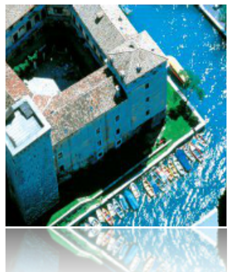

Riva Del Garda
Trento, Italy

Proceedings of the 28th IEEE International Conference on

Software Maintenance

ICSM 2012

September, 2012

IEEE Catalog Number: CFP12079-ART
ISBN: 978-1-4673-2312-3

Editors

Paolo Tonella, FBK, Trento, Italy
Massimiliano Di Penta, University of Sannio, Italy
Jonathan I. Maletic, Kent State University, USA

Sponsors

IEEE Computer Society
Technical Council on Software Engineering

Supporting Organizations

Fondazione Bruno Kessler (FBK)
SAP
GrammaTech

Foreword

Welcome to the 28th IEEE *International Conference on Software Maintenance* in Riva del Garda, Trento, Italy. Riva del Garda (or simply *Riva*) is a small city located at the north-western corner of Lake Garda, in the middle of the Alps. Thanks to the presence of the lake, the weather in Riva favors the Mediterranean vegetation: lemon trees, olive trees, laurel and palm trees. Riva is actually an oasis of Mediterranean at the foot of the Dolomites. We hope you will enjoy the location as much as the conference.

ICSM 2012 is the result of a huge effort undertaken by 21 people who served in the Organizing Committee. The Program Committee for the Technical Papers consists of 67 people. The total number of Program Committee members for all tracks is 129 people. Moreover, several additional reviewers contributed to the review process. The names of these reviewers are listed in the following pages and we want to thank all of them for their great work and contributions. The high quality of the program descends from the quality of the submissions and of the reviewers.

We also want to thank the technical sponsors of the conference, the IEEE Computer Society and the IEEE Technical Council on Software Engineering, for their help and support. We extend our gratitude to our supporters for their generous contributions: FBK, SAP and GrammaTech.

Four additional events are co-located with ICSM this year: the 4th *International Symposium on Search Based Software Engineering* (SSBSE), for the first time in co-location with ICSM, the 12th IEEE *International Working Conference on Source Code Analysis and Manipulation* (SCAM), the 14th IEEE *International Symposium on Web Systems Evolution* (WSE), and the 6th *International Workshop on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems* (MESOCA). In addition to the main Research track, ICSM 2012 features the *Early Research Achievements* (ERA) track, the Industry track, the Tool Demonstrations track, the PhD session, and two exciting Keynotes, given by Prof. Mauro Pezzè and by Prof. Lori Pollock. For the second time in its history, ICSM 2012 will award the Most Influential Paper, selected among those presented at ICSM 2002.

The Research track received 181 submissions. Out of them, the program committee accepted 46 papers (25%) by 126 authors from 19 countries. Each paper was reviewed by at least three members of the Program Committee. Each paper received at least three reviews and, after the authors had the possibility to submit a rebuttal, the reviews have been extensively discussed online during two weeks, and final decisions were made based on the reviews and the discussions.

The ERA track features 16 papers, the Tool Demonstrations track includes 9 tool demonstrations, the Industry track includes 8 full and 1 short paper. The ICSM program includes the 1st *Workshop on the Next Five Years of Text Analysis in Software Maintenance*.

We hope you will have a great time and an unforgettable experience at ICSM 2012 in Riva.

Paolo Tonella	**Jonathan I. Maletic**	**Massimiliano Di Penta**
ICSM 2012 General Chair	**ICSM 2012 Program Co-Chair**	**ICSM 2012 Program Co-Chair**
Fondazione Bruno Kessler	**Kent State University**	**University of Sannio**
Trento, Italy	**USA**	**Benevento, Italy**

DOCTORAL SYMPOSIUM

MARBLE: Modernization Approach for Recovering Business Processes from Legacy Information Systems

Ricardo Pérez-Castillo
Instituto de Tecnologías y Systemas de Información (ITSI), University of Castilla-La Mancha
Paseo de la Universidad 4 13071, Ciudad Real, Spain
ricardo.pdelcastillo@uclm.es

