Chapter 10
A Model-Driven Solution for the Automatic Generation of Executable Code from Business Process Models

Javier Fabra
University of Zaragoza, Spain

Verónica Andrea Bollati
Rey Juan Carlos University, Spain

Valeria de Castro
Rey Juan Carlos University, Spain

Pedro Álvarez
University of Zaragoza, Spain

Esperanza Marcos
Rey Juan Carlos University, Spain

ABSTRACT

The business goals of an enterprise process are traced to business process models with the aim of being carried out during the execution stage. The automatic translation from these models to fully executable code that can be simulated and round-trip engineered is still an open challenge in the Business Process Management field. Model-driven Engineering has proposed a set of methodologies to solve the existing gap between business analysts and software developers, but the expected results have not been reached yet. In order to rise to this challenge, in this chapter the authors propose a solution based on the integration of three previous proposals: SOD-M, DENEB, and MeTAGeM. On the one hand, SOD-M is a model-driven method for the development of service-oriented systems. Business analysts can use SOD-M to transform their business goals into composition service models, a type of model that represents business processes. On the other hand, DENEB is a platform for the development and execution of flexible business processes, represented by means of workflow models. The authors' approach focuses on the automatic transformation of SOD-M models to DENEB workflow models, resulting in a business process that is coded by a class of high-level Petri-nets, and it is directly executable in DENEB. The model transformation process has been automated using the MeTAGeM tool, which automatically generates the set of ATL rules required to transform SOD-M models to DENEB workflows. Finally, the integration of the three proposals has been illustrated by means of a real system related to the management of medical images.

DOI: 10.4018/978-1-4666-6026-7.ch010

1. INTRODUCTION

In recent years, rapid development in the field of Web service technologies and Service-Oriented Computing (SOC) has created an interest in the area of Business Process Management (BPM) (Watson, 2008). BPM has broadened to become a set of technologies and standards for the design, execution, administration and monitoring of business processes, and also as a way of dealing with frequent changes in business and value chains (Watson, 2008; Havey, 2005). In conceptual terms, a business process is a defined set of activities that represents the steps required to achieve a business objective (OMG, 2011). In BPM terminology, an activity can be seen as the work of a person, an internal system, a service provided by an external entity or the process of a partner company.

The motivation in BPM and Service Oriented Architectures (SOA) areas has also led to the emergence of several languages for the design and implementation of such processes. The Web Services Business Process Execution Language, WS-BPEL (BPEL, 2011), is a standard language widely used for the specification of executable business processes. However, WS-BPEL and most of the current existing approaches for the specification of processes have some relevant problems: i) it is difficult for business analysts to use them in the early stages of the development process (Verner, 2004) (analysis and modeling stages); ii) they are dependent on a specific implementation technology (more specifically, on the Web service technology) and, finally, iii) the lack of formal semantics that permit process analysis. These problems increase the gap between business analysts and software developers, which leads to serious limitations in the BPM field.

The evolution of the Business Process Model and Notation standard, BPMN 2.0 (OMG, 2011), was published as a future means to reduce this gap. The informal formalization of the execution semantics for all BPMN elements and the independence of specific implementation technologies proposed by the specification will solve the first two problems associated with previous approaches. Nevertheless, this promising proposal lacks formal semantics and needs time to mature. The aforementioned gap is, therefore, still open.

In order to overcome these limitations, the Web Engineering field has proposed a set of methodologies that make it easier to develop service-oriented systems based on current technologies. Model-driven Development (MDD) is a development methodology that is principally characterized by the use of models as a product (Selic, 2003). MDD is a subset of Model-driven Engineering (MDE) (Schmidt, 2006) because, as the E in MDE suggests, MDE goes beyond pure development activities and encompasses the other model-based tasks of a complete software engineering process (Ameller, 2009). Model-driven Architecture (MDA) (Miller & Mukerji, 2003) is the particular MDD methodology defined by the Object Management Group (OMG) and therefore relies on the use of OMG standards. More specifically, MDA offers an open, vendor neutral approach with which to master technological changes and provides a conceptual structure for: (i) specifying a system independently of the platform that supports it; (ii) choosing a particular platform for the system; (iii) specifying a system including platforms details; and (iv) transforming the system specification into the corresponding code for a particular target platform (i.e., moving the system specification to the technological one). Therefore, the two most relevant features of MDD methodologies are: the provision of models with which to cover the different stages involved in the design of a system, and the definition of models which are explicitly separated from implementation technologies and execution platform details.

However, after several years of the popularization of MDA proposals and the appearance of many MDD approaches for software development, the usefulness of these approaches is questionable (Selic, 2003). The main controversy is related to the lack of support in the automatic generation of code