Cardoso, J. and Amit P. Sheth, "Introduction to Semantic Web Services and Web Process Composition", Semantic Web Process: powering next generation of processes with Semantics and Web services, LNCS, Springer-Verlag Heidelberg, Vol. 3387, pp.1-13, 2005. ISBN: 3-540-24328-3.

Introduction to Semantic Web Services and Web Process Composition

Jorge Cardoso[1] and Amit Sheth[2]

[1] Departement of Mathematics and Engineering,
University of Madeira, Funchal, Portugal
jcardoso@uma.pt
[2] Large Scale Distributed Information Systems (LSDIS) Lab,
Department of Computer Science,
University of Georgia, GA, USA
amit@cs.uga.edu

Abstract. Systems and infrastructures are currently being developed to support Web services. The main idea is to encapsulate an organization's functionality within an appropriate interface and advertise it as Web services. While in some cases Web services may be utilized in an isolated form, it is normal to expect Web services to be integrated as part of Web processes. There is a growing consensus that Web services alone will not be sufficient to develop valuable Web processes due the degree of heterogeneity, autonomy, and distribution of the Web. Several researchers agree that it is essential for Web services to be machine understandable in order to support all the phases of the lifecycle of Web processes. This paper deals with two of the hottest R&D and technology areas currently associated with the Web — Web services and the Semantic Web. It presents how applying semantics to each of the steps in the Semantic Web Process lifecycle can help address critical issues in reuse, integration and scalability.

1 Introduction

E-commerce and e-services have been growing at a very fast pace. The Web coupled with e-commerce and e-services is enabling a new networked economy [1]. The scope of activities that processes span has moved from intra-enterprise workflows, predefined inter-enterprise and business-to-business processes, to dynamically defined Web processes among cooperating organizations.

There is a remarkable range for growth in trade through electronic interactions, simply because it can eliminate geographical distances in bringing buyers and sellers together. With the Internet dissemination and the e-commerce growth there is a shift from the traditional off-line distribution process based on organization's catalogs to on-line services. A shift that is marked by isolated initiatives guided by the business-to-customer and business-to-business promise of increased profit margins and reduced commission values. This leads us to the present situation where we can find diverse and numerous groups of on-line systems, most of them focused in one or in a few types of products. Therefore, organizations are increasingly faced with the challenge of managing e-business systems and e-commerce applications managing Web services, Web processes, and semantics. Web services promise universal interoperability and integration. The key to achieving this relies on the efficiency of discovering appropriate Web services and composing them to build complex processes. We will start this section by explaining what semantics are and their role and relationships with ontologies. We then explain the purpose of each of the Web process lifecycle phases.

2 Semantic Web Process Lifecycle

Semantic Web services will allow the semi-automatic and automatic annotation, advertisement, discovery, selection, composition, and execution of inter-organization business logic, making the Internet become a global common platform where organizations and individuals communicate among each other to carry out various commercial activities and to provide value-added services.

In order to fully harness the power of Web services, their functionality must be combined to create Web processes. Web processes allow representing complex interactions among organizations, representing the evolution of workflow technology. Semantics can play an important role in all stages of Web process lifecycle. The main stages of the Web process lifecycle are illustrated in Figure 1.

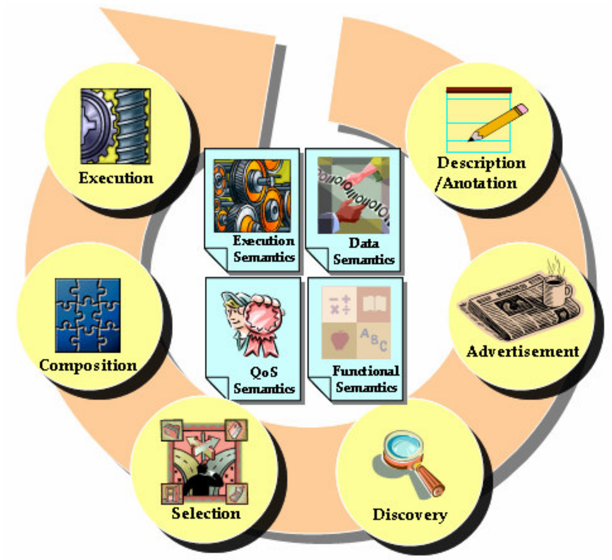

Fig. 1. Web process lifecycle and semantics.

