Joseph Barjis
Robert Pergl (Eds.)

LNBIP 191

Enterprise and Organizational Modeling and Simulation

10th International Workshop, EOMAS 2014
Held at CAiSE 2014, Thessaloniki, Greece, June 16–17, 2014
Selected Papers

Springer

Lecture Notes in Business Information Processing 191

More information about this series at http://www.springer.com/series/7911

Joseph Barjis · Robert Pergl (Eds.)

Enterprise and Organizational Modeling and Simulation

10th International Workshop, EOMAS 2014
Held at CAiSE 2014, Thessaloniki, Greece,
June 16–17, 2014
Selected Papers

Springer

Editors
Joseph Barjis
Delft University of Technology
Delft
The Netherlands

Robert Pergl
Czech Technical University in Prague
Prague
Czech Republic

ISSN 1865-1348 ISSN 1865-1356 (electronic)
ISBN 978-3-662-44859-5 ISBN 978-3-662-44860-1 (eBook)
DOI 10.1007/978-3-662-44860-1

Library of Congress Control Number: 2014950826

Springer Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Modern enterprises are growing in complexity in all dimensions from IT landscape to intricate business processes and workflows. Not only the internal factors, but also the external ones play a significant role in constant changes that an enterprise has to make. Efficiency of these enterprises is crucial in delivering service to the society and contributing to economic prosperity. For designing and redesigning an efficient and well-integrated enterprise, while qualitative methods play an important role, quantitative methods reinforce to make profound design decisions.

With this purpose, EOMAS was launched to become a forum among researchers and practitioners to share their research and practical findings. In this forum we encourage dissemination of research results under a more generic umbrella called enterprise engineering.

As any system, an enterprise is an object of continuous improvements, redesign, and reimplementation. The departure point for any design or redesign activity pertinent to an enterprise is first to understand the enterprise business processes. Therefore, in the overall enterprise engineering activities, business process modeling plays a central role. However, an extended enterprise and organizational study involves both analysis and design activities, in which not only modeling but also simulation plays a prominent role. Therefore this growing importance of modeling and simulation in the context of enterprises is attracting serious attention from researchers. Today, modeling and simulation are the tools and methods that are effective, efficient, economic, and widely used in enterprise engineering, organizational study, and business process management.

Complementary insights of modeling and simulation in enterprise engineering constitute a whole cycle of study of enterprises. In order to monitor and study business processes and interaction of actors in a realistic and interactive environment, simulation has proven to be a powerful tool and method, especially if simulation is supported with rich animation and gaming elements. In order to explore these topics, address the underlying challenges, find and improve solutions, and demonstrate applications of modeling and simulation in the domain of enterprise, its organization and underlying business processes, peer-refereed papers were accepted for presentation at EOMAS 2014, which was held on June 16–17, 2014, in Thessaloniki, Greece. This year, we had a total of 22 paper submissions, of which 12 were accepted for publication.

June 2014

Joseph Barjis
Robert Pergl

Organization

The EOMAS workshop is annually organized as an international forum for researchers and practitioners in the field of Enterprise & Organization Modeling and Simulation. Organization of this workshop, planning, and review of the contributions were accomplished by an international team of researchers.

Workshop Organizers

General Chair

Joseph Barjis	Delft University of Technology, The Netherlands

Program Chair

Robert Pergl	Czech Technical University in Prague, Czech Republic

Program Committee

Antonia Albani	University of St. Gallen, Switzerland
Anteneh Ayanso	Brock University, Canada
Joseph Barjis	Delft University of Technology, The Netherlands
Ygal Bendavid	University of Quebec in Montreal (UQAM), Canada
Kawtar Benghazi	University of Granada, Spain
Mahmoud Boufaida	Mentouri University of Constantine, Algeria
Tatiana Bouzdine-Chameeva	BEM, Bordeaux Management School, France
Manuel I. Capel-Tuñón	University of Granada, Spain
Davood Chitchian	XomicX Company, The Netherlands
José Luis Garrido Bullejos	University of Granada, Spain
Rafael Gonzalez	Javeriana University, Colombia
P. Radha Krishna	Infosys Technologies Ltd., India
Peggy Daniels Lee	Indiana University-Purdue University Indianapolis, USA
Prabhat Mahanti	University of New Brunswick, Canada
Vojtech Merunka	Czech University of Life Sciences Prague, Czech Republic
Alta van der Merwe	University of South Africa, South Africa
Martin Molhanec	Czech Technical University in Prague, Czech Republic

