IDEA GROUP PUBLISHING ITB10877

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

This chapter appears in the book, *Service-Oriented Software System Engineering: Challenges and Practices*, edited by Zoran Stojanovic and Ajantha Dahanayake. © 2005, Idea Group Inc.

Chapter VI

Service-Oriented Computing and the Model-Driven Architecture

Giacomo Piccinelli
University College London, UK

James Skene
University College London, UK

Abstract

Service-Oriented Computing (SOC) and the Model-Driven Architecture (MDA) are complementary systems development approaches with the mutual aim of reducing the cost of future systems integration. This chapter introduces the MDA concept and technologies to an SOC audience and employs these technologies to enhance support for SOC through the definition of a domain-specific modeling language for electronic services. The language is defined as an extension of the Unified Modeling Language (UML). Its semantics are defined using a domain model of electronic service systems based on concepts drawn from literature and experience with a range of commercial platforms for the deployment of electronic services.

Introduction

Service-Oriented Computing (SOC) and the Model-Driven Architecture (MDA) are both approaches to developing systems that anticipate the need for integration in heterogeneous computing environments.

SOC attempts to lessen the cost of future integration by making a recommendation about the *design* of systems: a service paradigm should be applied to software and business functions to support standardized communication and control technologies, such as Web services and workflow languages.

The MDA attempts to lessen the cost of future integration by making a recommendation about the *process* by which systems are developed: systems should first be developed as abstract models that do not contain technical details related to implementation, then transformed into platform-specific representations. Removing the need to disentangle business functionality from legacy platform decisions when redeploying all or part of a system on a new platform reduces the cost of integration when it is necessary to support a previously unexpected integration technology, such as a new middleware. As the MDA matures, the cost of redeployment may be reduced still further by the availability of reusable automated transformations to produce platform-specific models from platform-independent models.

The MDA is potentially a good complement to SOC. Service-oriented systems can be developed according to the MDA process in order to structure a system according to a services paradigm while maintaining a platform-independent specification. The first objective of this chapter is to introduce the MDA concept and its supporting technologies to an SOC audience.

The second objective of this chapter is to discuss how MDA standards such as the Enterprise Distributed Object Computing (EDOC) profile support SOC. The EDOC profile defines standard extensions to the Unified Modeling Language (UML) to allow the platform-independent modeling of enterprise computing systems, a class of system that subsumes electronic services.

The third objective of this chapter is to show how MDA support for SOC can be fruitfully expanded. The MDA approach gains productivity advantages when supported by domain-specific languages, such as the EDOC profile, for modeling systems in particular application areas. The advantage of such languages is that they allow the modeling of systems in a manner that is more concise and less error prone than if attempting to model the same systems using a more generic language. The EDOC profile does not allow the explicit modeling of several system facets unique to electronic services. We therefore believe it beneficial to provide more refined support for modeling electronic services by defining a UML extension (a profile) specifically for this purpose.

Our profile supports three modeling tasks particular to the development of electronic service systems:

- First, it allows the modeling of services at an abstract level, using the service vocabulary of capabilities, content, provisioning, offers, and information exchange. This allows the planning and documentation of the intended behavior for both single and coordinated services, essential when applying a service paradigm to systems development and a precursor to implementation efforts.
- Second, the profile allows the modeling of service deployments in terms of the concrete business assets that fulfil capability roles. This provides a concrete view