Chapter IX

Architecture, Specification, and Design of Service-Oriented Systems

Jaroslav Král
Charles University, Czech Republic

Michal Žemlička
Charles University, Czech Republic

Abstract

Service-oriented software systems (SOSS) are becoming the leading paradigm of software engineering. The crucial elements of the requirements specification of SOSSs are discussed as well as the relation between the requirements specification and the architecture of SOSS. It is preferable to understand service orientation not to be limited to Web services and Internet only. It is shown that there are several variants of SOSS having different application domains, different user properties, different development processes, and different software engineering properties. The conditions implying advantageous user properties of SOSS are presented. The conditions are user-oriented interfaces of services, the application of peer-to-peer philosophy, and the combination of different technologies of communication between services (seemingly the obsolete
ones inclusive), and autonomy of the services. These conditions imply excellent software engineering properties of SOSSs as well. Service orientation promises to open the way to the software as a proper engineering product.

Introduction

Service orientation (SO) is becoming the central topic of software engineering. There is an explosive growth in the number of conferences, products, and articles discussing and using the principles of SO and service-oriented architectures (SOA). Service-oriented software systems (SOSS) are of different types depending on the character of the functions the system provides, the system environment (for example, e-commerce or a decentralized international enterprise), and the way the system is developed. The common property of SOSS is that their components behave like the services in real life mass service systems. The SOSS must then be virtual peer-to-peer (p2p) networks of autonomous components (services). The services can have various properties; they need not be Web services in the sense of W3C (2002) and need not therefore use standard communication protocols, compare Barry and Associates (2003) and Datz (2004).

We shall show that the software engineering properties as well as the user-oriented properties of any SOSS strongly depend on the properties of the service interfaces and that user interfaces of the system should be implemented as specific services (peers of the network) as well. All these issues are related to the architecture of the system. We will discuss how the properties of the architecture influence the set of feasible functions, development (especially the requirements specifications), feasible development techniques (for example, agile ones), standards, politics of IT management, and marketing strategies of software vendors and/or system integrators (Figure 1). The feasible functions of SOSSs include the functions important for user top-management.

Figure 1. Central role of system architecture

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.