.5.1. Pilot, small-scope case studies	12, 24
	O.5.2. Industrial case studies with real-life Systems	36
	O.5.3. Proposal refinement	36

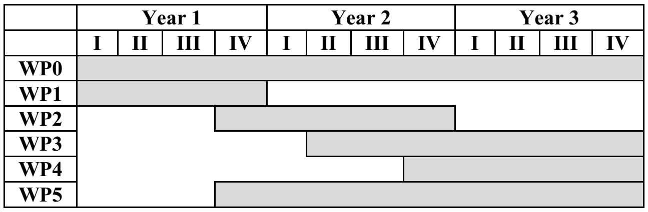

Figure 1. MIMOS project schedule

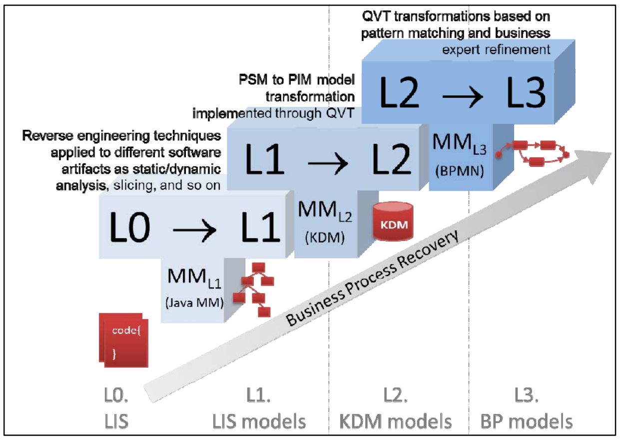

Figure 2. Extraction of business process views

Table II presents the six work packages of the project, which are divided into three milestones (one per year). Table III shows the description of the expected achievements and deliverables of the project. Figure 1 shows the temporal schedule of the MIMOS project according to the work packages.

IV. Work in Progress

The main effort during the first year of the MIMOS project has been focused on work packages WP0, WP1 and WP2. According to WP2 (see Table II), the work in progress is particularly addressing the discovery of business process views from source code by retrieving embedded knowledge, which have to be migrated.

For this purpose, a framework for recovering business processes from legacy source code has been developed. This framework is extensible for different programming languages since it is based on the ADM. In addition, this framework supports the KDM (Knowledge Discovery Metamodel) [6] standard proposed by the ADM initiative. KDM enables the representation and management of knowledge extracted by means of reverse engineering from all the different software artifacts of legacy systems in an integrated and platform-independent way. Thus, that legacy knowledge is gradually transformed into business processes. Hence, the framework is divided into 4 abstraction levels with 3 model transformations among them (see Figure 2):

Level L0. This level represents the legacy information system in the real world, and its source code to recover underlying business processes.

Level L1. This level represents several specific models, i.e., one model for each different software artifact involved in the archeology process like source code, database, user interface, and so on. Traditional reverse engineering techniques [3] such as static analysis, dynamic analysis, program slicing, formal concept analysis, and so on, could be used to extract the knowledge from any software artifact and build PSM models related to it. These PSM models are represented according to specific metamodels. For example, a Java metamodel may be used to model the legacy source code, or an SQL metamodel to represent the database schema, etc.

Level L2. This level consists of a single PIM that represents the integrated view of the set of PSM models at L1. The KDM metamodel is used so that L2 works as a KDM repository that can be progressively populated with knowledge extracted from the different legacy artifacts and information systems of an organization. In addition, L2 is represented in a technological-independent way due to the fact that KDM standard abstract all those details concerning the technological viewpoint (e.g. the program language). The transformation between levels L1 and L2 consists of a set of model transformations implemented using QVT (Query/View/Transformation).

Level L3. Finally, this level depicts, at the end of the archeology process, the business process models retrieved from a legacy system. Business process models at L3 represent a CIM and are represented according to the BPMN (Business Process Modeling and Notation). This level closes the conceptual gap between the software architecture views and underlying business rules. The last transformation is based on a set of patterns. When a specific structure is detected in the KDM model at L2, each pattern indicates what elements should be built and how they are interrelated in the business process model at L3 [10]. This pattern matching is implemented through a QVT transformation [11].

The obtained models are a first sketch of the business process, which can be refined by business experts. This is due, for instance, to the fact that not all parts of current business processes are executed by legacy information systems, i.e., there are some manual business activities. Although experts post-intervention can be necessary, the first version of business processes, compared with business process redesign by business experts from scratch, represents a more efficient and less error-prone solution to get business process models. In addition, the business process redesign by experts from scratch might discard meaningful business knowledge that is only embedded in legacy information systems. This knowledge is then used for migrate main business functionalities to the new system.

V. Preliminary Results & Open Issues

The framework implemented in WP2 has been applied to several pilot industrial case studies to retrieve business processes from a wide variety of legacy information systems. The conduction of these industrial case studies has allowed improving a preliminary tool and refining the migration framework. So far, the framework has been applied to six legacy information systems: (i) a system managing a Spanish author organization; (ii) an open source CRM (Customer Relationship Management) system; (iii) an enterprise information system of the water and waste industry; (iv) an e-government system used in a Spanish local e-administration; (v) a high school LMS (Learning Management System); and finally (vi) an oncological evaluation system used in Austrian hospitals.

