Chapter XV

A Service-Oriented Component Modeling Approach

Zoran Stojanovic, Delft University of Technology, The Netherlands

Ajantha Dahanayake, Delft University of Technology, The Netherlands

Henk Sol, Delft University of Technology, The Netherlands

ABSTRACT

Although implementation technology and standards for Component-Based Development (CBD) and Web services are nowadays widely used in enterprise system development, there is a strong need for truly component-oriented modeling methods. CBD methods proposed so far do not provide a necessary support for modeling various component and service concepts throughout a development life cycle. They mainly follow a bottom-up approach by treating components as implementation level artifacts for packaging software code. However, the component can be much more useful if it is treated as a building block of the logical system architecture. This chapter presents a service-oriented component modeling approach focused on the concepts of component and service as the main modeling and design artifacts. The approach provides a paradigm shift from components as objects to components as service managers. The approach is business-driven, flexible, and agile, providing an effective business/IT alignment in a component- and service-oriented manner.

INTRODUCTION

For most enterprises, creating an information system that can accommodate the changes that occur in business, on the Internet, and in information technology (IT) is the most important task they face. Today, it is important more than ever to provide a seamless integration between business and IT in order to achieve the high flexibility of the solution within short time-to-market. Recent IT developments, Component-Based Development (CBD), and Web services have been introduced as new effective ways of building complex enterprise systems and providing enterprise application integration (Brown & Wallnau, 1998; Szyperski, 1998). The CBD platforms and technologies, such as CORBA Components (Siegel, 2000), Sun's Enterprise Java Beans, and Microsoft's COM+/.NET, are now *de facto* standards in enterprise-scale system development. On the other hand, the growing interest in Web services has resulted in a number of industry standards (WSDL, UDDI, SOAP, etc.) and initiatives (IBM, 2003; W3C, 2003). Although technology is very important in building complex systems nowadays, there is a strong need for component- and service-oriented development processes, modeling methods, and tools that would maximize the benefits of this new way of thinking. Current achievements in this respect are much behind the technology ones.

The CBD approaches proposed so far do not provide a full support for various component concepts (Dahanayake, Sol, & Stojanovic, 2003). They mainly treat components as binary packages of code influenced by the standard Unified Modeling Language (UML) (OMG, 2002). This suggests handling components at the implementation and deployment phases of a development life cycle, while still following classical object-oriented modeling, analysis, and design. In this way, the component thinking is limited to the level of physical things deployed over the network nodes that package one or more object implementations (Booch, 1999). More advanced component-based approaches use components during the system analysis and design, but represent them often as coarser-grained business objects. Following this strategy, these approaches are not well equipped and cannot be straightforwardly applied in building service-oriented system architecture, where different aspects of collaboration and coordination of services and components must be the focus of a development process. Finally, proposed methods are often very complex and heavyweight; that makes them difficult and time consuming for understanding and application. In the current fast-changing business and IT, more flexible, adaptable, agile, and ease-of-use methods and approaches are needed to shorten time-to-market and produce high-quality, easily adaptable solutions, following the flexible and highly interactive process. In our opinion, an appropriate component- and service-oriented architectural modeling approach is crucial for more effective business/ IT alignment. It provides the way of capturing given business requirements in the platform-independent system architecture. This architecture can be further mapped to the particular technology, providing, in this way, a bidirectional link

21 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/service-oriented-component-modeling-approach/23020

Related Content

Investigating the Effects of Object-Relational Impedance Mismatch on the Efficiency of Object-Relational Mapping Frameworks

Derek Colley, Clare Stanierand Md Asaduzzaman (2020). *Journal of Database Management (pp. 1-23).*

www.irma-international.org/article/investigating-the-effects-of-object-relational-impedance-mismatch-on-the-efficiency-of-object-relational-mapping-frameworks/266402

Toward a Formal Semantics for Control-Flow Process Models

Henry H. Biand John Nolt (2012). *Journal of Database Management (pp. 72-97).*

www.irma-international.org/article/toward-formal-semantics-control-flow/65542

Exploring the Effects of Process Characteristics on Product Quality in Open Source Software Development

Stefan Kochand Christian Neumann (2010). *Principle Advancements in Database Management Technologies: New Applications and Frameworks (pp. 132-159).*

www.irma-international.org/chapter/exploring-effects-process-characteristics-product/39353

How NoSQL Databases Work

(2018). *Bridging Relational and NoSQL Databases (pp. 124-175).*

www.irma-international.org/chapter/how-nosql-databases-work/191982

The Knowledge Transfer Process: From Field Studies to Technology Development

M. Millie Kwanand Pak-Keung Cheung (2006). *Journal of Database Management (pp. 16-32).*

www.irma-international.org/article/knowledge-transfer-process/3345