***Abstract*— The volatile IT industry often tempts companies to replace legacy information systems with new ones. However, these 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. MARBLE semi-automatically rebuilds the hidden business processes embedded in legacy information systems. MARBLE supports a business process archeology method which allows business experts to attain a rapid and meaningful understanding of the organization's business processes. MARBLE-framed techniques are based on static and dynamic analysis by considering different legacy software artifacts (e.g., source code, event logs, etc.). Through the validation of MARBLE with several industrial systems, the proposal proved to be less time-consuming and more exhaustive (since it considers the embedded business knowledge) than a manual process redesigned by experts from scratch. The main implications are that MARBLE provides maintainers with a mechanism with which to modernize legacy information systems in line with the actual business processes of an organization.**

Keywords—Business Processes; Software Modernization; Static Analisis; Dynamic Analisis; Model Transformation.

I. INTRODUCTION

In the early 1990s, Business Process Management (BPM) arose as a mechanism with which to maintain and improve the quality of processes and operations carried out by enterprises and organizations. A business process depicts a sequence of coordinated activities, together with their roles and the data involved, that must be carried out to achieve a business goal [10]. Business processes have become a key asset in organizations, since they allow them to carry out their daily operations. Business process management also helps organizations to address technological and organizational changes in order to consequently improve their competitiveness [2].

Most business processes in organizations are supported by their enterprise information systems. The optimal business process management is therefore achieved when organizations additionally combine the management of their Legacy (existing) Information Systems (LIS) [10]. The configuration management of legacy information systems is particularly important since these systems undergo a considerable amount of changes during their lifecycles. Legacy information systems change as a result of evolutionary maintenance, in which new business requirements and functionalities are incorporated into the system [8, 11].

As a consequence of the evolutionary maintenance over time, new business knowledge and rules are embedded in legacy information systems. This embedded business knowledge may not be existent anywhere else [14]. The evolution of information systems in isolation consequently affects the evolution of business processes (see scenario 1 in Figure 1). In this case, it is necessary to discover and reconstitute the underlying business processes that are currently supported by legacy information systems [6].

However, there are many organizations that currently carry out a vast amount of daily transactions through their enterprise information systems without having ever done their own business process modeling. When these organizations deal with business process modeling for the first time, a recurrent method by which to attempt this modeling is the extraction of business processes from legacy information systems [22] (see scenario 2 in Figure 1). This is owing to the fact that legacy information systems are one of the few knowledge assets in organizations that can be used to attain an accurate understanding of the actual business process.

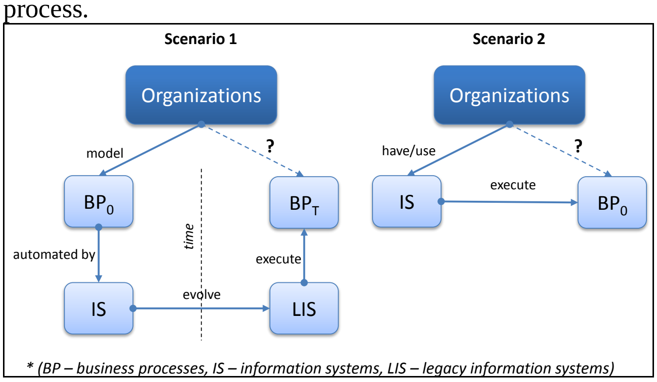

Figure 1. Scenarios to discover embedded business processes.

In both scenarios (see Figure 1), retrieving an up-to-date version of business processes from legacy information systems allows organizations to take advantage of at least two main benefits:

- **Benefits for business process modeling.** Business processes can always be up-to-date. Organizations may therefore conduct business process management by following the continuous improvement process. This kind of business process management facilitates

978-1-4673-2312-3/12/$31.00 ©2012 IEEE

an agile adaptation of business processes to meet all the changes that occur in the uncertain environment of a company. The rapid evolution of business processes allows organizations to maintain, or even improve, their degree of competitiveness [10].

