Chapter 2
Model-Driven Engineering of Service-Oriented Systems:
A Research Agenda

Dragan Gašević
Athabasca University, Canada

Marek Hatala
Simon Fraser University, Canada

ABSTRACT

Service-oriented architectures (SOA) are an essential platform to provide infrastructures that support widespread collaboration between organizations. These service-oriented systems are a new context for software developers, who must now be equipped with new development methods and technologies. This new context has specific requirements, such as better collaboration and communication between business users and software engineering across organizations and increased agility of the development and maintenance processes to better respond to newly emerged or changed requirements. In this paper, the authors present a research agenda that looks at the use of a novel software engineering discipline—model-driven engineering. By switching the focus from low-level technical details to high-level problem-specific details, model-driven engineering addresses challenges in the development of service-oriented systems. This paper particularly discusses the approach to the development of service-oriented systems based on business process modeling, which integrate business vocabularies and rules in different stages of the development lifecycle. Here, model-driven engineering can provide many promising solutions.

1. INTRODUCTION

Today many institutions in almost every sector strive to improve their mutual collaboration, increase data sharing, and decrease the response time in order to provide high-quality services to their customers. Service-oriented architectures (SOA) based on Web services are seen as an essential platform to fulfill this goal. A good progress in several areas of SOA has been made and several standards for web services cover the most important aspects of software services. However, developing applications for SOAs represents a major shift for the developers. The

DOI: 10.4018/978-1-4666-1583-0.ch002

long term vision for SOA-based systems lies in open environments where software services are not a-priori known to the system developers. On the one hand, the system development can fully focus on the business process design aspects and leave binding to concrete services for the later stages of the development. On the other hand, this approach needs to be supported by the mechanisms that allow for capturing design decisions at the business process levels to be enforced and utilized at the implementation level. In reverse, services to be selected and properly integrated into the compositions implementing the business process must be better described.

The enthusiasm for SOAs and service-oriented systems is growing with the increasing number of standards for these technologies. Many basic aspects of SOAs and Web services have been standardized enabling increase in implementation and deployment of SOA-based systems. However, as we demonstrate in the literature review section below, the standards are fragmented and a major effort by software developers is needed to bridge between them. The efficient development of high-quality systems based on SOAs requires extension of standards, modeling languages, and system development tools to facilitate system development process.

In this paper, we investigate some key research directions in addressing the above mentioned problems. Our primary focus is on investigating the promises of a novel software engineering area – Model-Driven Engineering (MDE) – for the development of service-oriented systems. MDE offers modeling methods that increase the level of abstraction when developing software, shifting from low-level platform specific details towards high-level software modeling approaches (i.e., computation-independent and platform-independent models). This not only allows us to reuse the same software design on various implementation platforms but more importantly to build more reliable service-oriented solutions which better responds to both stakeholder requirements and business rules and which is verified in early design stages. In Appendix A, we provide a brief introduction into the terminology of MDE.

Presently, development of SOAs and service-oriented systems mainly follows proven principles of business process modeling. Based on process-oriented modeling languages (e.g., Business Process Modeling–BPMN notation and UML Activity Diagrams), service developers start from the high-level business process models. Such models typically cover aspects such as flow of control, data, and activities. Once created, business process models are used as either blueprints for construction of service-oriented systems or even more ambitiously they are automatically translated into (skeletons of) implementation artifacts such as executable service compositions in Business Process Execution Language (OASIS BPEL, 2007). A significant progress in this area has been made in several critical aspects (see Sect. 2 for some details). In spite of the promising results, there are several research challenges to be addressed. In this paper, we outline some research challenges:

- *First*, process-oriented models are typically not well-connected with models of software structure or business vocabularies. This reduces the extent to which service definition can automatically be generated from the process models (e.g., definitions of message types).
- *Second*, constant changes of business logic require effective mechanisms for updating of process models. Currently, there is limited support for decoupling parts of business logic (e.g., constraints and process decisions) from the full process models. Consequently, this limits more dynamic updates of business logic and their propagation to the executable software systems.
- *Third*, the complete business process models are typically hard to be understood by business experts. Capturing smaller chunks