The lifecycle of semantic Web processes includes the description/annotation, the advertisement, the discovery, the selection, the composition of Web services that makeup Web processes, and the execution of Web processes. All these stages are significant for the Web process lifecycle and their success.

2.1 Semantics and Ontologies

There is a growing consensus that Web services alone will not be sufficient to develop valuable and sophisticated Web processes due the degree of heterogeneity, autonomy, and distribution of the Web. Several researchers agree that it is essential for Web services to be machine understandable in order to allow the full deployment of efficient solutions supporting all the phases of the lifecycle of Web processes.

The idea and vision of the "Semantic Web" [2] catches on and researchers as well as companies have already realized the benefits of this great vision. Ontologies [3] are considered the basic building block of the Semantic Web as they allow machine supported data interpretation reducing human involvement in data and process integration.

An ontology "is a formal, explicit specification of a shared conceptualization. *Conceptualization* refers to an abstract model of phenomena in the world by having identified the relevant concepts of those phenomena. *Explicit* means that the type of concepts used, and the constraints on their use are explicitly defined. *Formal* refers to the fact that the ontology should be machine readable. *Shared* reflects that ontology should capture consensual knowledge accepted by the communities" [4].

When the knowledge about a domain is represented in a declarative language, the set of objects that can be represented is called the universe of discourse. We can describe the ontology of a program by defining a set of representational terms. Definitions associate the names of entities in the universe of discourse (e.g. classes, relations, functions or other objects) with human-readable text describing what the names mean and formal axioms that constrain the interpretation and well-formed use of these terms.

A set of Web services that share the same ontology will be able to communicate about a domain of discourse. We say that a Web service commits to an ontology if its observable actions are consistent with the definitions in the ontology.

Example: Benefits of ontologies for the travel industry. The Web has permanently changed the manner travel packages can be created. Consumers can now acquire packages from a diversity of Web sites including online agencies and airlines. With the spread of Web travel, a new technology has surfaced for the leisure travel industry: dynamic packaging. For the development of dynamic packaging solutions it is necessary to look in detailed at the technology components needed to enhance the online vacation planning experience. By transitioning from a third-party service in most markets, dynamic packaging engines can better tailor its package offerings, pricing and merchandising to consumer demand.

Currently, the travel industry has concentrated their efforts on developing open specifications messages, based on eXtensible Markup Language (XML), to ensure that messages can flow between industry segments as easily as within. For example, the OpenTravel Alliance (OTA) [5] is an organization pioneering the development and use of specifications that support e-business among all segments of the travel industry. The cumulative effort of various teams, individuals, associations, companies, and international organizations, including air, car, cruise, rail, hotel, travel agencies, tour operators and technology providers, has produced a fairly complete set of XML-based specifications for the travel industry (more than 140 XML specification files exist).

The current development of open specifications messages based on XML, such as the OTA schema, to ensure the interoperability between trading partners and working groups is not sufficiently expressive to guaranty an automatic exchange and processing of information. The development of a suitable ontology for the tourism industry is indispensable and will serve as a common language for travel-related terminology and a mechanism for promoting the seamless exchange of information across all travel industry segments.

The development of such an ontology can be used to bring together autonomous and heterogeneous Web services, Web processes, applications, data, and components residing in distributed environments. Semantics allow rich descriptions of Web services and Web processes that can be used by computers for automatic processing in various tourism related applications. The deployment of ontologies help articulate a well-defined set of common data elements or vocabulary that can support communication across multiple channels, expedite the flow of information, and meet travel industry and customer needs.