Sponsoring Institutions

– SIGMAS (**S**pecial **I**nterest **G**roup on **M**odeling And **S**imulation of the Association for Information Systems)
– CAiSE 2014 (International **C**onference on **A**dvanced **I**nformation **S**ystems Engineering)
– Czech Technical University in Prague (Faculty of Information Technology, Department of Software Engineering)

Contents

Enterprise Conceptual Modelling and Simulation

Extraction and Reconstruction of Enterprise Models

Mario Sánchez(✉), Julio Cesar Reyes, and Jorge Villalobos

Systems and Computing Engineering Department, School of Engineering,
Universidad de Los Andes, Bogotá, Colombia
{mar-san1,jc.reyes159,jvillalo}@uniandes.edu.co

Abstract. Enterprise Models for analysis, and especially for automated analysis, should have five characteristics: they have to be accurate representations of the reality; they have to be well structured; they have to be complete with respect to their intended usage; they have to be kept up-to-date; and the cost of their construction and maintenance has to be as low as possible. In this paper we present an approach for the semi-automatic construction of enterprise models which gathers and weaves information from multiple sources such as information systems, databases, files (system's logs, source code, configuration), and previously existing models. This approach is based on modeling and metamodeling techniques, and has been implemented in a tool called EM-AutoBuilder.

Keywords: Enterprise modeling · MDE · Automatic documentation · Model analysis

1 Introduction

Enterprise Modeling (EM), the discipline and practice of building and analyzing models representing one or many concerns of an enterprise, is progressively becoming mature and widespread. The value that can be gained from doing EM is directly proportional both to the quality of the models, and the quality of the available tools and methods to perform the analyses: if models are small, have low level of detail, have low fidelity, or are unstructured (e.g., text documents), it is difficult to perform insightful analyses; on the other hand, if analyses are simple and naive, there is no point in building advanced and detailed models. A clear example of this are simulation-based analyses, which are very advanced but require high-quality models with information spanning several domains. In this paper we focus on the first concern (model quality) and make a proposal to address this problem and thus increment the value that can be gained from EM, and especially from model analysis.

The biggest issue affecting the quality of enterprise models is the elevated costs of construction and maintenance. Since building these models is typically a human-intensive task, compromises are made which go against quality. For example, the scope of the model may be limited, or its depth, or its completeness. Furthermore, the lack of widespread modeling tools usually results in the

J. Barjis and R. Pergl (Eds.): EOMAS 2014, LNBIP 191, pp. 3–20, 2014.
DOI: 10.1007/978-3-662-44860-1_1

usage of inadequate technologies (e.g., text processors, spreadsheets, unstructured diagrams) that produce models that are impossible to process. Finally, enterprises are ever changing, and thus enterprise models must be permanently maintained. However, the current situation makes it very expensive to make these permanent upgrades.

The hypothesis that we attempt to validate with this work is that a lot of the information that should be included in an enterprise model can be obtained and structured in a *largely automated* way. For example, it should be possible to gather information about the architecture of deployed information systems, combine it with information about the enterprise coming from structured documentation, and finally enrich it with real statistics about its behavior coming from systems' log registries. The result of this, would be a comprehensive enterprise model which can be easily kept updated. Automating steps in the process of collecting and structuring the models, should also remove potential sources of errors and inconsistencies. Ultimately, this all should lead to increasing the quality of enterprise models and the value that can be obtained from them.

To validate this hypothesis, we designed an approach and architecture for building enterprise models using information obtained from different sources. This approach was implemented in a tool called EM-AutoBuilder, which has already been tested in an internal case study, and is now starting to be applied in real scenarios.

