Semantics for E-Commerce Applications

Jorge Cardoso
University of Madeira, Portugal

INTRODUCTION

A few years ago, e-commerce applications were mainly focused on handling transactions and managing catalogs. Applications automated only a small portion of the electronic transaction process, for example: taking orders, scheduling shipments, and providing customer service. E-commerce was held back by closed markets that could not use distributed services, due to the use of incompatible communication protocols.

Recently, business needs are evolving beyond transaction support to include requirements for the interoperability and integration of heterogeneous, autonomous, and distributed service. Enabling technologies and business-centered design methodologies have addressed the shortcomings of contemporary e-commerce applications. New technological development such as Web services, Web processes, and semantics have allowed the creation of a new bread of e-commerce applications which can orchestrate cross-organizational and distributed services.

Web services and processes refer to a set of technologies that can universally standardize the communication of applications in order to connect systems, services, business partners, and customers cost-effectively through the World Wide Web. Semantics provide an agreed understanding of information between and among Web services encouraging the development of interoperable systems that can help create and support new collections of services to better meet the demands and expectations of customers.

In this article, we present seven reasons why semantics should be an integral part of Web services and Web processes technology managing e-commerce applications.

BACKGROUND

As organizations are increasingly faced with the challenge of managing e-commerce applications, important technological development such as Web services, Web processes, and semantics are emerging.

The main idea of Web services is to encapsulate an organization's functionality or service within an appropriate interface and advertise it in the Web using the Web service definition language (WSDL) (Christensen, Curbera, Meredith, & Weerawarana, 2001). Web services are a very general model for building distributed applications which can be used to link together computer programs from different suppliers and technologies. The principles behind Web services are very simple:

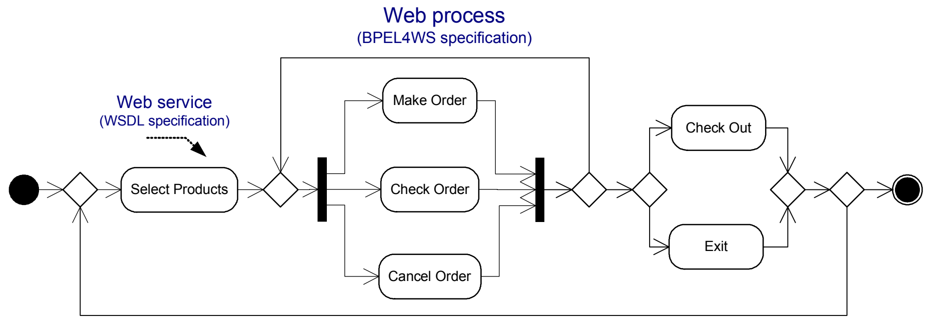
- A provider defines a standardized format for requests and responses for its Web services.
- A computer makes a request for the Web services across the network.
- The Web services perform some action and send the response back.

While in some cases Web services may be utilized in an isolated form, it is natural to expect that Web services will be integrated as part of Web processes. A Web process is an abstraction of a business process. It comprises a number of logic steps (i.e., Web services), dependencies among services, process flow, routing rules, and logic to control and coordinate services and partners. The most prominent solution to describe Web processes is BPEL4WS (BPEL4WS, 2003). BPEL4WS (Process Execution Language for Web Services) is a specification that enables a business process to be performed using a number of Web services, possibly provided by several companies. Figure 1 illustrates how a Web process can model an e-commerce application.

WSDL and BPEL4WS specifications are shallow and focus only on syntactical descriptions of Web services and Web processes. As a consequence, these descriptions are inadequate for an automated discovery or composition of Web services. Much richer and deeper machine-processable descriptions are required.

Several researchers have pointed out that Web services should be semantically enabled (Cardoso & Sheth, 2003; Fensel, Bussler, & Maedche, 2002; Martin et al., 2004). Semantics are indispensable to develop distributed e-commerce applications over the Web due to its heterogeneity, autonomy, and distribution. Semantics articulate a well-defined set of common data elements or vocabulary allowing a rich description of Web services and Web processes which can be used by computers for an automatic or semi-automatic processing and management of e-commerce applications.

Figure 1. Example of a Web process modeling an e-commerce application



THE IMPORTANCE OF SEMANTICS FOR E-COMMERCE APPLICATIONS

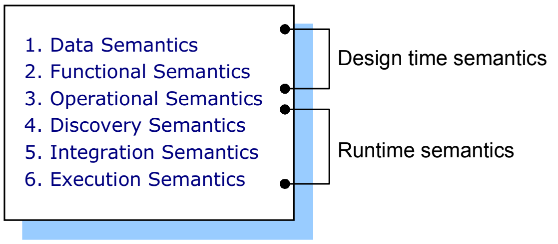
Semantic Web services will allow the automatic search (Klein & Bernstein, 2001), discovery (Verma et al., 2004), composition (Cardoso & Sheth, 2003), integration, orchestration (WSMX, 2004), and execution of inter-organizational services, making the Internet become a global common platform where organizations and individuals communicate among each other to carry out various e-commerce activities and to provide value-added services.

The idea of the "Semantic Web" (Berners-Lee, Hendler, & Lassila, 2001) catches on and researchers, as well as companies, have already realized the benefits of this great vision. Major companies and others are interested in creating industry-wide open e-business specifications for Semantic Web services and processes.

Different types of semantics can be used to enhance e-commerce applications. Semantics increase the description of capabilities, requirements, effects, and execution of Web services using ontologies (Gandon, 2002). E-commerce applications can benefit from six different kinds of semantics as illustrated in Figure 2.

These different types of semantics are discussed in the following sections.

Figure 2. Different types of semantics for e-commerce applications



Data Semantics

As e-commerce applications interconnect enterprises, Web services need to become available across systems, departments, and organizations. When organizations try to access and use local and remote Web services, they realize that their interfaces refer to incompatible data schema and cannot be called without a translation effort. In general, there is no common understanding which allows the data schema present in Web services' interfaces to be systematically manipulated.

Despite the fact that Web services use the same standardized technology, this incompatibility arises from semantic differences of data schema. In an e-commerce application, all the Web services take a set of data inputs and produce a set of data outputs represented in a WSDL specification file. However, the specification provides only syntactic and structural details of the input/output data. Each data schema is set up with its own structure and vocabulary. For example, a Web service may contain an output structure called "client" which includes the name, address, city, country, and telephone of a client; another Web service may have an input structure called "customer" and subdivides it into first name, last name, address, and tel. In such a scenario, how can the data output of the first Web service be transferred to the input of the second Web service? While the two structures do not match syntactically, they match semantically. To allow Web services to exchange data at the semantic level, the semantics of the input/output data has to be taken into account. Hence, if the data involved in Web service operations is annotated using an ontology (Patil, Oundhakar, Sheth, & Verma, 2004), the added semantics can be used in matching the semantics of the input/output of Web services when exchanging data, which was not possible when considering only syntactic information.

Related Content