Chapter IX

A Platform for Model-Driven Service Engineering

Introduction

Having discussed concepts and requirements for managed service delivery in the last chapter, we have now completely covered the whole service engineering life cycle, namely identification, modeling, design, realization, deployment, and management. More specifically, so far, we have dealt with the following issues:

- Service concepts and fundamentals (Chapter II).
- Formalized service semantics and service metamodels (Chapter V).
- Service derivation from business models described in the IDEF0/IDEF1X languages (Chapter VII).
- Service realization by model-driven transformations of business services to executable Web services (Chapters VI and VII).
- Environments and platforms for service execution and management (Chapter VIII).
This chapter introduces a service engineering platform (CLMS) that caters for all the phases of the service engineering life cycle. Although complete coverage of the service life cycle is important, currently, few software environments and platforms manage it, with the majority of such environments dealing with just service execution (delivery). It is nevertheless important to have a single environment/platform that handles the whole service life cycle with seamless transition between phases (CBDI Forum, 2005). To deal with these requirements, the CLMS platform adopts a model-driven approach (see Chapter VI). More specifically, the platform seamlessly links business service specification models to realizable services and manages the delivery of the latter in an environment that monitors services execution. Moreover, this approach ensures that alternative service deployment techniques (e.g., proactive as well as reactive service delivery systems) and service execution environments can be accommodated. This is of particular importance today, with service engineering still evolving and with new service paradigms and competing technologies appearing all the time.

The chapter is organized as follows. The next section discusses the rationale and motivation behind the conception of the CLMS platform. After that, we introduce the platform’s architecture in terms of major subsystems. Next, we present the service engineering method associated with the CLMS platform. This is essentially an approach derived from the methodology that was presented in Chapter VII, modified to accommodate the requirements and constraints of the available technologies for service realization. Finally, will illustrate the environment at run time (i.e., when services are deployed, executed, and managed). The CLMS platform will be used again in Chapter X to illustrate a step-by-step method for service realization in the accounts receivable/payable financial domain.

Platform Overview and Philosophy

The CLMS platform allows the modeling (specification), design, realization, deployment, and management of business services using software technologies that include, but are not restricted to, Web services. The CLMS platform adopts a model-driven philosophy for the realization of the service, with different metamodels guiding and automating activities (e.g., modeling, specification, code generation) and driving the generation of software artifacts. There are currently no systematic approaches for deriving enterprise service-oriented architectures. A top-down or bottom-up analysis of an enterprise’s data or processes will probably fail to reveal all important services. We argue that a systemic view of the enterprise provides an effective mechanism for identifying and modeling services and, more importantly, for linking and tracing services to business processes and activities.