- **Benefits for legacy information systems.** Legacy information systems may continue to be modernized on more occasions. A recent study by the SEI (Software Engineering Institute) states that it is first necessary to retrieve embedded business knowledge in order to modernize systems in line with the organization's business processes [12]. Organizations can thus modernize their legacy information systems whilst they align the new systems with their actual business processes. Legacy information systems are therefore evolved rather than being immediately retired and the ROI (return of investment) on such systems is improved. This is because the lifecycles of these systems are extended, which saves costs as regards new developments from scratch.

II. STATE-OF-THE-ART

There are two trends or approaches in literature that can be used to recover embedded business processes. In the first approach, several business process mining techniques and algorithms have been proposed [22]. Process mining takes event logs recorded during the execution of enterprise information systems as input. All these approaches assume the presence of event logs and they are not concerned with obtaining event logs from systems. In this respect, other works attempt to obtain event logs from different kinds of Process-Aware Information Systems (PAIS). For example, *Günther et al.* [5] provide a generic import framework with which to obtain event logs, while *Ingvaldsen et al.* [7] focus on Enterprise Resource Planning (ERP) systems to obtain event logs from the SAP's transaction data logs.

Unfortunately, the main problem of business process mining is that there are no well-proven techniques with which to obtain event logs from traditional legacy information systems (i.e., non-process-aware information systems) without an in-built logging mechanism. This is an important challenge since there are a vast amount of legacy information systems in industry that do not explicitly support business processes (i.e., non-process-aware information systems). The provision of techniques with which to obtain event logs from traditional legacy information systems may facilitate the application of all the efforts (techniques and algorithms) made in the business process mining field.

In the second approach, several proposals consisting of the analysis and inspection of different software artifacts have been proposed in literature. In this Thesis, these kinds of approaches have been termed as *Business Process Archeology* [15]. While business process mining relies on event logs produced at runtime, business process archeology analyzes different artifacts (e.g., source code, databases, or even event models) by means of reverse engineering in order to obtain the underlying embedded business processes.

Some of these kinds of proposals are based on the static analysis of source code. One of these proposals is that of *Zou et al.* [24], which statically analyses the source code and applies a set of heuristic rules to discover business processes. *Paradauskas* [14] present a framework to recover business knowledge through the static analysis of data stored in databases. Marcus et al. [13] and *Poshyvanyk et al.* [19] apply a concept location technique locating business concepts into source code. The major disadvantage of these proposals (which solely use static analysis as a reverse engineering technique) is that run-time information is ignored. Other methods using dynamic analysis are therefore proposed. *Di Francescomarino et al.* [4] discover business processes through the execution of graphical user interfaces in Web applications. *Cleve* [3] combines static and dynamic analysis of databases to obtain data-intensive business knowledge. *Van Geet et al.* [23] use concept analysis to extract business rules from COBOL mainframe systems.

The main weakness of the second approach, business process archaeology, 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 business process archaeology techniques in large-scale projects is therefore in doubt [1]. In fact, a 2005 study [21] states that 50% of reengineering projects (based on reverse engineering techniques) fail owing to the lack of standardization and automation, which often leads to overruns in costs.

Standardization and automation challenges limit the applicability of the aforementioned techniques to large and complex 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. The Meta Object Facility (MOF), a standard adopted from the OMG, is used for this purpose to provide both a metadata management framework and a set of metadata services to enable the development and interoperability of model and metadata driven systems. MOF has made a significant contribution to some of the core principles of both the MDA and ADM.

The research hypothesis is therefore that ADM can address the standardization challenge since it treats all the artifacts involved in the recovery process homogeneously, i.e., as models. The automation challenge may therefore be addressed thanks to the automated transformations together with the reuse of models at different abstraction levels. Thereby, the main research goal of the thesis is: "to define an ADM-based framework with which to discover and reconstitute business processes from legacy information systems".