For the travel industry, the simplest form to construct an ontology is to retrieve rich semantic interrelationships from the data and terminology present in the XML-based OTA specifications already implemented [5] and available to organizations. This procedure is illustrated in Figure 2.

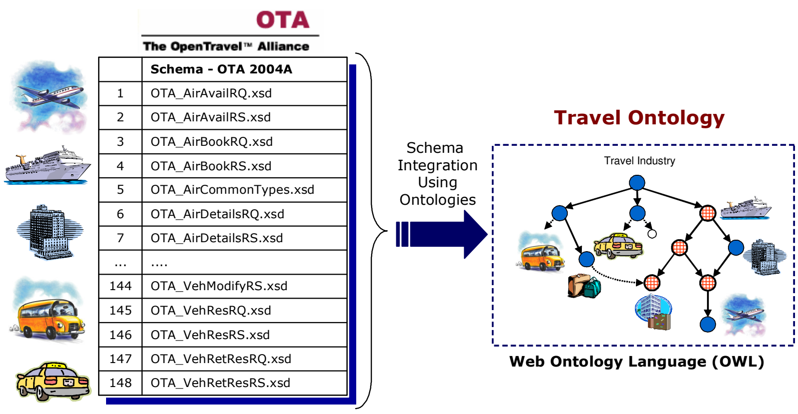

Fig. 2. Ontology for the travel industry.

One possible language to construct such an ontology is using the Web Ontology Language (OWL) [6] designed by the World Wide Web Consortium (W3C). The OWL is designed for use by applications that need to process the content of information instead of just presenting information to humans. OWL facilitates greater machine interpretability of Web content by providing additional vocabulary along with a formal semantics. It can be used to explicitly represent the meaning of terms in vocabularies and the relationships between those terms.

OWL is appropriate to develop an ontology for the travel industry since it is intended to be used when the information used by Web services needs to be processed by applications, as opposed to situations where the content only needs to be presented to humans.

The development of such an ontology lead to the spearhead and foster the cross-industry consensus needed to establish and maintain the most effective and widely used specifications designed to electronically exchange business data and information among all sectors of the travel industry.

This effort represents what can be achieved by the symbiotic synthesis of two of the hottest R&D and technology application areas: Web services and the semantic Web, as recognized at the Thirteenth International World Wide Conference (2004) and in the industry press. The intelligent combination of Web services and the semantic Web can start off a technological revolution with the development of semantic Web processes [7]. These technological advances can ultimately lead to a new breed of Web-based applications for the travel industry.

2.2 Semantics for Web Services

In Web services domain, semantics can be classified into the following types [8] illustrated in Figure 1:

- Functional Semantics
- Data Semantics
- QoS Semantics and
- Execution Semantics

These different types of semantics can be used to represent the capabilities, requirements, effects and execution of a Web service. In this section we describe the nature of Web services and the need for different kind of semantics for them.

Functional Semantics. The power of Web services can be realized only when appropriate services are discovered based on the functional requirements. It has been assumed in several semantic Web service discovery algorithms [9] that the functionality of the services is characterized by their inputs and outputs. Hence these algorithms look for semantic matching between inputs and outputs of the services and the inputs and outputs of the requirements. This kind of semantic matching may not always retrieve an appropriate set of services that satisfy functional requirements. Though semantic matching of inputs and outputs are required, they are not sufficient for discovering relevant services. For example, two services can have the same input/output signature even if they perform entirely different functions. A simple mathematical service that performs addition of two numbers taking the numbers as input and produce the sum as output will have the same semantic signature as that of another service that performs subtraction of two numbers that are provided as input and gives out their difference value as output. Hence matching the semantics of the service signature may result in high recall and low precision. As a step towards representing the functionality of the service for better discovery and selection, the Web services can be annotated with functional semantics. It can be done by having an ontology called Functional Ontology in which each concept/class represents a well-defined functionality. The intended functionality of each service can be represented as annotations using this ontology.

