Simula Research Laboratory
Software Engineering Department
Technical Report Series

Proceedings
International Workshop on

# Reverse Engineering Models from Software Artifacts — R.E.M. 2009 —

October 15, 2009, Lille, France
co-located with 16th Working Conference on Reverse Engineering
(WCRE 2009)

Organized by
Leon Moonen and Tarja Systä

Report SIMULA-SE-2009-07

SIMULA-SE-2009-07

Published, produced and distributed by:

Software Engineering Department
Simula Research Laboratory
P.O. Box 134
1325 Lysaker
Norway

For more information about the Software Engineering Department:
http://www.simula.no/se/

# Table of Contents

# Overview & Introduction

## R.E.M. 2009 Program

| | |
|---|---|
| 14:30 | opening |
| 14:35 | **topic: Round-trip-engineering and reverse engineering** |
| | Construct to Reconstruct Reverse Engineering Java Code with JaMoPP<br>*F. Heidenreich, J. Johannes, M. Seifert, and C. Wende* |
| | Deriving EMF Models from Java Source Code<br>*A. Wolff and P. Forbrig* |
| 15:05 | **topic - Towards reconstructing CIM models** |
| | Reverse Engineering Domain Models from Source Code<br>*D. Ratiu* |
| | MARBLE: A Modernization Approach for Recovering Business Processes from Legacy Systems<br>*R. Pérez-Castillo, I. García-Rodríquez de Guzmán, M. Piattini and O. Ávila-García* |
| 15:35 | **topic: Applying MDE technology** |
| | Measuring Discovered Models<br>*J-S Sottet, F. Jonault, J. Bézivin, M. Venisse, and V. Fady*<br>Coupling Discovered Models<br>*J-S Sottet, F. Jonault, and J. Bézivin* |
| 16:00 | break |
| 16:30 | **continuation of topic: Applying MDE technology** |
| | Megamodeling Software Platforms: Automated Discovery of Usable Cartography from Available Metadata<br>*V. Mahé, F. Jouault, and H. Bruneliere* |
| 16:45 | **topic: Migration to MDE** |
| | Summary of NW-MoDE special session on Migrating to MDE<br>*Leon Moonen and Tarja Systä* |
| | On the Application of Model Driven Engineering Technologies to Legacy Systems<br>*B. Trask and A. Roman* |
| 17:15 | general discussion |
| 17:45 | workshop ends |

## Workshop Background

Model driven engineering (MDE) has become an increasingly popular approach to deal with the complexity of modern day software engineering. In MDE, besides supporting communication and comprehension among different stakeholders, the high level models are no longer considered as informal drafts or initial plans, but they *are* the system specified. In addition, model transformations can be used to keep all the models of a software system consistent and up-to-date during maintenance and development. That is, the maintenance activities are assumed to be performed on the models, from which the executable code is generated through model transformation steps.

In addition to greenfield development, where model driven approaches can be adopted from scratch, there is a strong need to investigate approaches that can help to bring the benefits of model driven engineering to continue the development and maintenance of the large amounts of existing software systems.

## Motivation, Topics and Goals

The purpose of R.E.M. is to bring together practitioners, researchers, academics, and students to discuss the state-of-the-art of model reconstruction based on the reverse engineering of existing software artifacts including, but not limited to, source code, build and configuration files, versioning data, and design documentation. In R.E.M. we focus especially on model reconstruction methods and tools that aim at model-driven software evolution and development processes.

The goal of the workshop is to share experiences, consolidate successful techniques, collect guidelines and best practices, and identify open issues for future work.

R.E.M. topics of interest include, amongst others:

- Challenges in the reconstruction of structural models
- Challenges in the reconstruction of behavioral models
- Opportunities and disadvantages of various types of artifacts
- Achieving meaningful abstractions
- Model reuse and evolution
- Reversed model transformation techniques
- Quality of the constructed models
- Reconstruction and/or migration pitfalls
- Distributed systems / Crossing system boundaries
- Case studies in model reconstruction
- Experience reports of migration to MDE
- Tool support for the above-mentioned activities

## Workshop Organizers

R.E.M. 2009 is organized by:

- Leon Moonen (Simula Research Laboratory, Norway)
- Tarja Systä (Tampere University of Technology, Finland)