III. MARBLE

MARBLE is an ADM-based framework which uses the KDM (Knowledge Discovery Metamodel) [9] to represent all the knowledge involved, the cornerstone specification of the ADM initiative (see Figure 2). The KDM specification defines a standard metamodel that facilities an integrated representation and management of all the knowledge extracted by reverse engineering from all the different software artifacts of legacy information systems. MARBLE extracts legacy knowledge which is integrated into a common KDM repository and it then gradually transforms this knowledge into business process models. The abstraction gap between software artifacts (such as source code) and business process models must be reduced progressively. MARBLE therefore defines four kinds of models at four different abstraction levels. MARBLE additionally establishes, according to the ADM initiative, three model transformations between the four abstraction levels.

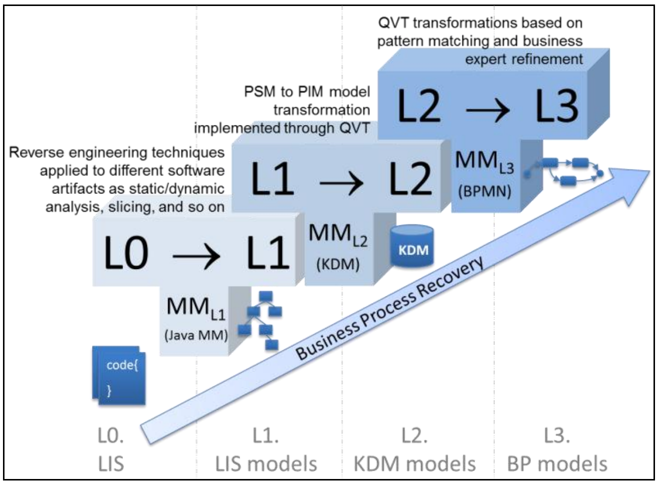

Figure 2. An overview of the MARBLE framework

The four generic abstraction levels proposed in MARBLE are the following:

- **Level L0.** This is the lowest level of abstraction. L0 represents the legacy information system in the real world as a collection of different software artifacts (e.g. source code, database, documentation, etc.).
- **Level L1.** This level consists of several specific models, i.e., one model for each different software artifact involved in the archeology process, such as source code, database, user interface, and so on. These models are considered to be PSM (Platform-Specific Models) since they depict the software artifacts according to their specific technology or platforms.
- **Level L2.** This level consists of a common PIM (Platform-Independent Model) which represents the integrated view of the set of PSM models at L1. The standard KDM metamodel is used for this purpose, since it makes it possible to model all the artifacts of the legacy system in an integrated and technological-independent manner.
- **Level L3**. This is the highest level of abstraction which represents a computational independent model of the system. L3 depicts the business processes retrieved from the knowledge concerning legacy information systems represented in the KDM repository at L2. Business process models at L3 are represented according to the BPMN (Business Process Model and Notation) metamodel proposed by the OMG.

MARBLE additionally defines the following three model transformations between the four abstraction levels in order to incrementally obtain business process models:

- **Transformation L0-to-L1.** The first transformation obtains PSM models from each legacy software artifact. Traditional reverse engineering techniques [1] such as static analysis, dynamic analysis, subsystem decomposition, and so forth, are used to discover and extract knowledge from a software artifact and build the respective PSM model (see Figure 2). Each PSM model is represented according to a specific metamodel, e.g., a Java metamodel can be used to model source code, or an SQL metamodel can be used to represent the database schema of a legacy information system.
- **Transformation L1-to-L2.** The second transformation consists of a set of model transformations with which to obtain a common PIM model based on the KDM metamodel. This PIM model is built from the PSM models from level L1. The L1-to-L2 transformation can be implemented by means of QVT (Query / Views / Transformations) standard proposed by the OMG. The QVT scripts translate instances of metaclasses of respective metamodels used at L1 and metaclases of the standard KDM metamodel. The transformation from the legacy information system (at L0) to the KDM model (at L2) is made through the level L1 since in many cases, the platform-specific knowledge at the intermediate level L1 might be used to infer more business knowledge at L2. The semantic gap between the legacy information system and the respective KDM model is thus reduced incrementally.
- **Transformation L2-to-L3.** The last transformation is based on a set of business patterns and a matching technique with which to identify them. When a particular structure (according to a pattern) is detected in the KDM model (at L2), that pattern indicates the respective structure of elements that must be built into the business process model (at L3). MARBLE framework implements this pattern matching technique through a model transformation also written in QVT. The L2-to-L3 transformation can additionally be aided by business experts who know the organization. Experts provide external information to refine the first sketch of the business processes obtained after pattern matching.

