

# 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 breed 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 appro-

priate 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:

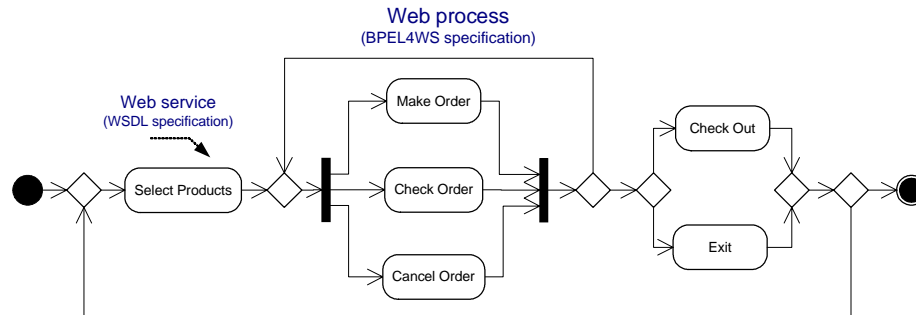
- A provider defines a standardized format for requests and responses for its Web services.
- A computer makes a request for the Web services across the network.
- The Web services perform some action and send the response back.

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

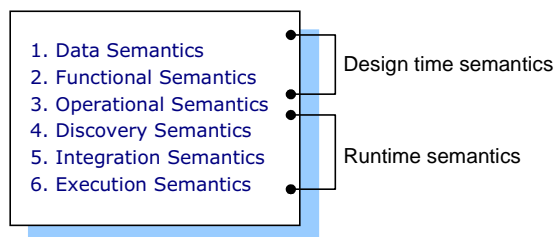
Semantic Web services will allow the automatic search (Klein & Bernstein, 2001), discovery (Verma et al., 2004), composition (Cardoso & Sheth, 2003), integration, orchestration (WSMX, 2004), and execution of inter-organizational services, making the Internet become a global common platform where organizations 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 seven 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.

## Functional Semantics

The goal of specifying the functionality of a service has a long tradition in computer science and includes work in the fields of program methodology, formal programming language semantics, and software engineering. The problems are complex, but the potential payoff is enormous.

Web service specifications are based on the WSDL standard that only defines syntactic characteristics. A WSDL document contains a set of definitions describing Web services using input and output messages, and operations. The signature of an operation provides only the syntactic details of the input data, output data, and operation's name.

Technological solutions to construct e-commerce applications based on Web services' operations signatures are not sufficient since services' functionality cannot be precisely expressed. Two services can have an operation with the same signature even if they perform entirely different functions. For example, a Web service called "add" that performs the addition of two integers taking the numbers as input and producing the sum as output (i.e.,  $add(x,y)=x+y$ ) will have the same signature of another service with the same name that performs the logarithmic addition of two numbers that are provided as input (i.e.,  $add(x,y)=\log x + \log y$ ).

As a step towards representing the functionality of a service, Web services can be annotated with functional semantics. This can be achieved by having a functional ontology in which each concept/class represents a well-defined functionality.

## Operational Semantics

When Web processes model e-commerce applications, suppliers, and customers define a binding agreement between the two parties, specifying operational constraints, also known as quality of service (QoS) requirements, such as goods to be delivered, deadlines, and cost of services or products.

The autonomy of Web services does not allow for business analysts to identify their operational metrics. When developing e-commerce applications it is indispensable to analyze and compute the QoS of the services and processes available to customers (Cardoso, Miller, Sheth, Arnold, & Kochut, 2004). This allows organizations to translate their vision into their business processes more efficiently, since Web processes can be designed according to QoS metrics. The management of QoS directly impacts the success of organizations participating in e-commerce applications. A good management of quality leads to the creation of quality products and services, which in turn fulfill customer expectations and satisfac-

tion. To achieve these objectives, operational metrics need to be described using operational semantics which represent the QoS model of services and processes.

Operational semantics (Cardoso, 2002) are very important, not only because they allow to specify the QoS of services, but also because they allow the computation of the QoS of Web processes that orchestrate Web services that use, for example, different unit systems. A Web process may orchestrate two Web services, one that uses the English metric system and the other that uses the International System of Units. In order to compute meaningful values for the QoS of the overall process, a conversion of units need to be done. For example, a conversion from miles to kilometers. Operational semantics can make this conversion task automatic and very simple.