# MARBLE: A Modernization Approach for Recovering Business Processes from Legacy Systems

Ricardo Pérez-Castillo, 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, ignacio.grodriguez, mario.piattini}@uclm.es

Orlando Ávila-García
Open Canarias, S.L.
C/ Elías Ramos González, nº 4 - Oficina 304 38001,
Santa Cruz de Tenerife, Spain
orlando@opencanarias.com

***Abstract*—In business processes, companies have found an ally to improve their competitiveness. The performance of these companies depends on their information systems, which may have evolved separately from the business processes. In order to align the legacy systems and the business processes, it is necessary to recover the current business processes hidden in legacy systems. Reverse engineering toward business logic is a problem that companies and academics have been trying to solve for many years. In order to contribute to finding a solution to this problem, this paper proposes MARBLE, a model-driven approach to obtain models representing preliminary schemas of business processes. For this purpose, MARBLE is divided into four abstraction levels with three model transformations between them. This paper also presents a tool to support MARBLE in a real-life case study.**

***Keywords—ADM, Business Processes, Model Transformation.***

## I. INTRODUCTION

The interest among organizations in getting to know their *Business Processes* (BP) has increased, since they are now considered to be a key resource (i) for their daily performance and (ii) for improving their competitiveness. Indeed, *Information Systems* (IS) support most of the business logic defined in the BPs [4]. However, the uncontrolled maintenance of LIS has two effects: (1) much unknown business logic is embedded in the LIS [6]; and (2) BPs implemented in the LIS are not aligned with the BPs that really govern the organization, with all the consequences that this entails. Thus, this business logic hidden in the source code lines must be recovered through reverse engineering processes. Recovery business logic has two main advantages:

- An organization can know the entire set of truthful BPs including the BPs hidden in the LIS. Thus the organization can implement an optimal management of their BPs in order to improve its competitiveness.
- Evolutionary maintenance based on re-engineering processes can be carried out in order to obtain new IS aligned with the recovered BPs.

This paper presents MARBLE (*Modernization Approach for Recovering Business process from LEgacy systems*), an approach to contribute to the well-known effort to recover the business logic of organizations. At this point, MARBLE, which follows the ADM approach (*Architecture-Driven Modernization*) [9], focuses on outlining preliminary schemas of BPs from LIS. ADM is the concept of modernizing existing systems with a focus on all aspects of the current system architecture and the ability to transform current architectures to target architectures [8]. ADM advocates carrying out re-engineering processes following the principles of model-driven development, taking into account all the involved artifacts as models and implementing transformations between them.

MARBLE focuses on the reverse engineering stage of ADM. It consists of a path through four abstraction levels to obtain BPs: L0 - legacy system; L1 – views of the system through specific models; L2 - integrates those models into an independent model; and L3 - business process model.

In order to represent the independent model on level L2, MARBLE uses the *Knowledge Discovery Metamodel* (KDM). KDM is an ADM specification proposed by the Object Management Group (OMG) and has been recognized as the standard ISO 19506 [5]. KDM provides a common repository structure that makes it possible to exchange information about existing software assets in LIS. KDM can be compared with the *Unified Modeling Language* (UML) standard; while UML is used to generate new code in a *top-down* manner, the ADM processes involving KDM start from the existing code and build a higher level model in a *bottom-up* manner [7].

A preliminary version of MARBLE has already been addressed. Additionally, a tool to support the proposal has been developed. Thus a case study was conducted by means of this tool in order to validate MARBLE.

The remainder of this paper is structured as follows: Section II summarizes the related works; Section III provides a detailed presentation of MARBLE, the proposed approach to recover BPs; Section IV briefly shows some details of the tool developed to support MARBLE; Section V presents a case study to validate the process; and finally, Section VI presents the conclusions of this work.

## II. RELATED WORK

The recovery of business knowledge is a common challenge addressed in the literature. However, the recovery of BP from LIS following the ADM approach has not been

widely studied. *Zou et al* developed a framework based on a set of heuristic rules for extracting business processes following an MDA approach by means of the static analysis of legacy source code [16]. However, the legacy source code is not the only artifact considered to obtain BPs. *Di Francescomarino et al* consider user interfaces [2], and even the data stored in the application database has been used [12, 13]. *Ghose et al* propose a set of text-based queries in source code and documentation for extracting business knowledge [3]. Other works have focused on studying BP alignment with LIS [4].