Data Semantics. All the Web services take a set of inputs and produce a set of outputs. These are represented in the signature of the operations in a specification file. However the signature of an operation provides only the syntactic and structural details of the input/output data. These details (like data types, schema of a XML complex type) are used for service invocation. To effectively perform discovery of services, semantics of the input/output data has to be taken into account. Hence, if the data involved in Web service operation is annotated using an ontology, then the added data semantics can be used in matching the semantics of the input/output data of the Web service with the semantics of the input/output data of the requirements. Semantic discovery algorithm proposed in [9] uses the semantics of the operational data.

QoS Semantics: After discovering Web services whose semantics match the semantics of the requirements, the next step is to select the most suitable service. Each service can have different quality aspect and hence service selection involves locating the service that provides the best quality criteria match. Service selection is also an important activity in web service composition [10]. This demands management of QoS metrics for Web services. Web services in different domains can have different quality aspects. For organizations, being able to characterize Web processes based on QoS has several advantages: a) it allows organizations to translate their vision into their business processes more efficiently, since Web processes can be designed according to QoS metrics, b) it allows for the selection and execution of Web processes based on their QoS, to better fulfill customer expectations, c) it makes possible the monitoring of Web processes based on QoS, and d) it allows for the evaluation of alternative strategies when Web process adaptation becomes necessary.

Execution Semantics. Execution semantics of a Web service encompasses the ideas of message sequence, conversation pattern of Web service execution, flow of actions, preconditions and effects of Web service invocation, etc. 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. However, the globalization of 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 like long running interactions and complex conversations. Also, a proper model for execution semantics will help in coordinating activities in transactions that involve multiple parties.

3 Phases of the Web Process Lifecycle

As stated previously, the lifecycle of semantic Web processes includes the description/annotation, the advertisement, the discovery, and the selection of Web services, the composition of Web services that makeup Web processes, and the execution of Web processes. In this section, we discuss the characteristics of each of these stages.

3.1 Semantic Web Service Annotation

Today, Web service specifications are based on standards that only define syntactic characteristics. Unfortunately, it is insufficient, since the interoperation of Web services/processes cannot be successfully achieved. One of the most recognized solutions to solve interoperability problems is to enable applications to understand methods and data by adding meaning to them.

Many tools are available to create Web services. Primarily programs written in Java or any object oriented language can be made into Web services. In technical terms any program that can communicate with other remote entities using SOAP [11] can be called a Web service. Since the development of Web services is the first stage in the creation of Web services, it is very important to use semantics at this stage. During Web service development data, functional and QoS semantics of the service needs to be specified.

All the Web services (operations in WSDL file [12]) take a set of inputs and produce a set of outputs. These are represented in the signature of the operations in a WSDL file. However the signature of an operation provides only the syntactic and structural details of the input/output data.

To effectively perform operations such as the discovery of services, semantics of the input/output data has to be taken into account. Hence, if the data involved in Web service operation is annotated using an ontology, then the added data semantics can be used in matching the semantics of the input/output data of the Web service with the semantics of the input/output data of the requirements.

The Meteor-S Web Service Annotation Framework (MWSAF) [13] provides a framework and a tool to achieve automatic and semi-automatic annotation of web services using ontologies.

Figure 3 illustrates one solution to annotate WSDL interfaces with semantic metadata based on relevant ontologies [14]. A Web service invocation stipulate an input interface that specifies the number of input parameters that must be supplied for a proper Web service realization and an output interface that specifies the number of outputs parameters to hold and transfer the results of the Web service realization to other services.