MARBLE can be used as standard guidelines to provide concrete techniques with which to obtain business processes from legacy information systems. Each concrete technique is characterized by (i) the legacy software artifacts that are considered at L0; (ii) the reverse engineering technique used to extract embedded business knowledge from these artifacts; (iii) the set of business patterns employed in the last transformation; and (iv) whether business expert post-intervention does or does not take place.

This PhD Thesis provides two concrete techniques, which are summarized in Figure 3. Although both techniques consider legacy source code as the input software artifact, these techniques address business process recovery from two different perspectives. On the one hand, the first technique statically analyzes the source code and represents the embedded knowledge in a KDM repository, which is then analyzed by applying a pattern matching technique to discover business process models. On the other hand, the second technique analyzes the source code at runtime to obtain event logs that are then analyzed in order to retrieve business processes. While the first technique follows the business process archaeology approach, the second follows that of business process mining.

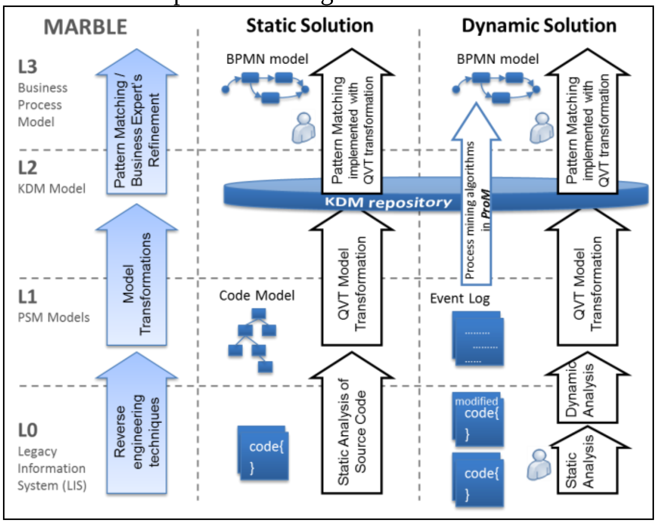

Figure 3. Static and dynamic techniques framed in MARBLE.

A. Static Technique

The static technique framed in MARBLE considers legacy source code as the key software artifact at L0 and static analysis as a reverse engineering technique with which to extract the embedded knowledge at L0 and represent it at L1. The static analysis of source code consists of the sequential, syntactic inspection of the all source code files.

This technique is supported by a tool that has been especially developed for this purpose. The tool has a module with which to analyze source code files (Java files in particular) and builds an abstract syntax tree of the source code, i.e. a code model at level L1 (see Figure 3). The advantage of the static approach is that the syntactic parser that is used to analyze the source code is easy and non-time-consuming to build. Moreover, the tool implements a QVT model transformation to transform the code model into a KDM model at L2. The KDM model provides a standard inventory of all the source code elements and their relationships, and can therefore be used for other business process recovery techniques framed in MARBLE or any other software modernization activity.