## III. MARBLE

The proposed ADM architecture focuses on the reverse engineering stage to obtain BPs from LIS. KDM is the core of this architecture, since it enables the representation and management of knowledge extracted from LIS in an integrated and standardized manner. Then, this knowledge will be gradually transformed into BPs. For this purpose, MARBLE proposes four abstraction levels with three transformations among them (see Figure 1).

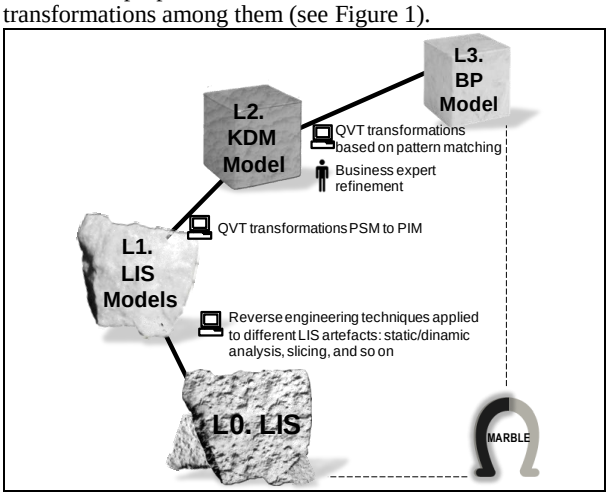


Figure 1. The overview of MARBLE for recovering Business Processes.

- Level L0 represents the LIS in the real world, which is the source system to extract their BPs.
- Level L1 represents several models for the different software artifacts of the LIS. These models are PSM models (Platform-Specific Models) since they are linked to the specific technological platform of each artifact.
- Level L2 consists of a single PIM model (Platform-Independent Model) that represents an integrated view of the set of models in L1. Due to this fact, the KDM metamodel is used, since it makes it possible to model all the artifacts of the LIS in an integrated and technological-independent manner.
- Level L3, finally, represents the BPs recovered from the LIS that is represented by a KDM model in L2. The BP model depicts the CIM model (Computation-Independent Model) of the system.
- Level L3. The last level represents the BPs recovered from the KDM model. The BP model is the CIM model and defines a set of BP diagrams.

### A. L0-to-L1 Ttransformation

The L0-to-L1 transformation obtains PSM models from each legacy software artifact. For this purpose, the classical reverse engineering techniques are used [1]: static and dynamic analysis, program slicing, and so on.

The PSM models are built according to specific metamodels. For instance, a Java metamodel for legacy source code, an SQL metamodel for modeling database schema, and so on.

### B. L1-to-L2 Ttransformation

The L1-to-L2 transformation consists of a set of model transformations to obtain a PIM model based on the KDM metamodel from the PSM models of level L1. These model transformations are implemented by means of QVT (*Query/View/Transformation*) [11]. The QVT specification consists of two different but related languages: (i) *QVT-Operational* as a language of procedural nature; and (ii) *QVT-Relations*, a declarative language. The latter language was used to establish the L1-to-L2 transformation in MARBLE.

The transformation from the LIS (L0) to the KDM model (L2) is not direct because in many cases, the platform-specific knowledge in the intermediate level L1 can be used to infer specific business knowledge.

### C. L2-to-L3 Ttransformation

KDM defines a common metamodel of knowledge related to software artifacts. Using KDM according to the ADM approach, the reverse engineering tools recover different knowledge related to different artifacts, but this knowledge is represented and managed in an integrated manner. Then, the analytic tools analyze the KDM model and generate new knowledge representing a particular point of view of the LIS.

The BP models in level L3, in turn, are represented according to the metamodel of BPMN (*Business Process Modeling Notation*) [10]. The BPMN metamodel represents BP diagrams that involve four kinds of elements: (i) flow object elements such as *events*, *activities* and *gateways*; (ii) connecting object elements like *sequence flows*, *message flows* and *associations*; (iii) artifact elements such as *data objects*, *groups* and *annotations*; and (iv) swimlane elements for grouping elements such as *pools* and *lanes*.

