Chapter V

Model-Driven Service Engineering

Introduction

This chapter argues that modeling is at the core of every service engineering method. Modeling not only allows us to understand business services but, what is equally important, to transform them into software-realized services.

In general, models provide abstractions of a physical system that allow engineers to reason about that system while ignoring irrelevant details and focusing on relevant ones (Brown, 2004). All forms of engineering and science rely on models to understand complex, real-world systems. Models are used to predict system properties, reason about how changes in some parts of it will affect the rest of a system, and communicate key system characteristics to various stakeholders. The models may be developed as a mock-up or blueprint prior to implementing the physical system, or they may be derived from an existing system or a system in development, as an aid to understanding its behavior.

Models are therefore artifacts that represent real-world objects at some abstract level, giving the engineers the opportunity to modify, test, and preview certain proper-
ties of the actual object, in a cost- and effort-effective manner. Model construction also allows engineers to receive feedback from the stakeholders and validate their requirements early in the process.

In the electronics industry, models of circuits are utilized by computer-controlled machinery in order to produce the physical hardware circuits without human intervention. This idea of automating production of hardware has influenced recent thinking about how software should be developed. This premise of deriving software programs directly from software models, in the same way that hardware is manufactured from designs, is the essence of *model-driven development* (MDD) that is discussed in this chapter. MDD promises increased productivity by speeding up delivery times. To deliver such benefit, MDD is based on the concept of highly automated software environments for model management (Atkinson & Kühne, 2003).

Since the objective of service engineering is to realize business services as software programs (e.g., Web services), the principles of MDD apply to service engineering, too. Thus, the chapter examines MDD in the context of concrete service realization from abstract service models. The infrastructure for MDD is currently growing constantly with the introduction and evolution of standards; however, there are still a number of issues to resolve. As a result, although MDD is recognized to be a promising proposal for the future, it currently lacks general acceptance and applicability in the software industry.

Nevertheless, as we argue in this chapter, the future of service engineering is inextricably linked with progress in model-driven software development.

### The Use of Modeling in Software Development

Today, the majority of software developers still produce exclusively code; that is, they do not produce any other software artifacts at all. Instead, they rely almost entirely on the code they write, and they express their model of the system they are building directly in a programming language such as Java, C++, or C# within an integrated development environment (IDE), such as IBM Web Sphere Studio, Eclipse, or Microsoft Visual Studio. Of course, code produced in this manner implicitly contains more abstract models that, however, are hidden inside programming language constructs such as modules, packages, libraries, and so forth.

Any separate modeling of software designs is informal and intuitive, and only exists on paper drawings, or in the programmers’ heads. This approach, while adequate for small projects, fails to scale up to larger ones where properties of the system under development, such as its internal construction (architecture), needs to be communicated to a larger group of stakeholders, some of whom will be responsible to continue to maintain the system even after the original development team has gone.
Profightstore.com: Developing an Online Store for the Niche Market
Mirjana Pejic-Bach and Miran Pejic-Bach (2009). *Cases on Managing E-Services (pp. 76-88).*
[www.igi-global.com/chapter/profightstore-com-developing-online-store/6415?camid=4v1a](www.igi-global.com/chapter/profightstore-com-developing-online-store/6415?camid=4v1a)