The static technique (see Figure 3) provides a set of business patterns to support the discovery of business processes at L3 [17]. The set of patterns is divided into three categories:

- **Structural patterns**, which deal with the structural elements (e.g., business activities or tasks) and their combinations (such as sequence flows or gateways). There are four patterns which transform: (i) packages, compilation units or other aggregation units (e.g. Java classes or interfaces) into business process diagrams; (ii) methods into tasks; (iii) calls between methods into sequence flows between the respective tasks; and (iv) conditional branching, e.g. if-then-else or switch statements, into exclusive gateways that branch the main sequence flow.
- **Data patterns**, which deal with data objects and how these objects are related to other structural elements. There are two data patterns, which transform: (i) read program variables for a method into a data object with an association to the respective task; and (ii) written program variables into data objects with an association from the task.
- **Event patterns**, which build all the elements involved in the event management. There are three patterns, which transform: (i) the start method into a start event and sequence flow to the respective task; (ii) end tasks into sequence flows from such tasks to an end event; and (iii) conditional calls into sequence flows with an intermediate conditional event.

The pattern recognition and generation of business process models is implemented in the tool developed by means of QVT transformations [16]. The tool can also be used by business experts to refine the business processes discovered, since it provides graphical model editors for both KDM and business process models.

B. Dynamic Technique

The dynamic technique framed in MARBLE (see Figure 3) considers knowledge derived from system execution. Hence, the reverse engineering technique considered in the L0-to-L1 transformation combines static analysis and dynamic analysis.

The static pre-analysis injects statements into certain places of the source code to register execution events in a log when these statements are executed (see Figure 3). Each event recorded in the log specifies the execution of an underlying business task supported by a certain piece of source code. According to [18], the injection of these tracing statements into the source code therefore entails five key challenges that must be addressed: (i) process definitions are implicitly described in legacy source code and, thus, it is not obvious which events should be recorded in the event log; (ii) the granularity of callable units of an information system and activities of a business process often differs; (iii) legacy code not only contains business activities, but also technical aspects which have to be discarded from target business processes; (iv) since traditional systems do not explicitly define processes, it is necessary to establish when a process starts and ends; (v) finally, owing to the missing process-awareness, it is not obvious how the business activities and process instances executed should be correlated.

This technique partially addresses these challenges by considering information provided by business experts and system analysts. This information is necessary to reduce potential noise in the event log caused by the aforementioned challenges. Although manual intervention by experts might appear to be a time-consuming task, this task is less time-consuming and even less error-prone than business process modeling from scratch.

After static pre-analysis, the dynamic analysis records events during system execution (see Figure 3). When the instrumented code is executed, the injected statements invoke a function that records the respective event in the event log. The event log can then be used to discover the business processes by taking the system execution information into account. In the same manner as the static technique, this dynamic technique transforms the event log at L1 into a KDM model at L2, and then transforms this model into business process models at L3. However, this technique reuses existing process mining techniques such as those supported by *ProM* tool [22] in order to discover business process models (at L3) from event logs generated after dynamic analysis (see Figure 3).

IV. Lessons Learned

The key lessons learned from this research are mainly two: the first one is related to the fieldwork necessary to obtain advances in the research, and the second one is related to the limitations of the proposal.

A. *Fieldwork is necessary*

During the Thesis, the office work is very comfortable and bearable. However, research proposals on the role have to be empirically validated through a hard fieldwork. Particularly, in order to ensure the applicability of our proposal, we always want to validate each tiny proposal with industrial systems in exploitation by means of supporting tools. It implied making a great effort during this Thesis, since the investigation was combined with the implementation of supporting tools in parallel, and we applied our techniques to several systems at different companies and organizations in various countries. This additionally entails a personal offering due to the various pre-doctoral stays and several trips. However, I believe that the fieldwork was the most contributing part during the thesis. In fact, I learnt more with the any observation of faults or unexpected results obtained in industrial case studies than writing or reading papers at office.

B. *Limitations of our Work*

During the attendance at different conferences and revisions of journal submissions, I was able to detect the most common and recurrent criticisms of our work. In order to be honest, I would like to share all these limitations with the research community.

One of the most frequent clarification questions was related about the possibility (or not) of discovering cross-cutting business processes from different applications or subsystems integrating a whole enterprise information system, which are presents in some companies. Unfortunately, this challenge is outside of the scope of the thesis. However, it is an important open issue to be addressed in the future according to the problems of *delocalization* and *interleaving* of the embedded business knowledge [20]. 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 limitation revealed by research community is related to the time spent on manual post-intervention to refine the first sketch of business process models that were automatically obtained. Although the case studies show that our proposal is less error-prone and time-consuming than manual modeling from scratch, the manual time should be reduced in future work.