The structure of the paper is as follows. Section 2 discusses in more detail what Enterprise Models are, the possible sources of information to build them, and the current state of art in automatic construction of enterprise models. Section 3 introduces a scenario to illustrate the solution. Then, the proposed approach and its implementation are presented and illustrated in Sect. 4. Section 5 concludes the paper and discusses the outlook for the presented work.

2 Automatic Construction of Enterprise Models

An Enterprise Model is a representation of elements of an enterprise that typically belong to different domains (e.g., business processes, business and regulatory environment, organizational structure, or information technology). The cost of building enterprise models varies depending on two factors: the required level of detail, and the scope that the model should cover (i.e., how much of the enterprise and how many domains should be represented in the model). Before making a commitment to build a model, these two variables should be analyzed and balanced against the cost of construction, and against the benefits that the model can eventually bring. To further complicate the matter, these benefits are not intrinsic to the model, but depend on how it is used. For instance, it can be used for (i) documentation or communication purposes; (ii) as a way to increase understanding of the enterprise; (iii) as the starting point for analyzing the current situation of the company; (iv) or to evaluate transformation projects.

Given the aforementioned potential uses of enterprise models, there are a number of desirable qualities that said models should posses. The first and most important one is *accuracy*: a model that does not reflect the reality cannot answer truthfully any kind of answer, and thus is useless. On top of that, making decisions based on incorrect information can be worse than not making the decisions at all. A second, related quality is that a model has to be *up to date* (old information is just a particular kind of incorrect information, and just as risky as inaccurate information). Next, an enterprise model should to be *structured*: while models are frequently represented using unstructured means such as documents, diagrams, and spreadsheets, their real value can only be achieved if they are built around well defined structures and using representations that favor automatic processing. The fourth quality is *completeness*: a model should be complete with respect to its intended usage, and it should not lack information that it is expected to have. Finally, the *cost of building and maintaining* an enterprise model should be as low as possible; otherwise, it will probably be incomplete or will quickly cease to be up to date.

To build an enterprise model, it is critical to discover the potential sources of information. The typical sources include personnel of the company that deliver the information by means of interviews; manuals and documentations about processes, procedures, organizational structures, and responsibilities; and technical documentation about applications and technological elements. Other powerful sources of information are the Information Systems (IS) themselves, which are usually capable of providing structural and behavioral information. This can be achieved by direct observation of the IS technological components (interfaces, configuration files, source code, etc.), or by studying the relevant documentation and architectural documents. Furthermore, observing the storage systems and logging records of those IS provides valuable information to build models that are also behavioral, instead of purely structural.

Automatically building enterprise models is ideal, considering the discussed qualities. However, most sources of information are not suitable for this: only very well structured documentation (e.g., based in a Quality Management System Software), and the information systems themselves, can be automatically studied in order to build accurate, up to date, structured, and complete models, at relatively low costs.

In the past, some projects have made attempts at automatically building models with different levels of detail and focusing on particular domains. The work of Buschle et al. [1] reconstructs models of enterprise architecture based on network scanning and retrieving, in a graph-based structure, the main components of the application infrastructure of information systems. In this work they collected information using network analysis tools and vulnerability examination tools, and the results were application deployment models. Similarly, the work of Binz et al. [2] combined a manual and a semi-automatic approach for model construction. The automatic part was limited to network assets discovery (e.g., operative systems, DBMS, Application Servers), and the result was a graph representing the topology of enterprise systems.

Other works have tackled the problem from a source code perspective. Instead of obtaining the information from the systems already deployed, they have collected the information from their source code. These works include MoDisco [3], the work of Schmerl et al. [4], and the work of Song et al. [5]. MoDisco is an extensible framework that is targeted to support software modernization. Moreover, as part of this, MoDisco is capable of analyzing artifacts such as source code, database structures, and configuration files, in order to create models representing existing systems. The work that we present in this paper borrows some architectural ideas from MoDisco, especially with respect to the extensibility capabilities.

