ABSTRACT

All software today is written using application programming interfaces (APIs). We performed a user study of the online documentation of a large and complex API for Enterprise Service-Oriented Architecture (eSOA), which identified many issues and recommendations for making API documentation easier to use. eSOA is an appropriate testbed because the target users include high-level business experts who do not have significant programming expertise and thus can be classified as “end-user developers.” Our study showed that the participants’ background influenced how they navigated the documentation. Lack of familiarity with business terminology was a barrier for developers without business application experience. Both groups avoided areas of the documentation that had an inconsistent visual design. A new design for the documentation that supports flexible navigation strategies seems to be required to support the wide range of users for eSOA. This paper summarizes our study and provides recommendations for future documentation for APIs.

Keywords: API Design, Business Solution Architects, Documentation, Natural Programming, Service-Oriented Architecture, Usability, Web Services

DOI: 10.4018/joeuc.2010101903
1 INTRODUCTION

“Service-Oriented Architecture” (SOA) is a way to structure large and distributed software systems, where services communicate over a network with the client and with other services, and can be combined into composite applications. Increasingly, companies are providing their operations, which may previously have required installing a custom application, as services on the Web, so they can be accessed from a browser. Enterprise SOA (eSOA) is focused specifically on supporting business processes across an enterprise by reusing existing services. When an eSOA application is being planned and developed, many kinds of people are involved. For example, business process experts, who might be titled “Solution Architects,” are knowledgeable about the business context but may not necessarily be professional programmers, and are often responsible for identifying and selecting which services will be used. These users might therefore be classified as “end-user developers” (Wulf, Paterno, & Lieberman, 2006). Specifications they write might then be passed to professional programmers, who are responsible for writing code that uses the actual services. Therefore, the documentation and some of the tools must be accessible to both business process experts and professional programmers.

In a service-oriented architecture, code on the user’s machine communicates with a remote service using messages across the Internet. When using Web services, the communication is usually encoded in XML, and the format of the messages might be described using a WSDL (Web Services Description Language) file, which has been formalized by the World Wide Web Consortium (e.g., http://www.w3.org/TR/wsdl).

As part of the “Natural Programming Project” (Myers, Pane, & Ko, 2004), we are interested in a whole range of usability issues around programming. Recently, we have begun to focus on the usability of libraries, frameworks, toolkits, and other application programming interfaces (APIs) (Ellis, Stylos, & Myers, 2007; Stylos, Faulring, Yang, & Myers, 2008; Stylos & Myers, 2008). APIs are crucial since most of modern development of all kinds involves finding, understanding, and connecting pre-built items, from small library calls to large-scale components. Enterprise SOA APIs are particularly interesting to study, because they can be large and complex, and therefore expose interesting issues of scale, and because they often target a wide range of developers.

Our previous research has shown that programming using eSOA APIs is not simple when the APIs are providing access to powerful business functionality (Beaton, Jeong, Xie, Stylos, & Myers, 2008; Beaton, Myers, Stylos, Jeong, & Xie, 2008). Some barriers we identified included long names for services, different behaviors of services due to different business behavior, pa-
Related Content

End User Computing Ergonomics: Facts of Fads?
www.irma-international.org/chapter/end-user-computing-ergonomics/18164/

Organizational and Implementation Issues of Patient Data Management Systems in an Intensive Care Unit
www.irma-international.org/article/organizational-implementation-issues-patient-data/3740/

The Effect of Social Software on Academic Libraries
www.irma-international.org/chapter/effect-social-software-academic-libraries/69759/

Rethinking Expertise in the Web 2.0 Era: Lessons Learned from Project Durian
www.irma-international.org/chapter/rethinking-expertise-web-era/69768/

Believe It or Not: Virtual Religion in the 21st Century
www.irma-international.org/chapter/believe-not-virtual-religion-21st/18297/