V. Future Work

The future research lines of this PhD thesis are related to the mentioned limitations. According to the discovery of cross-cutting business processes, a future research line will be devoted to apply MARBLE to large and complex enterprise information systems with a large portfolio of applications in order to obtain cross-cutting business processes from heterogeneous, pervasive systems. Clustering techniques will therefore be provided to combine atomic business processes into a more complex one.

In line with the second threat to the validity of this Thesis, the first sketch of business processes obtained by reverse engineering is often fine-grained and very intricate. The development of refactoring techniques will therefore help to obtain more accurate, understandable and less complex business processes. Refactoring techniques will be employed to obtain alternative business processes, thus reducing the complexity of the business process models retrieved while preserving the external behavior. Thanks to this automatic refinement, the effort related to manual post-intervention by business experts will be minimized or even avoided, and the accuracy of business process models will in turn be increased.

Both future research lines have already begun to be addressed. The first research line, related to business process clustering, is being addressed by a student in his BSc project, and the second line, related to business process refactoring, is being carried out by an MSc student in her master thesis, who will probably become in a PhD student.

Finally, in line with these future research lines, we have planned to develop and maintain supporting tools. Since MARBLE Tool has been released as an Eclipse™ plug-in, it can be easily extended and integrated into other related tools, e.g., tools supporting both future MARBLE-based techniques and other related modernization and mining tools.

VI. CONCLUSIONS

The main contribution of this PhD Thesis is MARBLE, a generic ADM-based framework with which to obtain business processes from legacy information systems. The main implication of this contribution is that MARBLE is generic and based on international standards, and is not therefore limited to only one concrete technique or platform. This is the key difference with regard to related works in literature, which often present *ad hoc* or/and non-standard solutions.

Both the static and the dynamic techniques demonstrate the possibility of defining concrete techniques from MARBLE. Unlike most proposals in literature, both techniques have been successfully applied to various real-life legacy information systems, thus demonstrating its applicability in real environments. In fact, the conduction of different industrial case studies is another important contribution since these studies facilitates the transference of technology to industry.

Moreover, there exist implications regarding all the possible usages of business processes obtained by applying the MARBLE-based techniques. As was mentioned before, business processes discovered from legacy information systems can be used with two main objectives. Firstly, the business processes retrieved represent the actual, current processes carried out by an organization. These business processes may be used to discover (if it is the first discovery) or align business processes (when a previous version exists) with the actual business activities of a company. The second usage of business processes consists of the software modernization of legacy information systems. When a legacy information system is modernized, evolved or migrated, the whole behavior must be preserved in the new version. The research proposal facilitates the extraction of business rules (through business process discovery) which must be incorporated into new versions of information systems. A further implication of this Thesis is consequently its contribution towards extending the lifespan of legacy information systems, and the fact that it therefore also helps to improve the ROI.

ACKNOWLEDGMENT

I would like to express my gratitude to my supervisors Ignacio García and Mario Piattini. Additionally, this PhD Thesis has been funded by the Spanish FPU program.

REFERENCES

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

[2] Castellanos, M., K.A.d. Medeiros, J. Mendling, B. Weber, and A.J.M.M. Weitjers, Business Process Intelligence, in Handbook of Research on Business Process Modeling, J. J. Cardoso and W.M.P. van der Aalst, Editors. 2009, Idea Group Inc. p. 456-480.

[3] Cleve, A. Program analysis and transformation for data-intensive system evolution. IEEE International Conference on Software Maintenance (ICSM). 2010. p. 1-6.

[4] Di Francescomarino, C., A. Marchetto, and P. Tonella, Reverse Engineering of Business Processes exposed as Web Applications, in 13th European Conference on Software Maintenance and Reengineering (CSMR'09). 2009, IEEE C. S.: Fraunhofer IESE, Kaiserslautern, Germany. p. 139-148.