On the other hand, the work of Schmerl et al. combines the analysis of source code with the analysis of low-level system events to obtain a representation that relates events that happen in specific use cases of the system, in a specific runtime scenario. Finally, the work of Song et al. [5] analyzes application API and calls, and uses this information to reconstruct the structure of the systems.

Our approach differs from the above mainly because they focus on a single domain, whereas we intend to create multi-dimensional enterprise models. We later show that our enterprise models are produced by combining static information obtained from documents, structural information coming from the observation of deployed information systems and/or their source code, and from the capture of behavioral information that is usually stored in such information system's logfiles.

3 An Illustrative Case Study

To illustrate the work presented in this paper, a case study is now introduced. This case study is a fictitious but realistic company called BPO Los Alpes[1], which offers outsourced services. In particular, we are going to focus on services for creating and managing donation campaigns. These include managing information, publicity, and press releases about the campaigns; gathering information about potential donors; contacting the donors; collecting payments; and tracking the success or failure of the campaigns.

The BPO bases its operation on the following internal information systems, which are illustrated in Fig. 1.

- **Donations System:** this is the system that handles donation campaigns and it is responsible both for managing the business logic as for storing the relevant information. This includes information about donation campaigns, potential and actual donors, donations, payments, and certificates. Information is stored in a relational database (DB DON).
- **CCA:** this is the system used by the call-center of the BPO to contact the potential donors and register information about the outcomes of the calls.

[1] We call the company fictitious because it does not really exist or offer any service. However, it is realistic because it was modeled after real companies that provide similar services, and because its information systems are completely build and operational. We use this, and other similar scenarios, to support research and initial prototypes, and also for educational purposes in the courses we teach.

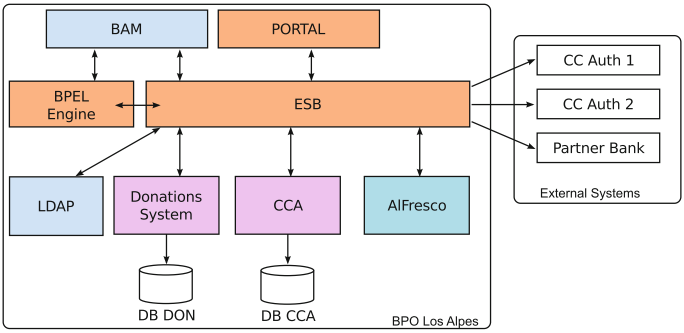

Fig. 1. Architecture of the case study

This system handles the contact-center agents, the lists of pending calls and their assignment, and an annotated registry of calls. Information is stored in a relational database (DB CCA).

- **AlFresco (ECM):** this is the enterprise content management system used to store the certificates for donors (which have legal validity for tax-exemption purposes), and to publish information about donation campaigns. The storage of its information is entirely handled by AlFresco.
- **LDAP:** this is the system that stores information about the users of the different applications and is capable of granting or denying usage privileges. All the applications use this LDAP system for authentication purposes.

In addition to these internal systems, the BPO depends on consuming services from payment transaction systems provided by allied banks and by credit card authorization systems. There are also three elements used for integration and coordination purposes: these are a BPEL engine, an Enterprise Service Bus (ESB), and a Portal. Finally, there is a BAM system to calculate and display indicators based on information obtained from the different applications, the bus and the BPEL engine.

In addition to this, there are several documents and artifacts describing the structure and operation of the BPO. Some of them are unstructured documents created with a common text editor, and thus are very difficult to process; others are diagrams and models created with specialized tools (i.e., ArchiMate and BPMN editors) and thus can be processed with relative ease.

4 Automatic Documentation: EM-AutoBuilder

The goal of this section is to present an approach to build enterprise models with the qualities that were discussed in Sect. 2. This approach has been implemented in a tool called EM-AutoBuilder, has been validated using the scenario presented in Sect. 3, and is currently starting to be used in a real scenario.