TABLE I. RULES TO OBTAIN BP ELEMENTS FROM LIS ELEMENTS

| Elements detected in L2 | Elements built in L3 |
|---|---|
| Package, compilation units or other aggregation units | Business process diagram |
| Callable units | Tasks |
| Calls between callable units | Sequence flows between tasks |
| Conditional branching (typically if-then-else or switch sentences) | Exclusive gateways that branch the sequence flow |
| Storable units read or written for callable unit | Data objects with associations to or from the task |
| Conditional calls | Sequence flows with an intermediate conditional event |
| The first callable unit called | Start event and sequence flow to the task |

The L2-to-L3 transformation is based on a set of patterns. Each pattern indicates what elements should be built and how they are interrelated in the BP model from L3, when a specific structure is detected in the KDM model of level L2. That is known as pattern matching and is implemented in MARBLE by means of QVT relations. TABLE I shows the most important rules defined by the set of patterns.

In addition, the human factor must be involved in this latter transformation, since the preliminary BP model obtained after pattern matching can be successively refined by business experts. For instance, the business expert might add the manual tasks of the organization, remove redundant sequences of tasks, rename some tasks, and so on.

## IV. A TOOL TO SUPPORT MARBLE

A tool based on the *Eclipse* platform was implemented in order to automate MARBLE to obtain BPs from LIS (see Figure 2). The tool consists of three modules that support the three proposed transformations. In order to support the first transformation, a tool module to carry out static analysis was developed. In this case, the module was built specifically for parsing Java source code. This tool module was developed through *JavaCC* (a parser/scanner generator for Java), from the EBNF (*Extended Backus–Naur Form*) grammar of Java 1.5. This module takes a Java file as input and then generates an XML (*Extensible Markup Language*) file as the output that represents the Java code model.

The second module executes the QVT transformation to obtain the KDM model from the Java code model obtained previously. The third module also executes the QVT transformation to support the third transformation of MARBLE. Both modules execute the QVT transformations by means of *MediniQVT*, an open source QVT engine. Moreover, a graphical editor of BP diagrams was developed using EMF/GMF (*Eclipse and Graphical Modeling Framework)* technology. Thus, the business expert can visualize and modify the obtained BP models.

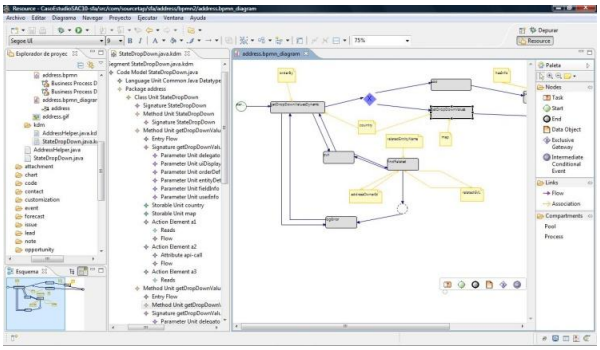

Figure 2. The tool to support MARBLE

## V. CASE STUDY

The case study applies the MARBLE process to *RapidMiner-Text* [14], a plug-in of an open source data mining application. *RapidMiner* (formerly *YALE*) is a worldwide leading open-source data mining solution. *RapidMiner* provides an easy-to-use extension and plug-in mechanism that makes it possible to integrate new operators. The Text *plug-in* supports the creation of word vector representations of text documents in the vector space model (each document is represented by the terms it contains). This is the point of departure for many text-processing applications such as web mining, text classification and information retrieval.

The source code of the system is written in Java. This system consists of 78 source code files divided into 11 packages and the size of the systems is 10.15 *KLOC*.

In order to analyze the obtained results, two research questions were designed to determine whether the BP models are obtained with the adequate quality level:

*Q1. Are the obtained BP models cohesive?*

*Q2. What is the degree of coupling of the BP models?*

This case study uses two metrics to measure the quality of the BP diagrams proposed by *Rolón et al.* [15]. The *Cohesion* metric indicates whether the elements of the BP diagram are stuck together. *Cohesion* (1) is equal to the number of sequence flows between tasks divided by the total number of tasks. The *Coupling* metric (2), in turn, represents the degree of dependence between elements of the diagram, thus this metric is equal to the number of output data objects divided by the total number of tasks of the BP diagram.