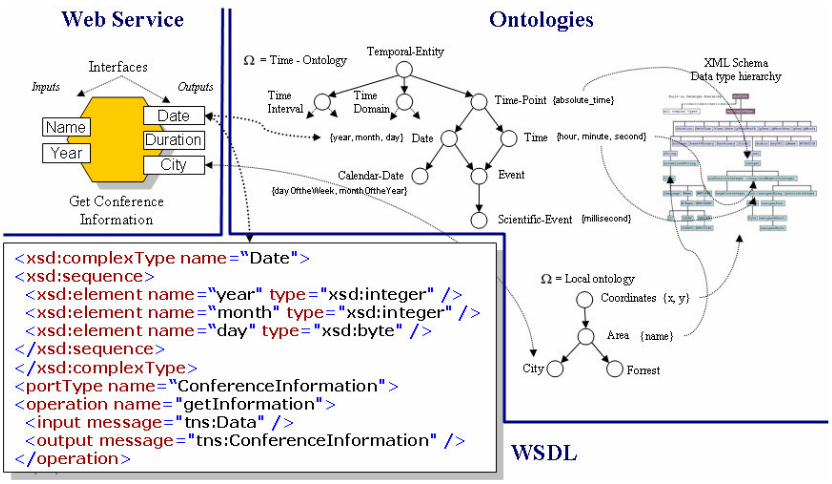

Fig. 3. Semantic annotation of a Web service specified with WSDL

3.2 Semantic Web service Advertisement

After the service is developed and annotated, it has to be advertised to enable discovery. The UDDI registry is supposed to open doors for the success of service oriented computing leveraging the power of the Internet. Hence the discovery mechanism supported should scale to the magnitude of the Web by efficiently discovering relevant services among tens and thousands (or millions according to the industry expectations) of the Web services.

The present discovery supported by UDDI is inefficient as services retrieved may be inadequate due to low precision (many services you do not want) and low recall (missed the services you really need to consider). Effectively locating relevant services and efficiently performing the search operation in a scalable way is what is required to accelerate the adoption of Web services. To meet this challenge, Web service search engines and automated discovery algorithms need to be developed. The discovery mechanisms supported need to be based on Web services profiles with machine process-able semantics.

3.3 Semantic Web Service Discovery

This stage is the process of discovering appropriate services before selecting a specific Web service to and binding it to a Web processes [15]. The search of Web services to model e-commerce applications differs from the search of tasks to model traditional processes. One of the main differences is in terms of the number of Web services available to the composition process. In the Web, potentially thousands of Web services are available. One of the problems that need to be solved is how to efficiently discover Web services [10].

The discovery of Web services has specific requirements and challenges as compared to previous work on information retrieval systems and information integration systems. Several issues need to be considered:

- Precision of the discovery process. The search has to be based, not only on syntactic information, but also on data, functional, and QoS semantics.
- Enable the automatic determination of the degree of integration of the discovered Web services and a Web process host.
- The integration and interoperation of Web services differs from previous work on schema integration due to the polarity of the schema that must be integrated [10].

Typically, a cluster of Web services that match initial requirements is constructed. In the next phase (semantic Web service selection), we selected, from the cluster, the Web service that more closely matches our requirements. The cluster which contains the list of other services, which also match the requirements, is maintained. This is because a service may be chosen later in case of failure or breach of contract.

3.4 Semantic Web service Selection

Web service selection is a need that is almost as important as service discovery. After discovering Web services whose semantics match the semantics of the requirement, the next step is to select the most suitable service. Each service can have different quality aspect and hence service selection involves locating the service that provides the best quality criteria match.

Service selection is also an important activity in Web service composition [10]. This demands management of QoS metrics for Web services. Web services in different domains can have different quality aspects. These are called *Domain Independent* QoS metrics. There can be some QoS criteria that can be applied to services in all domains irrespective of their functionality or specialty. These are called *Domain Specific* QoS metrics. Both these kind of QoS metrics need shared semantics for interpreting them as intended by the service provider. This could be achieved by having an ontology (similar to an ontology used for data semantics) that defines the domain specific and domain independent QoS metrics.

3.5 Semantic Process Composition

The power of Web services can be realized only when they are efficiently composed into Web process. This requires a high degree of Interoperability among Web services. Interoperability is a key issue in e-commerce because more and more companies are creating business-to-customer and business-to-business links to better manage their value chain. In order for these links to be successful, heterogeneous systems from multiple companies need to interoperate seamlessly. Automating inter-organizational processes across supply chains presents significant challenges [16].