The approach was designed to address four critical requirements. The first one required models to be constructed in a way as automatic as possible. Therefore, the approach is targeted to collecting information from sources that can be automatically processed, i.e. information systems and well-structured documents and models. The second requirement was supporting heterogeneity. That is, it should be possible to gather information from various sources that have different structures, support different purposes, and are built around different technologies. These sources may also include documents, further increasing the complexity and heterogeneity. The next requirement is also closely related: the approach should be extensible, in order to be applicable to new sources of information. The fourth and final requirement regards the output of the approach: it should be a single, integrated artifact that can be processed or loaded into other tools that provide analysis capabilities.

Given these requirements, we designed the approach that is illustrated in Fig. 2. The core of this approach is a component that hosts several *configurable extractors* and processes their outputs. Extractors are independent components, and each one is capable of connecting to some specific kind of source to extract information. This information is returned to the core structured as a model. After each extractor has provided one or several models, the core has to process them to build an integrated one. However, the models may conform to different metamodels, and the core cannot know those metamodels before hand. Therefore, each extractor also has the capacity to provide the metamodels that it uses to build the models.

A final point in the strategy is the capacity of the core to weave the models, based on the weaving of the metamodels. It has been shown that completely automating the latter procedure is not feasible [6]. Therefore, we decided to let this step under the responsibility of the user, which has to specify the necessary relations between the metamodels. Finally, models are woven based on the information that the user provided, and a single tuple <metamodel, model> is produced.

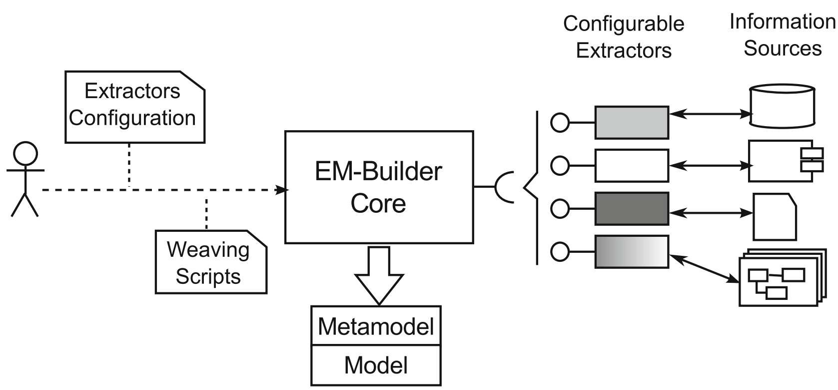

Fig. 2. Overview of EM-AutoBuilder architecture

The approach has been implemented in a tool called EM-AutoBuilder. This tool is based on Java and EMF[2] technologies because this makes it compatible with many tools for model processing, analysis, and visualization. Moreover, it should be possible to build similar tools using other technologies that match other tool environments. The following sections provide more details about each of the elements involved and about the responsibilities of each one.

4.1 Individual Extraction of Information

The first step in the automatic creation of enterprise models is to collect information from all the relevant data sources. However, these sources have a level of diversity that makes it impossible to have an universal component capable of querying them all. Even in the simplest cases, such as extracting information from relational databases, small technical differences in the way of managing the schema structure may prevent the same component for querying any RDMS.

In EM-AutoBuilder, the solution for this diversity problem was to build a framework and define an abstract component, the *extractor*, that is capable of querying systems to obtain and structure information. Concrete extractors are built using the framework, and they only share an API and some libraries to assemble and process models. The API that all concrete extractors implement has the following two main methods:

- `configure`: this method receives a set of Java properties with the information that the extractor needs to find its data source. For example, in the case of an extractor to collect information from a relational database that is accessed through JDBC, the properties include the name of the driver to use, the url to locate the DB, and the username and password to connect.
- `collectInformation`: this method uses the configuration information to obtain information and structure it in an EMF model. The output of this method is a tuple containing a model and a metamodel that are used subsequently by the EM-AutoBuild core.