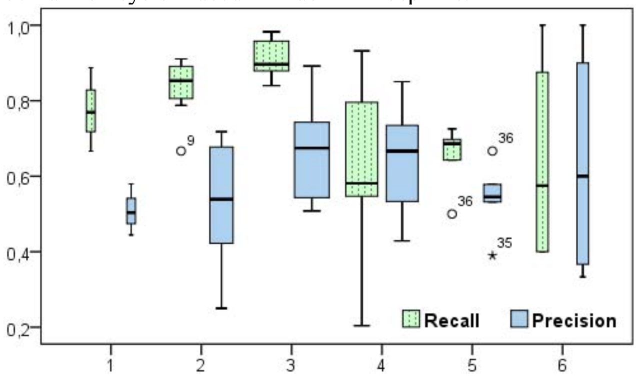

Figure 3. Effectiveness summary of case studies

These studies evaluated the effectiveness and efficiency of the technique applied through the tool. On one hand, effectiveness is measured through precision and recall. Precision measures the exactness or fidelity of the business processes recovered, whereas recall measures their completeness. These measures are computed regarding retrieved functionalities. On the other hand, efficiency of the technique developed in WP2 is evaluated through the time spent on the recovery as well as the scalability to larger legacy information systems. Figure 3 summarizes results obtained from case studies regarding effectiveness. Precision and recall values vary from a system to another, although the value trend is a recall higher than precision. This means that the technique retrieves a great number of business activities although a few of them could be erroneously recovered.

Further empirical validation will be done in next work package WP5 with *Universitas XXI*, a large industrial system in charge of electronic administration of several Spanish universities. *Universitas XXI* is a client-server application written in *Visual Basic* and a set of *Oracle* forms with a total size of 2241 KLOC.

A. *Open issues*

One of the most frequent clarification questions is related to the possibility (or not) of migrating cross-cutting business functionalities from various heterogeneous applications or subsystems integrating a whole enterprise information system, which are present in several companies. It is an important open issue to be addressed during the project according to the problems of *delocalization* and *interleaving* of the embedded business knowledge [13]. These problems lie in the fact that pieces of knowledge are usually scattered between many applications and, in turn, a single application contains several pieces of business knowledge.

The second major limitation revealed is related to the time spent on manual post-intervention to refine the first sketch of the migrated system that was automatically obtained. This time could be the bottleneck in some real migrating projects. Although the pilot case studies show that our proposal is less error-prone and time-consuming than manual modeling from scratch, the manual time should be reduced during MIMOS project.

REFERENCES

[1] Bennett, K.H. and V.T. Rajlich, Software maintenance and evolution: a roadmap, in Proceedings of the Conference on The Future of Software Engineering. 2000, ACM: Limerick, Ireland.

[2] Bianchi, A., D. Caivano, V. Marengo, and G. Visaggio, Iterative Reengineering of Legacy Systems. IEEE Trans. Softw. Eng., 2003. 29(3): p. 225-241.

[3] Canfora, G., M. Di Penta, and L. Cerulo, Achievements and challenges in software reverse engineering. Commun. ACM, 2011. 54(4): p. 142-151.

[4] Chikofsky, E.J. and J.H. Cross, Reverse Engineering and Design Recovery: A Taxonomy. IEEE Softw., 1990. 7(1): p. 13-17.

[5] Ghazarian, A., A Case Study of Source Code Evolution, in 13th European Conference on Software Maintenance and Reengineering (CSMR'09), R. Ferenc, J. Knodel, and A. Winter, Editors. 2009, IEEE Computer Society: Kaiserslautern, Germany. p. 159-168.

[6] ISO/IEC, ISO/IEC 19506. Knowledge Discovery Meta-model (KDM), v1.1 (Architecture-Driven Modernization). http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?ics1=35&ics2=080&ics3=&csnumber=32625. 2012, ISO/IEC. p. 302.

[7] Lehman, M.M., D.E. Perry, and J.F. Ramil, Implications of Evolution Metrics on Software Maintenance, in Proceedings of the International Conference on Software Maintenance. 1998, IEEE Computer Society. p. 208-217.

[8] Marchetto, A. and C. Di Francescomarino, Parameterised trace selection technique for process model recovering. Software, IET, 2011. 5(6): p. 563-575.

[9] Mens, T., Introduction and Roadmap: History and Challenges of Software Evolution Software Evolution (Springer Berlin Heidelberg), 2008. 1: p. 1-11.

[10] Pérez-Castillo, R., I. García-Rodríguez de Guzmán, O. Ávila-García, and M. Piattini, On the Use of Patterns to Recover Business Processes, in 25th Annual ACM Symposium on Applied Computing (SAC'10). 2010, ACM: Sierre, Switzerland. p. 165-166.

[11] Pérez-Castillo, R., I. García-Rodríguez de Guzmán, and M. Piattini, Implementing Business Process Recovery Patterns through QVT Transformations, in International Conference on Model Transformation (ICMT'10). 2010, Springer-Verlag. p. 168-183.

[12] Polo, M., M. Piattini, and F. Ruiz, Advances in software maintenance management: technologies and solutions. 2003: Idea Group Publishing. 286.

[13] Ratiu, D., R. Marinescu, and J. Jurjens, The Logical Modularity of Programs, in Working Conference on Reverse Engineering (WCRE'09). 2009, IEEE C. S.: Lille, France. p. 123-127.

[14] Sneed, H.M., Estimating the Costs of a Reengineering Project. Proceedings of the 12th Working Conference on Reverse Engineering. 2005: IEEE Computer Society. 111 - 119.

[15] Visaggio, G., Ageing of a data-intensive legacy system: symptoms and remedies. Journal of Software Maintenance, 2001. 13(5): p. 281-308.

[16] Zou, Y. and M. Hung, An Approach for Extracting Workflows from E-Commerce Applications, in Proceedings of the Fourteenth International Conference on Program Comprehension. 2006, IEEE Computer Society. p. 127-136.