ABSTRACT

Components-Based Development (CBD) and Web Services (WS) nowadays are prominent paradigms for implementing and deploying advanced distributed information systems. They have been proposed as the ways to support effective business/IT alignment and produce high quality and flexible software solutions that fulfill business goals within short time-to-market. However, current achievements in these areas at the level of methodology are much behind the technology ones. CBD methods proposed so far lack a comprehensive support for component and service concepts throughout the development process. By treating components as packages of implementation artifacts during software deployment or as larger-grained business objects during analysis and design, these methods are not well equipped for modeling loosely coupled coarse-grained components that offer business meaningful services organized in a Service-Oriented Architecture (SOA). This chapter presents an evaluation framework that highlights the extent to which a particular method is component-based and service-oriented.
service-oriented. The CBD method sample is selected and evaluated using the framework’s concepts and requirements. Based on the evaluation, the method improvements are proposed in order to provide consistent, systematic, and integrated CBD and WS methodology support throughout the lifecycle.

INTRODUCTION

Modern enterprises are in the flux of rapid and often unpredictable changes in both business and Information Technology (IT). New business demands caused by the enterprise’s need to be competitive on the market require an immediate support of the advanced IT solutions. At the same time, new IT opportunities and achievements are constantly emerging and must be rapidly adopted to provide new and more effective ways of conducting business. Therefore, today more than ever it is important to provide an effective business/IT alignment in order to produce high quality and flexible software solutions within short time-to-market, that as close as possible support business goals and match business needs.

During the last years, new development paradigms and models have been proposed to support these aims. First Component-Based Development (CBD) (Brown & Wallnau, 1998), and then Web Services (WS) and Service-Oriented Architecture (SOA) (IBM, 2003; W3C, 2003) have been introduced as the ways to build complex enterprise systems and provide effective enterprise application integration. The CBD platforms and technologies, such as CORBA Components, Sun’s Enterprise Java Beans (EJB), and Microsoft’s COM+/+.NET are now de facto standards in web-based systems development. On the other hand, the growing interest in Web Services has resulted in a number of industry standards and initiatives (XML, WSDL, UDDI, SOAP, etc.) (W3C, 2003). What they have in common is that CBD and WS have both been first introduced through new technology standards and infrastructures, and after that corresponding methods, tools and modeling techniques have been proposed. While the technology is a necessary element of any solution, it is not sufficient on its own. Methods, techniques and tools for developing component-oriented applications based on business requirements are equally important (Welke, 1994). Such development methods need to incorporate the concepts of component and service as an integral part of the whole system life cycle, from business to implementation.

While there is an established development methodology practice in the case of CBD, in the field of WS and SOA, current achievements in this respect are much behind the technology ones. The former question of how to make use of object-oriented methods and techniques in practicing CBD is now largely replaced by whether and in what ways CBD methods can be used in developing WS applications. Therefore, of great importance is proposing an approach for architecting the system that consists of collaborating components and services. Such an approach should specify the way of capturing and organizing business requirements within the platform-independent logical system architecture that closely maps business concepts and goals. The approach should further provide mapping of the architecture to the particular technology settings that ensures bi-directional traceability between business concepts and implementation artifacts. This is the main idea behind the current Object Management Group’s (OMG) Model Driven Architecture (MDA) (OMG, 2003).

Current object-oriented and component-based development methods do not provide a necessary support for designing and developing component-based and service-oriented business applications. Methods that have evolved from pure object-oriented backgrounds
Out of Control? The Real ID Act of 2005
www.igi-global.com/chapter/out-control-real-act-2005/7989?camid=4v1a