E-commerce applications that have a worldwide spread inherently need to use operational semantics to resolve the differences that exist among operational measurement adopted and followed by different countries. Two ontologies can be devised to describe operational characteristics: *Domain Independent QoS* and *Domain Specific QoS* ontologies. The first ontology, accounts for the evidence that Web services in different domains can have different quality aspects. The second ontology is to be applied to services in all domains irrespective of their functionality or specialty.

## Discovery Semantics

After a Web service is developed and annotated with data, functional, and operational semantics, it has to be advertised to enable discovery. The Universal Description, Discovery, and Integration (UDDI) (UDDI, 2002) registry is a system to open doors for the success of service oriented computing.

UDDI registries enable global e-commerce by creating an organized approach to categorizing, storing, and retrieving information about the kind of services provided and who provides them. One of the major benefits of being listed in a public UDDI registry is that it provides equal exposure for all organizations. A large international organizations and a small locally owned company are listed in the same way.

Currently, UDDI only supports keyword (string) matching, which is considered the simplest type of syntactic matching. Therefore, the present discovery supported by UDDI is inefficient as services retrieved may be inadequate due to low precision (many services not wanted) and low recall (missed services that need to be considered). Effectively discovering relevant Web services in a scalable way is what is required to accelerate the adoption of Web services. To meet this challenge,

UDDI registries need to support not only syntactic matching, but also semantic matching (Verma et al., 2004). Semantic matching is the process of matching requests to Web services on the basis of the requested Web services functionality (functional semantics). Here again, Web services may not match syntactically but match semantically.

### Integration Semantics

Integration is a key issue in e-commerce because more and more companies are creating business-to-customer (B2C) and business-to-business (B2B) links to better manage their value chain. Automating inter-organizational Web processes across supply chains presents significant challenges (Stohr & Zhao, 2001). In order for these B2C and B2B links to be successful, Web services from multiple companies need to be integrated to interoperate seamlessly.

Compared to traditional distributed applications, Web services are highly autonomous and heterogeneous. Therefore, sophisticated methods are indispensable to support the integration of Web services. Here again, one possible solution is to explore the use of semantics to enhance the interoperability among Web services.

To achieve interoperability of e-commerce applications, it is necessary to address the problem of semantic integration—the identification of semantically similar objects that belong to different systems and the resolution of their schematic differences (Kashyap & Sheth, 1996). When Web services are put together, their interfaces (inputs and outputs) need to interoperate (Cardoso & Sheth, 2003); therefore, structural and semantic heterogeneity needs to be resolved. Structural heterogeneity exists because Web services use different data structures to define the parameters of their interfaces. Semantic conflicts occur when a Web service output connected to another service input does not use the same interpretation of the information being transferred. The general approach to semantic integration has been to map the local terms onto a shared ontology. Even though a shared ontology ensures total integration, constructing such an ontology has been costly (Rodríguez & Egenhofer, 2002). Recently however, significant progress is being made to deal with the issues of ontology evolution/management (Gandon, 2002) and multi-ontology environments (Fonseca, 2001), leading to increased momentum in developing and applying semantics to enable the integration of different systems.

### Execution Semantics

Execution semantics of a Web service (WSMX, 2004) encompasses the ideas of message sequence, conversa-

tion pattern of Web service execution, flow of actions, preconditions, and effects of Web service invocation. Some of these details may not be meant for sharing and some may be, depending on the organization and the application that is exposed as a Web service. In any case, the execution semantics of these services are not the same for all services and hence before executing or invoking a service, the execution semantics or requirements of the service should be verified.

Some of the issues and solutions with regard to execution semantics are inherited from traditional workflow technologies (Cardoso, 2005). However, the globalization of e-commerce, Web services, and processes result in additional issues. In e-commerce, using execution semantics can help in dynamically finding partners that will match not only the functional requirements, but also the operational requirements. Also, a proper model for execution semantics will help in coordinating activities in e-commerce applications that involve multiple parties.

### FUTURE TRENDS

According to TopQuadrant, a consulting firm that specializes in Semantic Web technologies, the market for semantic technologies will grow at an annual growth rate of between 60% and 70% until 2010, when it will grow from its current size of U.S. \$2 billion to \$63 billion. Existing applications that will add Semantic Web capabilities include Adobe's Extensible Metadata Platform and Oracle's database.