Compared to traditional process tasks, Web services are highly autonomous and heterogeneous. Sophisticated methods are indispensable to support the composition of Web process. Here again, one possible solution is to explore the use of semantics to enhance interoperability among Web services.

This stage involves creating a representation of Web processes. Many languages like BPEL4WS [17], BPML [18] and WSCI [19] have been suggested for this purpose. The languages provide constructs for representing complex patterns [20] of Web service compositions. While composing a process, four kinds of semantics have to be taken into account. The process designer should consider the functionality of the participating services (functional semantics), data that is passed between these services (data semantics), the quality of these services, the quality of the process as a whole (QoS semantics) and the execution pattern of these services, the pattern of the entire process (Execution semantics). Since Web process composition involves all kind of semantics, it may be understood that semantics play a critical role in the success of Web services and in process composition.

3.6 Execution of Web processes

Web services and Web processes promise to ease several of nowadays infrastructure challenges, such as data, application, and process integration. With the emergence of Web services, workflow management systems (WfMSs) become essential to support, manage, enact, and orchestrate Web processes, both between enterprises and within the enterprise. Several researchers have identified workflows as the computing model that enables a standard method of building Web process applications and processes to connect and exchange information over the Web [21].

Execution semantics of a Web service encompasses the ideas of message sequence (e.g., request-response, request-response), conversation pattern of Web service execution (peer-to-peer pattern, global controller pattern), flow of actions (sequence, parallel, and loops), preconditions and effects of Web service invocation, etc.

Traditional formal mathematical models (Process Algebra [22]), concurrency formalisms (Petri Nets [23], state machines [24]) and simulation [25] techniques) can be used to represent execution semantics of Web services. Formal modeling for workflow scheduling and execution are also relevant [26]. With the help of execution semantics process need not be statically bound to component Web services. Instead, based on the functional and data semantics a list of Web services can be short listed, QoS semantics can be used to select the most appropriate service, and execution semantics the service can be bound to a process and used to monitor process execution.

3.7 Semantic Web Process QoS

New trading models, such as e-commerce, require the specification of QoS metrics such as products or services to be delivered, deadlines, quality of products, and cost of service. To enable adequate QoS management, research is required to develop mechanisms that semantically specify, compute, monitor, and control the QoS of the products or services to be delivered [10, 27].

In e-commerce and e-business Web processes, suppliers and customers define a binding agreement between the two parties, specifying QoS items such as services to be delivered, deadlines, and cost of services. The management of QoS metrics of semantic Web processes directly impacts the success of organizations participating in e-commerce. Therefore, when services or products are created or managed using Web processes, the underlying WfMS must accept the specifications and be able to estimate, monitor, and control the QoS rendered to customers.

A comprehensive QoS model that allows the description of Web processes components from a QoS perspective have already been developed [28]. One of the models includes three dimensions: time, cost, and reliability. The QoS model is coupled with an algorithm (the SWR algorithm [28]) to automatically compute the overall QoS of Web processes. These developments can be easily applied to automatically compute the duration, cost, and reliability of Web processes.

4. Ongoing Work

The industrial research related to semantic Web services depends on the ongoing development of open standards that ensure interoperability between different implementations. Several initiatives have been conducted with the intention to provide platforms and languages that will allow easy integration of heterogeneous systems. The standardization efforts for the technologies that underlie Web services include Simple Object Access Protocol (SOAP)[29], Web Services Description Language (WSDL)[12], Universal Description, Discovery and Integration (UDDI)[30], and process description languages, such as Business Process Execution Language for Web Services (BPEL4WS)[17] (from Microsoft, IBM, BEA).

Recently, the Semantic Web Services Initiative (SWSI)[31], an initiative of academic and industrial researchers has been composed to create infrastructure that combines Semantic Web and Web Services to enable the automation in all aspects of Web services. In addition to providing further evolution of OWL-S [32], SWSI will also be a forum for working towards convergence of OWL-S with the products of the WSMO[33]/WSML[34]/WSMX[35] research effort.