$$COHESION = \frac{Number\ of\ sequence\ flows\ between\ tasks}{Number\ of\ tasks} \quad (1)$$

$$COUPLING = \frac{Number\ of\ output\ data\ objects}{Number\ of\ tasks} \quad (2)$$

TABLE II. RESULTS OBTAINED IN THE CASE STUDY

| Package | N. of KDM models | N. of BPMN models | N. of Tasks | N. of Sequence Flows between Tasks | N. of Output Data Objects | COHESION | COUPLING | N. of Elements | Transf. time (ms) |
|---|---|---|---|---|---|---|---|---|---|
| operator | 15 | 1 | 72 | 80 | 9 | 1,11 | 0,13 | 270 | 2032 |
| crawler | 2 | 1 | 8 | 2 | 0 | 0,25 | 0,00 | 18 | 445 |
| extraction | 6 | 1 | 18 | 16 | 1 | 0,89 | 0,06 | 62 | 309 |
| segmenter | 3 | 1 | 3 | 0 | 2 | 0,00 | 0,67 | 20 | 446 |
| util | 2 | 1 | 9 | 6 | 5 | 0,67 | 0,56 | 52 | 271 |
| loganalysis | 4 | 1 | 19 | 20 | 2 | 1,05 | 0,11 | 96 | 418 |
| reducer | 4 | 1 | 5 | | | | | 13 | 155 |
| tokenizer | 2 | 1 | 9 | 6 | 1 | 0,67 | 0,11 | 43 | 158 |
| transformer | 1 | 1 | 4 | 4 | 2 | 1,00 | 0,50 | 31 | 165 |
| wordfilter | 2 | 1 | 3 | | | | | 9 | 81 |
| matrix | 6 | 1 | 23 | 12 | 1 | 0,52 | 0,04 | 63 | 224 |
| ***Total*** | 47 | 11 | 173 | 146 | 23 | 6,16 | 2,16 | 677 | 4704 |
| ***Mean*** | 4,27 | 1,00 | 15,73 | 16,22 | 2,56 | 0,68 | 0,24 | 61,55 | 427,64 |
| ***Std Deviation*** | 3,93 | 0,00 | 19,91 | 24,81 | 2,79 | 0,38 | 0,26 | 73,99 | 546,90 |

The execution of the case study is carried out by means of the developed tool. Then, the BP models obtained after the execution are inspected and the necessary measures are registered. TABLE II summarizes the obtained results: *package* represents the source code package which is transformed into a BP model; *N. of KDM models* and *N. of BPMN models* represent the number of those models obtained in this package; *N. of Tasks, N. of Sequence Flows between Tasks and N. of Output Data Object* are the base metrics necessary to measure (1) and (2); *COHESION* is the

derived metric (1); *COUPLING* is the derived metric (2); *N. of Elements* represents the total number of elements; and finally, *Transf. time (ms)* represents the total milliseconds spent to obtain the BPMN model by means of the tool.

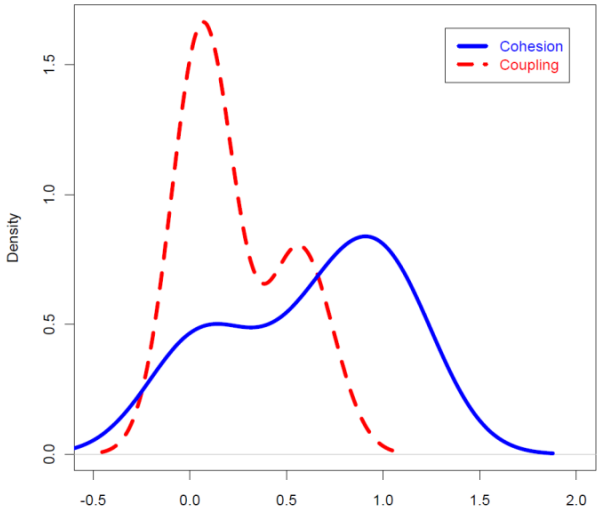


Figure 3. Density functions of the Cohesion and the Coupling.