[5] Günther, C.W. and W.M.P. van der Aalst, A Generic Import Framework for Process Event Logs. Business Process Intelligence Workshop (BPI'06), 2007. LNCS 4103: p. 81-92.

[6] Heuvel, W.-J.v.d., Aligning Modern Business Processes and Legacy Systems: A Component-Based Perspective (Cooperative Information Systems). 2006: The MIT Press.

[7] Ingvaldsen, J.E. and J.A. Gulla, Preprocessing Support for Large Scale Process Mining of SAP Transactions. Business Process Intelligence Workshop (BPI'07) 2008. LNCS 4928: p. 30-41.

[8] ISO/IEC, ISO/IEC 14764:2006. Software Engineering -- Software Life Cycle Processes -- Maintenance. http://www.iso.org/iso/catalogue_detail.htm?csnumber=39064. 2006, ISO/IEC.

[9] ISO/IEC, ISO/IEC DIS 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. 2009, ISO/IEC. p. 302.

[10] Jeston, J., J. Nelis, and T. Davenport, Business Process Management: Practical Guidelines to Successful Implementations. 2nd ed. 2008, NV, USA: Butterworth-Heinemann (Elsevier Ltd.). 469.

[11] Lehman, M.M., Program evolution. Information Processing & Management, 1984. 20(1-2): p. 19-36.

[12] Lewis, G.A., D.B. Smith, and K. Kontogiannis, A Research Agenda for Service-Oriented Architecture (SOA): Maintenance and Evolution of Service-Oriented Systems. 2010, SEI. p. 40.

[13] Marcus, A., A. Sergeyev, V. Rajlich, and J.I. Maletic, An Information Retrieval Approach to Concept Location in Source Code, Working Conference on Reverse Engineering. 2004, IEEE C. S. p. 214-223.

[14] Paradauskas, B. and A. Laurikaitis, Business Knowledge Extraction from Legacy Information Systems. Journal of Information Technology and Control, 2006. 35(3): p. 214-221.

[15] Pérez-Castillo, R., I.G.-R. de Guzmán, and M. Piattini, Business Process Archeology using MARBLE. Information and Software Technology, 2011. 53: p. 1023–1044.

[16] Pérez-Castillo, R., M. Fernández-Ropero, I. García Rodríguez de Guzmán, and M. Piattini, MARBLE. A Business Process Archeology Tool, in 27th IEEE International Conference on Software Maintenance (ICSM'11). 2011, IEEE C. S. p. 578-581.

[17] Pérez-Castillo, R., I. García-Rodríguez de Guzmán, O. Ávila-García, and M. Piattini, Business Process Patterns for Software Archeology, in 25th Annual ACM Symposium on Applied Computing (SAC'10). 2010, ACM: Sierre, Switzerland. p. 165-166.

[18] Pérez-Castillo, R., B. Weber, I. García Rodríguez de Guzmán, and M. Piattini, Generating Event Logs from Non-Process-Aware Systems Enabling Business Process Mining. Enterprise Information System Journal, 2011. 5(3): p. 301–335.

[19] Poshyvanyk, D. and A. Marcus, Combining Formal Concept Analysis with Information Retrieval for Concept Location in Source Code, in International Conference on Program Comprehension (ICPC'07). 2007, IEEE C. S.: Banff, Alberta, Canada. p. 37-48.

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

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

[22] van der Aalst, W., H. Reijers, and A. Weijters, Business Process Mining: An Industrial Application. Information Systems, 2007. 32(5): p. 713-732.

[23] Van Geet, J. and S. Demeyer. Feature location in COBOL mainframe systems: An experience report. in Software Maintenance, 2009. ICSM 2009. IEEE International Conference on. 2009.

[24] Zou, Y. and M. Hung, An Approach for Extracting Workflows from E-Commerce Applications, International Conference on Program Comprehension. 2006, IEEE C. S. p. 127-136.