MARBLE. A Business Process Archeology Tool

Ricardo Pérez-Castillo, María Fernández-Ropero, Ignacio García-Rodríguez de Guzmán and Mario Piattini
Alarcos Research Group, University of Castilla-La Mancha
Paseo de la Universidad 4 13071, Ciudad Real, Spain
{ricardo.pdelcastillo, marias.fernandez, ignacio.grodriguez, mario.piattini}@uclm.es

Abstract—Modernization of legacy information systems is usually triggered by the need of introducing new business requirements or due to the technology obsolescence. During modernization software projects, there exists a lot of business knowledge that was embedded in source code owing to progressive maintenance, which is not present anywhere else. In order to preserve embedded business knowledge this paper presents MARBLE, a tool to recover business processes from legacy information systems. MARBLE provides an extensible framework, thus it is developed as an Eclipse™ plugin to ensure its future extension. So far, MARBLE supports Java-based system and focuses mainly on legacy source code. To facilitate its adoption in the industry, MARBLE has been applied in some real-life modernization projects, which demonstrated that MARBLE is suitable to retrieve business processes and facilitated its continuous improvement to reach an appropriate maturity level.

Keywords—Reverse Engineering, Business Process, Model-Driven Development, Knowledge Discovery Metamodel.

I. INTRODUCTION

Reverse engineering techniques have become very important within the maintenance process providing several benefits. Firstly, reverse engineering allows maintainers to retrieve abstract representations to facilitate the comprehension of different legacy systems. For example, it focuses on relational databases [6], aspect oriented systems [3], quality of the system design [8], links between e-mail and source code [2], and so on. Secondly, abstract representations obtained by reverse engineering from legacy systems can be refactored to improve their maintainability or add new functionalities to evolve legacy systems.

To address the mentioned maintenance activities, reverse engineering techniques are nowadays well-supported by tools which often obtain artifacts at system design abstraction level (e.g., class or sequence diagrams from source code) [5]. However, software engineering industry is demanding additional reverse engineering techniques and tools to retrieve business-aware artifacts at higher abstraction level [11]. In fact, the Software Engineering Institute (SEI) argues that business rules recovery is the cornerstone to evolutionary maintenance towards modern paradigms like Service Oriented Architecture (SOA) [12].

To meet these demands, business process archeology have emerged as a set of techniques and tools to recover business processes from source code [15]. Maintenance benefits of business process archeology are that they preserve business behavior buried in legacy source code and it retrieves business processes providing more opportunities for refactoring (due to the higher abstraction level).

There are some techniques in literature that support business process recovery. For example Zou et al. [19] recover workflows by statically analyzing source code and applying some heuristic rules to discover business knowledge. Paradauskas et al. [14] retrieve business knowledge through the inspection of the data stored in databases. These studies rely solely on static analysis. Thereby, other solutions have been suggested based on dynamic analysis. For example, Di Francescomarino et al. [7] consider graphical user interfaces of web applications to discover business processes. Cai et al. [4] combine requirement reacquisition based on use cases with dynamic and static analysis techniques. Finally, Van der Aalst et al. [18] focus on mining business processes from event logs registered during system execution.

This paper particularly presents MARBLE™, a technique and tool, supporting business process archeology [15]. MARBLE™ is an extensible framework, although it so far supports static analysis of legacy source code in a similar way than technique proposed by Zou et al. However, MARBLE™ applies a set of business patterns formalized by model transformations. The main difference regarding mentioned proposals is that MARBLE™ is easily automatable and highly extensible due to it follows model-driven architecture principles (i.e., it considers all involved software artifacts as models, and it provides formal transformations to move models between different abstraction levels).

MARBLE™ tool [1] is implemented as an Eclipse™ plugin improving its extension and integration with other techniques and tools as well as to facilitate its adoption by the industry. The tool creates and manages an entire repository by integrating code models from legacy information systems. After that, the tool allows maintainers to discover, visualize and edit business process models.

The effectiveness and suitability of the tool has been demonstrated through several industrial case studies which the tool has been applied to a healthcare system, an e-government system, enterprise systems, among other kinds of legacy systems.

The remaining of the paper is organized as follows: Section II explains MARBLE™ technique supported by the tool. Section III presents in detail functionalities and implementation details of MARBLE™ tool. Section IV briefly summarizes the case studies which the tool has been used. Finally, Section IV discusses the conclusions of this work.
II. BUSINESS PROCESS ARCHAEOLOGY PROCEDURE

Business process archeology [15] studies the business processes in an organization by analyzing the existing software artifacts. The objective is to discover the business forces that motivated the construction of the enterprise information systems. On the one hand, traditional archeologists investigate several artifacts and situations, trying to understand what they are looking at, i.e., they must understand the cultural and civilizing forces that produced those artifacts. Similarly, a business process archeologist analyzes different legacy artifacts such as source code, databases and user interfaces and then tries to learn what the organization was thinking to understand why the organization developed the information system in a particular way. Business process archeology initiative is being progressively supported by new reverse engineering techniques and tools to retrieve and elicit the embedded business knowledge. Particularly, this paper focuses on MARBLE™ (Modernization Approach for Recovering Business processes from LÉgacy Systems) [15]. MARBLE™ is an extensible framework based on the ADM (Architecture-Driven Modernization) initiative proposed by the OMG (Object Management Group). ADM advocates carrying out reengineering processes by considering model-driven development principles.