It is worth noting that the metamodel that each extractor returns is sometimes calculated as part of the process of collecting information, i.e. it cannot be known a priori and it depends on the information obtained from the source. The reason for this has two parts. Firstly, the relation *instanceOf*, between *instances* and their *types* naturally exists within some domains. Secondly, both the types and the instances should appear in the same model, and the structure of the instances should conform to the restrictions imposed by their corresponding types. Unfortunately, this is not something typically supported by modeling frameworks, and thus it would require ad hoc solutions in each case. We experimented with some alternatives to manage these situations, but they required capabilities not supported by EMF, such as deep instantiation and potencies [7]. Ultimately, the chosen strategy was also adopted because it simplifies the weaving process.

[2] EMF - Eclipse Modeling Framework: http://www.eclipse.org/modeling/emf/.

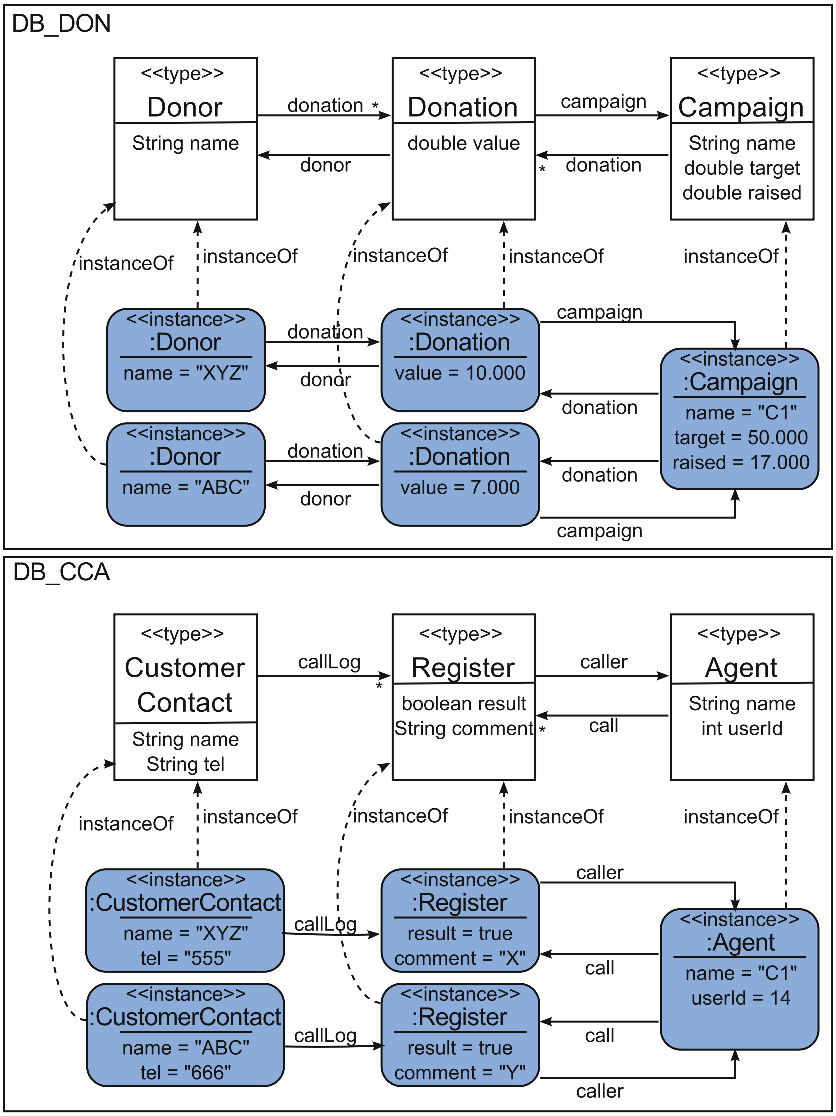

Fig. 3. Models obtained from the CCA and Donations System database

Figures 3 and 4 present a fragment of the results obtained from applying extractors to the BPO case study. In this case, the extractors utilized were capable of collecting information from relational databases, and they were applied to the Donations System database and to the CCA database. The results obtained were one model and one metamodel for each system.

Figure 3 presents the two models: on the top of the figure, there is the model obtained from analyzing the Donations System database; on the bottom of the