WSMO is a complete ontology for the definition of Semantic Web Services. It follows the WSMF as a vision of Semantic Web Services. WSML is a family of languages that allow Semantic Web Service designers to define Semantic Web Services in a formal language. The WSMX provides a standard architecture for the execution of Semantic Web Services.

Besides these major standards and initiatives, there are two ongoing projects being developed in the US, the LSDIS METEOR-S project [36], and in Europe, the DERI SWWS project [37].

The METEOR-S (METEOR for Semantic Web services) project is focused on the usage of semantics for the complete lifecycle of semantic Web processes, namely, annotation, discovery, composition, and execution.

DERI [38] is currently working on a project titled Semantic Web enabled Web Services (SWWS). DERI researchers recognize that to use the full potential of Web services and the technology around UDDI, WSDL and SOAP, it is indispensable to use semantics, since current technologies provide limited support for automating Web service discovery, composition and execution. 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.

5. Conclusions

Systems and infrastructures are currently being developed to support Web services. The main idea is to encapsulate an organization's functionality within an appropriate interface and advertise it as Web services.

While in some cases Web services may be utilized in an isolated form, it is normal to expect Web services to be integrated as part of Web processes. There is a growing consensus that Web services alone will not be sufficient to develop valuable Web processes due the degree of heterogeneity, autonomy, and distribution of the Web.

For example, a new requirement for the travel industry is the ability to dynamically compose travel packages from the aggregation and orchestration of distributed Web services. Current approaches, using XML-based specification messages, are not sufficient to enable the creation of dynamic travel packages. One solution is the use of ontologies to overcome semantic problems that arise from the autonomy, heterogeneity, and distribution of Web services.

Several researchers agree that it is essential for Web services to be machine understandable in order to support all the phases of the lifecycle of Web processes. In this paper we have presented a set of challenges that the emergence of 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 Semantic Web Process lifecycle can help address critical issues in reuse, integration and scalability.

References

1. Sheth, A.P., W.v.d. Aalst, and I.B. Arpinar, Processes Driving the Networked Economy. IEEE Concurrency, 1999. 7(3): p. 18-31.
2. W3C, W3C Semantic Web Activity. http://www.w3.org/2001/sw/. 2004.
3. Uschold, M. and M. Gruninger, Ontologies: Principles, methods and applications. Knowledge Engineering Review, 1996. 11(2): p. 93-155.
4. Gruber, T.R., Toward principles for the design of ontologies used for knowledge sharing. International Journal of Human-Computer Studies, 1995. 43(5-6): p. 907-928.
5. OTA, OpenTravel Alliance. www.opentravel.org, 2004.
6. OWL, Web Ontology Language (OWL). http://www.w3.org/2004/OWL/, 2004, World Wide Web Consortium (W3C).

