Chapter X
Automatic Creation of GUI’s for Web–Based ERP Systems

Jorge Marx Gómez
Universität Oldenburg, Germany

Daniel Lübke
Leibniz Universität Hannover, Germany

ABSTRACT

Service-oriented architecture (SOA) is an emerging architectural style for developing and structuring business applications, especially enterprise resource planning (ERP) systems. SOA applications are composed of small, independent, and network-accessible software components, named services. The service composition is normally based on the enterprise’s business processes. However, current composition standards like BPEL have no ability to interact with users. Therefore, we propose a mechanism for including user interaction descriptions into the composition and extending the composition platform for generating user interfaces. In our case study, a federated ERP (FERP) system, this mechanism has been implemented in a prototype based on yet another workflow language (YAWL) dynamically generating Web pages for accessing the ERP system. Because every aspect including the user interfaces can be manipulated through the service composition, such systems are highly flexible yet maintainable.

INTRODUCTION

Enterprise resource planning (ERP) systems have become the most critical system in the IT infrastructure for most enterprises. However, most ERP software packages still follow a monolithic design. In recent years the software design of such systems tends to move to a new emerging architectural style—service-oriented architectures (SOA). Within an SOA, functionality is realized by small, fine-grained, independent, and network-accessible components called services. These services can be composed along the business processes of an enterprise. Since the compositions should be on a
very high abstraction level, the resulting system can be changed easily.

An SOA is defined in this chapter as an enterprise-wide distributed software architecture for business applications that consists of services as its elementary software components. Those services are composed according to given business processes, and linked to the processes at run-time. SOA's main design goals are flexibility and maintainability in regard to changes affecting these business processes.

SOA is currently most often realized by utilizing Web service standards. Web services are based on XML. For calling Web services the SOAP protocol (Gudgin, Hadley, et al., 2003) has been defined. Composition of Web services is defined in the business process execution language (BPEL) (Andrews, Curbera, et al., 2003). While BPEL aims to support the alignment of Web services with business processes, it lacks one critical component of business processes—interaction with and involvement of responsible users. Recently, BPEL4People (Kloppmann, Koenig, et al., 2005)—an extension for BPEL has been proposed. However, BPEL4People treats end-users merely as just other services. The developers and system integrators still have to develop matching user interfaces in traditional programming languages. Development of user interfaces requires much effort and runs contrary to the suggested flexibility of SOA.

This problem becomes even more dominant for smaller enterprises, which need to be very flexible in the market for maintaining their competitive edge but do not have many resources that they can devote to their IT infrastructure.

In this chapter we present a solution on how to make user interfaces for ERP systems more flexible by incorporating parts of user interface models into the service composition. Thereby, the user interface can be easily modeled and updated without having programming skills.

Within the course of this chapter, the differences between business processes and service compositions are explained first. Afterwards, our concept for annotating user interfaces is presented. In the following section the generation algorithm is presented. Then a case study, namely the federated ERP system, is presented, which heavily utilizes the generation facilities described before. Finally, a conclusion and an outlook are given.

**BUSINESS PROCESSES AND SERVICE COMPOSITIONS**

Business processes define how activities in an enterprise should be performed. Additionally, they define who is responsible for performing these activities and which resources are affected. Business processes are designed from the management point of view. Based upon their definition, activities within enterprises can be improved, and supporting IT systems can be designed. There are many notations available for modeling business processes, among them event-driven process chains (EPCs) (Mendling & Nüttgens, 2005) and the Business Process Modeling Notation (BPMN) (White, 2006). Taking an order as an example for an administrative business process is illustrated in EPC notation in Figure 1.

Service Compositions have a similar structure compared to business processes (Henkel, Zdravkovic, et al., 2004). However, they offer a technical view of the IT landscape of an enterprise. They define the control-flow between different services, related error-handling, and data management. While the control-flow should optimally match the one of the underlying business processes, technical constraints often require slight differences. For example, it is possible that certain activities cannot be performed in parallel due to constraints of the software. For maintaining a SOA it is therefore a critical task to synchronize changes of the business process with the service composition.

However, this is not the only difference between business processes and service compo-
Related Content

**Conceptual Model Driven Software Development (CMDSD) as a Catalyst Methodology for Building Sound Semantic Web Frameworks**
www.igi-global.com/chapter/conceptual-model-driven-software-development/5220?camid=4v1a

**Access Control and Information Flow Control for Web Services Security**
www.igi-global.com/article/access-control-and-information-flow-control-for-web-services-security/150001?camid=4v1a

**A Structured Methodology for Developing 3D Web Applications**
www.igi-global.com/chapter/structured-methodology-developing-web-applications/40491?camid=4v1a

**Experiences in Developing a Micro-payment System for Peer-to-Peer Networks**
www.igi-global.com/article/experiences-developing-micro-payment-system/41726?camid=4v1a