Semantic software is being experimentally used by banks to help them to comply with the U.S. government's Patriot Act (the Patriot Act requires banks to track and account for the customers with whom they do transactions), by a European police force to follow crime patterns, and by a telephone service provider to create applications that provides information about pay-per-view movies (Lee, 2005).

In addition to investment banks, the Metropolitan Life Insurance Company, the U.S. Department of Defense, and the Tennessee Valley Authority have also used semantic software to integrate enterprise data, to comply with federal regulations.

For Web services and Web processes to become a platform for semantic service oriented computing to fully support e-commerce application, academic and industrial researchers will need to create ontologies, terminologies, technologies, and products that enable sophisticated solutions for the annotation, advertisement, discovery, selection, composition, and execution of Web services.

Three major projects will provide, in the near term, several advances in the area of Semantic Web processes. The Semantic Web Services Initiative (SWSI), an initia-



tive of academic and industrial researchers, has been composed to create an infrastructure that combines the Semantic Web and Web services to enable the automation in all aspects of Web services. In addition, to providing further evolution of the ontology language for Web services, OWL-S (OWL-S, 2004), SWSI will also be a forum for working towards convergence of OWL-S with other research efforts in this area.

The METEOR-S (LSDIS, 2004) (METEOR for Semantic Web services) project is focused on the usage of semantics for the complete lifecycle of Semantic Web processes and targets research on four important areas, namely; annotation, discovery, composition, and execution of Web services.

Finally, DERI (DERI, 2004) is currently working on a project titled Semantic Web-Enabled Web Services (SWWS). Important objectives of the SWWS initiative include: providing a richer framework for Web service description and discovery, as well as, providing scalable Web service mediation middleware based on semantic data, process ontologies, and semantic interoperation.

## CONCLUSION

E-commerce has permanently changed the manner trading activities are carried out. Consumers can now acquire services and products from a diversity of Web sites. With the spread of the Web, a new technology has surfaced for the development of e-commerce application: Semantic Web processes.

This new trend in the global economy requires the ability to deploy e-commerce applications based on Web processes from the aggregation and orchestration of distributed Web services. Several researchers agree that it is essential for Web services to be semantically enabled in order to efficiently develop and execute Web processes.

In this chapter we have presented a set of challenges that the emergence of e-commerce and Semantic Web processes has brought to organizations. Designing Semantic Web processes entails research in two areas: Web services and the Semantic Web. We have presented how applying semantics to each of the steps in the Web process lifecycle can help address critical issues such as Web services annotation, discovery, selection, integration and execution.

## REFERENCES

Berners-Lee, T., Hendler, J., & Lassila, O. (2001, May 2001). The Semantic Web: A new form of Web content that

is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*.

BPEL4WS. (2003). *Specification: Business process execution language for Web services Version 1.1*. IBM. Retrieved February, from <http://www-106.ibm.com/developerworks/webservices/>

Cardoso, J. (2002, November 13-15). *Composing e-workflows using syntactic, operational and semantic information*. Paper presented at the International Conference WWW/Internet, Lisbon, Portugal.

Cardoso, J. (2005). Evaluating workflows and Web process complexity. In L. Fischer (Ed.), *Workflow handbook 2005* (pp. 284). Lighthouse Point, FL: Future Strategies Inc.

Cardoso, J., Miller, J., Sheth, A., Arnold, J., & Kochut, K. (2004). Modeling quality of service for workflows and Web service processes. *Web Semantics: Science, Services and Agents on the World Wide Web Journal*, 1(3), 281-308.

Cardoso, J., & Sheth, A. (2003). Semantic e-workflow composition. *Journal of Intelligent Information Systems (JIIS)*, 21(3), 191-225.

Christensen, E., Curbera, F., Meredith, G., & Weerawarana, S. (2001). *W3C Web Services Description Language (WSDL)*. Retrieved from <http://www.w3c.org/TR/wsdl>

DERI. (2004). *Digital Enterprise Research Institute (DERI)*. Retrieved from <http://www.deri.ie/>

Fensel, D., Bussler, C., & Maedche, A. (2002). *Semantic Web-enabled Web services*. Paper presented at the First International Semantic Web Conference, Sardinia, Italy.