7. Cardoso, J. and A.P. Sheth. Semantic Web Processes: Semantics Enabled Annotation, Discovery, Composition and Orchestration of Web Scale Processes. in Fourth International Conference on Web Information Systems Engineering (WISE'03). 2003. Roma, Italy.
8. Sivashanmugam, K., et al., Metadata and Semantics for Web Services and Processes, in Datenbanken und Informationssysteme (Databases and Information Systems). Festschrift zum 60. Geburtstag von Gunter Schlageter, W. Benn, et al., Editors. 2003: Hagen, Germany. p. 245-271.
9. Paolucci, M., et al. Importing the Semantic Web in UDDI. in Proceedings Web Services, E-Business and Semantic Web Workshop, CAiSE 2002. 2002. Toronto, Canada.
10. Cardoso, J. and A. Sheth, Semantic e-Workflow Composition. Journal of Intelligent Information Systems (JIIS). 2003. 21(3): p. 191-225.
11. Graham, S., et al., Building Web Services with Java: Making Sense of XML, SOAP, WSDL, and UDDI. 2002: SAMS.
12. Christensen, E., et al., W3C Web Services Description Language (WSDL). http://www.w3c.org/TR/wsdl, 2001.
13. Patil, A., et al. MWSAF - METEOR-S Web Service Annotation Framework. in 13th Conference on World Wide Web. 2004. New York City, USA.
14. Cardoso, J., F. Curbera, and A. Sheth. Tutorial: Service Oriented Archiectures and Semantic Web Processes. in The Thirteenth International World Wide Web Conference (WWW2004). 2004. New York, USA.
15. Verma, K., et al., METEOR-S WSDI: A Scalable Infrastructure of Registries for Semantic Publication and Discovery of Web Services. Journal of Information Technology and Management (in print), 2004.
16. Stohr, E.A. and J.L. Zhao, Workflow Automation: Overview and Research Issues. Information Systems Frontiers, 2001. 3(3): p. 281-196.
17. BPEL4WS, Web Services. http://www-106.ibm.com/developerworks/webservices/, 2002, IBM.
18. BPML, Business Process Modeling Language. http://www.bpmi.org/, 2004.
19. WSCI, Web Service Choreography Interface (WSCI) 1.0. http://www.w3.org/TR/wsci/, 2002, World Wide Web Consortium (W3C).
20. Aalst, W.M.P.v.d., et al. Advanced Workflow Patterns. in Seventh IFCIS International Conference on Cooperative Information Systems. 2000.
21. Fensel, D. and C. Bussler, The Web Service Modeling Framework. http://www.cs.vu.nl/~dieter/ftp/paper/wsmf.pdf, 2002, Vrije Universiteit Amsterdam (VU) and Oracle Corporation.
22. Bergstra, J.A., A. Ponse, and S.A. Smolka, Handbook of Process Algebra. 2001: Elsevier.
23. Aalst, W.M.P.v.d., The Application of Petri Nets to Workflow Management. The Journal of Circuits, Systems and Computers, 1998. 8(1): p. 21-66.
24. Hopcroft, J.E., R. Motwani, and J.D. Ullman, Introduction to Automata Theory, Languages, and Computation. 2000, Mass.: Addison-Wesley Publishing Company.
25. Bosilj, V., M. Stemberger, and J. Jaklic, Simulation Modelling Toward E-Business Models Development. International Journal of Simulation Systems, Science & Technology, Special Issue on: Business Process Modelling, 2001. 2(2): p. 16-29.
26. Attie, P., et al. Specifying and Enforcing Intertask Dependencies. in Proceedings 19th Intlernational Conference on Very Large Data Bases. 1993. Dublin, Ireland: Morgan Kaufman.
27. Cardoso, J., A. Sheth, and J. Miller. Workflow Quality of Service. in International Conference on Enterprise Integration and Modeling Technology and International Enterprise Modeling Conference (ICEIMT/IEMC’02). 2002. Valencia, Spain: Kluwer Publishers.

28. Cardoso, J., et al., Quality of service for workflows and web service processes. Web Semantics: Science, Services and Agents on the World Wide Web Journal, 2004. 1(3): p. 281-308.
29. SOAP, Simple Object Access Protocol 1.1. http://www.w3.org/TR/SOAP/, 2002.
30. UDDI, Universal Description, Discovery, and Integration. http://www.uddi.org/, 2002.
31. SWSI, Semantic Web Services Initiative (SWSI). http://www.swsi.org/, 2004.
32. OWL-S, OWL-based Web Service Ontology. http://www.daml.org/services/owl-s/, 2004.
33. WSMO, Web Services Modeling Ontology (WSMO). http://www.wsmo.org/, 2004.
34. WSML, Web Service Modeling Language (WSML). http://www.wsmo.org/wsml/index.html, 2004.
35. WSMX, Web Services Execution Environment (WSMX). http://www.wsmx.org/, 2004.
36. LSDIS, METEOR-S: Semantic Web Services and Processes. http://lsdis.cs.uga.edu/Projects/METEOR-S/index.php, 2004.
37. SWWS, Semantic Web Enabled Web Service. http://swws.semanticweb.org/, 2004, Digital Enterprise Research Institute (DERI).
38. DERI, Digital Enterprise Research Institute (DERI). http://www.deri.ie/, 2004.