Chapter XII

Generic Model of the Business Model and Its Formalization in Object-Z

Marcela Daniele, Universidad Nacional de Río Cuarto, Argentina
Paola Martellotto, Universidad Nacional de Río Cuarto, Argentina
Gabriel Baum, Universidad Nacional de Río Cuarto, Argentina

Abstract

This chapter shows the generic model of the business model, represented graphically with a UML class diagram, product of the analysis of the artifacts that compose the business model and their relationships, according to rational unified process (RUP) (2000). Moreover, the chapter defines a set of rules that the model must verify. It has been demonstrated that graph modeling is useful to visualize, specify, build, and document the artifacts of a system, offering a common language, easy to understand and apply. However, it lacks a precise semantics, which causes problems of ambiguities that in turn generate incorrect or different interpretations. In order to improve upon this, the generic model is translated into a formal specification language. This translation, carried out in Object-Z, allows the expression of a specific business model in a particular domain without ambiguity, and it facilitates the analysis of the properties of the system, showing possible inconsistencies, ambiguities, or incompleteness.
Introduction

The purpose of the development of a software system is to solve problems through programs that present a variety of characteristics such as efficiency, robustness, data safety, dynamism, and portability among others. The design of such a system is only possible once the problem has been identified, followed by a proposal of possible solutions.

The unified software development process (Jacobson, Booch, & Rumbaugh, 1999; RUP, 2000) is a methodology that defines who is doing what, when, and how to build or improve a software product. This method uses UML (unified modeling language) (Booch, Rumbaugh, & Jacobson, 1999; UML 2.0, 2003) as a way to express the different models that are created in the different stages of development. Business modeling allows one to obtain an abstraction of the organization. This is a set of activities whose goal is to help one visualize and understand business processes. As applied to software systems or other systems, a business model acts as a blueprint that guides the construction of the system. In effect, the model becomes an operational description of the business that can illuminate value/cost tradeoffs, priorities, and risks. This level of understanding is frequently essential to helping system analysts, designers, and developers make informed decisions about the processes they are automating and the technologies most appropriate for implementing them.

There are basic reasons to model a business. The first is to re-engineer a business. This involves analyzing and fundamentally rethinking how the business operates and interacts with the outside world. For this highest-risk form of process and system design, business modeling is essential. Second is to improve a business process, to streamline how the business works, and/or to enhance its competitiveness. Third is to automate a business process. The goal here is to reduce the resource requirements associated with a process by enabling more of it to happen without human intervention. In this context, a model of the current business allows one to understand the environment in which a software system will function. Whether the plan is to re-engineer the business or to automate an existing process, business modeling is the first step toward defining a software system that will solve the precise business problem and is fundamental to capturing and to structuring the system requirements.

Business-process automation is also the focus of the business-modeling discipline in the rational unified process. The business modeling is proposed by RUP (2000) through the development of a group of artifacts by means of which the business is completely modeled.

Research on business modeling that refers specifically to the unified software development process use the definition of artifacts as presented by their authors. In this context, it is worth mentioning Eriksson and Penker’s (2000) definition of an extension of UML to describe business processes. In addition to this, they describe a set of business rules in object constraint language (OCL) as well as how to use business models in use cases. Ortín, García Molina, Moros, and Nicolás (2001) propose a systematic method for obtaining the requirements model from the business model. Salm (2003) uses the extensions to UML for the business modeling proposed by the Object Management Group.