Fonseca, F. (2001). *Ontology-driven geographic information systems*. Unpublished PhD Thesis.

Gandon, F. (2002). *Ontology engineering: A survey and a return on experience* (TR 4396). France: INRIA.

Kashyap, V., & Sheth, A. (1996). Schematic and semantic similarities between database objects: A context-based approach. *Very Large Data Bases (VLDB) Journal*, 5(4), 276-304.

Klein, M., & Bernstein, A. (2001). *Searching for services on the Semantic Web using process ontologies*. Paper presented at the International Semantic Web Working Symposium (SWWS), Stanford University, California, USA.

Lee, Y. L. (2005, April). Apps make Semantic Web a reality. *SDTimes*. Retrieved from <http://68.236.189.240/article/story-20050401-05.html>

LSDIS. (2004). *METEOR-S: Semantic Web services and processes*. Retrieved 2004, from <http://lsdis.cs.uga.edu/Projects/METEOR-S/index.php>

Martin, D., Paolucci, M., McIlraith, S., Burstein, M., McDermott, D., McGuinness, D., Parsia, B., Payne, T. R., Sabou, M., Solanki, M., Srinivasan, N., & Sycara, K. (2004). *Bringing semantics to Web services: The OWL-S approach*. Paper presented at the First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004), San Diego, CA.

OWL-S. (2004). *OWL-based Web service ontology*. Retrieved 2004, from <http://www.daml.org/services/owl-s/>

Patil, A., Oundhakar, S., Sheth, A., & Verma, K. (2004, May 17-22, 2004). *MWSAF - METEOR-S Web Service Annotation Framework*. Paper presented at the 13<sup>th</sup> Conference on World Wide Web, New York.

Rodríguez, A., & Egenhofer, M. (2002). Determining semantic similarity among entity classes from different ontologies. *IEEE Transactions on Knowledge and Data Engineering (in press)*.

Stohr, E. A., & Zhao, J. L. (2001). Workflow automation: Overview and research issues. *Information Systems Frontiers*, 3(3), 281-196.

UDDI. (2002). *Universal description, discovery, and integration*. Retrieved from <http://www.uddi.org/>

Verma, K., Sivashanmugam, K., Sheth, A., Patil, A., Oundhakar, S., & Miller, J. (2004). METEOR-S WSDI: A scalable infrastructure of registries for semantic publication and discovery of Web services. *Journal of Information Technology and Management (in print)*.

WSMX. (2004). *Web services execution environment (WSMX)*. Retrieved 2004, from <http://www.wsmx.org/>

## KEY TERMS

**BPEL4WS:** BPEL4WS stands for Business Process Execution Language for Web Services. It provides a language for the formal specification of business pro-

cesses. It extends the Web services model and enables it to support business transactions.

**Business Process:** A set of one or more linked activities which collectively realize a business objective or goal, normally within the context of an organizational structure.

**HTTP:** The underlying protocol used by the World Wide Web. HTTP defines how messages are formatted and transmitted between Web servers and browsers.

**Ontology:** An ontology provides the basis of representing, acquiring, and utilizing knowledge. Ontologies consist of entities, attributes, interrelationships between entities, domain vocabulary, and factual knowledge.

**Semantics:** The purpose of semantics is to assign a meaning to syntactical elements. For Web services, it is the agreed upon meaning of data, functions, QoS, etc., exchanged between two or more services.

**UDDI:** UDDI (Universal Description, Discovery, and Integration) is a registry for businesses to list themselves on the Internet. Its goal is to facilitate online transactions by enabling companies to find one another on the Web and make their systems interoperable for e-commerce. UDDI is often compared to a telephone book's white, yellow, and green pages.

**Web Process:** A Web process describes the logic to control and coordinate Web services participating in a process flow to carry out a specific goal. It directly addresses business process challenges such as control flow, data flow between Web services, long-running nested units of work, faults and compensation.

**Web Service:** A Web service describes a standardized way of integrating Web-based applications using open standards. A Web service is defined as a remotely callable function or procedure which communicates via the HTTP (HyperText Transfer Protocol) using standardized protocols.

**WSDL:** WSDL stands for Web Services Description Language. It is written in XML and it is used to describe and locate Web services.