In addition, MARBLE™ supports the KDM (Knowledge Discovery Metamodel) [10] 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 way. Thus, that legacy knowledge is gradually transformed into business processes. For this purpose, MARBLE™ is divided into four abstraction levels with three transformations among them (see Figure 1):

**Level L0.** This level represents the legacy information system in the real world, and is the source system 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 [5] 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 (Platform-Specific Model) 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 (Platform-Independent Model) 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 (Computational-Independent Model) 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 [16]. This pattern matching is implemented through a QVT transformation [17].

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 archeology. In addition, the business process redesign by experts from scratch might discard meaningful business knowledge that is only embedded in legacy information systems.

III. IMPLEMENTATION DETAILS

MARBLE™ technique is supported by a tool with the same name. MARBLE™ tool supports entirely the three model transformations presented between the four MARBLE™ levels. In addition, MARBLE™ allows maintainers and business experts to visualize and modify the first sketch of business processes retrieved after the last transformation.

MARBLE™ tool was initially developed as a desktop application. However, due to the extensible nature of MARBLE™, it is now online available as an Eclipse™ plugin [1]. This fact ensures, in the future, easy extension and integration with other maintenance tools.
A. Technologies involved

Due to the fact that MARBLE™ technique provides a generic and extensible framework, MARBLE™ tool is firstly developed to support Java-based legacy systems. The tool is implemented through three key technologies. The first technology is JavaCC, which is a parser and scanner generator for Java. It is used to develop a Java static analyzer in the first transformation to obtain code models. Parsers for other programming languages may be developed in the future using JavaCC technology.

The second technology consists of two related Eclipse™ frameworks: EMF (Eclipse Modeling Framework) and GMF (Graphical Modeling Framework). EMF is a modeling framework and code generation facility for building tools and other applications based on ECORE-compliant models. ECORE is a meta-metamodel, i.e. it is the metamodel proposed by the Eclipse™ platform to define metamodels. EMF provides tools to automatically produce, from an ECORE-compliant metamodel, the source code of an editor to enable viewing and editing of the respective models. Moreover, GMF is used together with EMF to generate graphical editors from ECORE-compliant metamodels.

Finally, the third technology is QVT, which is a language specially developed to formalize model transformations. Model transformations involved in MARBLE™ tool are particularly implemented using QVTr (QVT Relations, the declarative language part of the QVT). In addition, model transformations are executed through MediniQVT™, an open source QVTr transformation engine [9]. This transformation engine is integrated within the MARBLE™ plugin.

B. Main Functionalities

MARBLE tool functionalities are organized into four main modules. The first module is the project explorer (see Figure 2 A) which groups the models within each MARBLE level. Each model transformation can be done by selecting a whole level or a particular set of models.

The second model is the editor frame (see Figure 2 B) which visualizes the different kinds of models (i.e., java files, code models, KDM models and BPMN models). For example, the screenshot shows a business process model in the graphical editor, which additionally provides a palette of elements to facilitate the graphical edition of the model.

The third module provides a special tree editor with which visualize code models at L1 obtained by the Java parser developed (see Figure 2 C). This editor additionally enable feature location since it links code model elements of the abstract syntax tree with the java source code (i.e., an element of the code model can be located by clicking on a certain line of source code).

Finally, the fourth module provides additional relevant information about the particular model opened in the second module (see Figure 2 D). This module particularly provides an outline view as well as a property set view.
Figure 3. Effectiveness summary of case studies

IV. CASE STUDIES

MARBLE has been applied to several industrial case studies to recover business processes from a wide variety of legacy information systems. The conduction of these industrial case studies has allowed improving the tool and refining the MARBLE technique. So far, MARBLE has been used with six legacy systems in total: (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.

These studies evaluate the effectiveness and efficiency of the MARBLE 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 tasks and other related business process elements such as sequence flows, data objects and gateways. On the other hand, efficiency is evaluated through the time spent on the recovery as well as the MARBLE 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 MARBLE retrieves a great number of business activities although a few of them could be erroneous. In addition, a little set of activities could not be retrieved. Anyway, the results are appropriate regarding benchmark values around 0.5 [13].

V. CONCLUSIONS

MARBLE is a tool to retrieve business processes from legacy source code. MARBLE provides an extensible framework and it is therefore implemented as an Eclipse™ plugin. So far, MARBLE provides a Java parser to obtain code model, which are transformed and integrated in a model repository according to the KDM standard. After that, KDM model are transformed to business process models by applying business pattern recognition. MARBLE has already been applied to real-life modernization projects where business process recovery was a mandatory requirement to preserve the embedded business knowledge.

ACKNOWLEDGMENT

This work was supported by the Spanish FPU Programme, and the R&D projects ALTAMIRA (JCCM, PI1209-0106-2463), MOTERO (JCCM and FEDER, PEI11-0366-9449) and PEGASO/MAGO (TIN2009-13718-C02-01).

REFERENCES