In order to respond to questions *Q1* and *Q2*, Figure 3 shows the density charts of the *Coupling* (left side) and *Cohesion* (right side). Both density functions follow a normal distribution, approximately. *Cohesion* has a mean of 0.68 and a standard deviation of 0.38. The mean is closer to 1.0, thus this value indicates that the BPMN models obtained through the proposed patterns are extremely cohesive. Moreover, *Coupling* has a mean of 0.24 and a standard deviation of 0.26. That mean value is very low, and therefore the low coupling of the preliminary BPMN models can also be assured.

## VI. Conclusions

This paper presents MARBLE, an ADM approach to recover the underlying BPs in LIS. Thus the recovered business knowledge of the LIS is preserved and can be used to modernize and maintain the LIS, and preserve the alignment between the business logic and the IS.

MARBLE carries out reverse engineering processes in order to progressively build models on four abstraction levels until it obtains preliminary BP models. Therefore, three transformations are established between those levels: firstly, (1) the software artifacts of the LIS are transformed into PSM models by means of typical techniques of reverse engineering; secondly, (2) the PSM models are integrated into a KDM model (PIM model) through model transformations; and finally (3) preliminary BP models are obtained from the KDM model by means of model transformations based on pattern matching. Then, the concise BP models must be refined by a business expert in order to enrich those models.

Additionally, a tool and a case study make it possible to validate the quality levels of the BP models obtained through the proposal. The cohesion and coupling levels were adequate. In spite of this fact, the large LIS can result in large BP models, and thus the future extensions of this research will focus on introducing clustering techniques in order to reduce the size of the BP models.

## Acknowledgment

This work was supported by the *FPU Spanish Program*; by the R+D projects funded by *JCCM*: ALTAMIRA (PII2I09-0106-2463), INGENIO (PAC08-0154-9262) and PRALIN (PAC08-0121-1374); and MITOS (TC20091098) funded by the *UCLM*.

## References

[1] Canfora, G. and M. Di Penta. "New Frontiers of Reverse Engineering". in 2007 Future of Software Engineering. 2007: IEEE CS. p. 326-341.

[2] 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 CS. p. 139-148.

[3] Ghose, A., G. Koliadis, and A. Chueng. "Process Discovery from Model and Text Artefacts". in IEEE Congress on Services (Services'07). 2007 p. 167-174.

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

[5] ISO/IEC, ISO/IEC DIS 19506. Knowledge Discovery Meta-model (KDM), v1.1 (Architecture-Driven Modernization). http://www.iso.org/iso/catalogue_detail.htm?csnumber=32625. 2009, ISO/IEC. p. 302.

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

[7] Moyer, B. (2009) Software Archeology. Modernizing Old Systems. Embedded Technology Journal, http://adm.omg.org/docs/Software_Archeology_4-Mar-2009.pdf

[8] OMG, ADM Glossary of Definitions and Terms. http://adm.omg.org/ADM_Glossary_Spreadsheet_pdf.pdf. 2006, OMG. p. 34.

[9] OMG. ADM Task Force by OMG. 2007 [accessed on 06/15/2009]; Available from: http://www.omg.org/.

[10] OMG, Business Process Model and Notation (BPMN) 2.0. 2008, Object Management Group: Needham, MA 02494 USA. p. 34.

[11] OMG, QVT. Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification. http://www.omg.org/spec/QVT/1.0/PDF. 2008, OMG.

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

[13] Pérez-Castillo, R., I. García-Rodríguez de Guzmán, I. Caballero, M. Polo, and M. Piattini, PRECISO: A Reengineering Process and a Tool for Database Modernisation through Web Services in 24th Annual ACM Symposium on Applied Computing (SAC'09). 2009: Hawaii, USA. p. 2126-2133.

[14] RapidMiner, RapidMiner. Open Source Data Mining. http://rapid-i.com/content/blogcategory/38/69/lang,en/. 2009.

[15] Rolón, E., F. Ruiz, F. García, and M. Piattini. "Evaluation measures for business process models". in 21th ACM Symposium on Applied Computing, Track on Organizational Engineering (SAC-OE'06). 2006. Dijon, France: ACM p. 1567-1568.

[16] Zou, Y., T.C. Lau, K. Kontogiannis, T. Tong, and R. McKegney, Model-Driven Business Process Recovery, in Proceedings of the 11th Working Conference on Reverse Engineering (WCRE 2004). 2004, IEEE Computer Society. p. 224-233.