Applying Semantic Web Services to Web-based IS and Applications

Ranjit Bose
Anderson School of Management, University of New Mexico, Albuquerque, NM 87131
E-mail: bose@mgt.unm.edu

Abstract

Semantic Web and Web Services are emerging Web-based technologies, each having its niche contribution within the context of developing Web-based information systems and applications. Their combination, called Semantic Web Services (SWS), provides several potential opportunities and challenges to all individual stakeholders of e-business. This study aims to bring awareness and discussion of important issues related to the impact SWS would likely have on the design, development, and utilization of e-business systems. Additionally, the research sheds some light for IS researchers and practitioners by identifying several research areas for further advancement of SWS technology applications within e-business.

Keywords: Semantic Web, Web Services, e-business

1. Introduction

In recent years, the Web (WWW) has become the platform of choice for building information systems (IS) applications. Web-based technologies are having a major impact on design, development, and implementation processes of all types of IS applications. Distributed IS over the Internet and intranets enables the dissemination of IS applications at a reasonable cost. Thus considerable attention is now being given to the design and development of Web Services (WS) based IS.

In spite of these trends, most of today’s Web content is only suitable for human consumption. That is, the meaning of Web content is not machine-interpretable. Some of the typical uses of the Web by individuals today involve searching for and making use of information, reviewing catalogs on online stores and ordering products by filling out forms. These activities are not particularly well supported by software tools other than search engines. Currently there are tools available that can retrieve texts, split them into parts, check spelling, count the number of words and so on, but when it comes to interpreting sentences and extracting useful information for users, the capabilities of current software are still very limited.

The next generation Web, called the Semantic Web (SW), is envisioned to represent Web content in a form that is equally meaningful to the humans as well as to the computers [3]. The SW, which is a Web technology that provides machine interpretation and processing capability of the existing Web information, is designed to live on top of the existing Web without modifying it. This extension of the current Web, to one where information is given well-defined meaning (semantics), better enables computers and people to work in cooperation and to use intelligent techniques to take advantage of these representations. Thus SW is intended to create a universal medium for information exchange by giving semantics, in a manner understandable by machines to the content of resources. The World Wide Web Consortium (W3C) spearheaded the SW development project, which is the international standardization body for the Web.

The SW builds on this idea with the aim of providing machine-processable information, which is made possible by the addition of “semantic markup” to Web-based resources. In contrast to HTML markup, semantic markup includes definitions of concepts and relationships involved in the information being transmitted. This richer definition allows client programs to process the contents instead of merely the markup. The idea behind the SW is to add defining tags to information within Web pages and to provide links to this information so that an application can discover data and make associations between different data elements.

As the dynamics of the global e-business markets increase, the need for accurate, more diverse and immediate information by all types of users will continue to grow [20]. At the same time, the availability of information from diverse sources through the Web also provides the opportunity to widen the scope of input to Web-based IS, if this information can be made accessible through automated means. SW technologies are primarily designed to be a powerful means for supporting information-centric interoperability and to provide intelligent content for use by automated systems. Yet at the same time, organizations are experimenting with moving their e-business applications software design and development approach to WS [16], which are “services” offered by one application to other applications via the Web [4,
Developers can aggregate the services to form an end-user application, enable business transactions, or create new services. WS are software components and applications that use Internet technologies and standards, and they can be accessed through the Internet or intranet. They can bind together autonomous heterogeneous applications, data services, and components residing in distributed environments. Thus WS, like SW, facilitate interoperability and integration, which have long been sought after by e-business application designers and developers [17].

In summary, SW and WS are emerging Web-based technologies, each having its niche contribution, as mentioned above, within the context of developing Web-based information systems and applications. Their combination, called Semantic Web Services (SWS), provides several potential opportunities and challenges to all individual stakeholders of e-business. This study aims to bring awareness and discussion of important issues related to the impact SWS would likely have on the design, development, and utilization of e-business systems. Additionally, the research sheds some light for IS researchers and practitioners by identifying several research areas for further advancement of SWS technology applications within e-business.

2. Semantic Web

The word semantics implies the study of meaning [12]. The SW provides a common framework to: (a) represent data on the Web or as a database that is globally linked, in a manner understandable by machines, to the content of documents on the Web; and (b) allow data to be shared and reused across application, enterprise, and enterprise boundaries [10, 12]. If a computer understands the semantics of a document, it doesn’t just interpret the series of characters that make up that document: it understands the document’s meaning. SW technologies therefore help separate meanings from data, document content, or application code, using technologies based on open standards. SW technologies represent meaning using ontologies and provide reasoning through the relationships, rules, logic, and conditions represented in those ontologies [9].

The metadata, data about data, is one of the core organizing concepts behind the W3C’s vision of the SW. To represent the SW knowledge, one uses the following technologies: (a) RDF (a standard syntax for describing data); (b) RDF Schema (a standard means of describing the properties of that data); and (c) Ontologies – defined using OWL, the Web Ontology Language (a standard means of describing relationships between data items).

2.1 RDF – Resource Description Framework

The RDF is the underlying unified data model for representing semantics. The data model and XML serialization syntax is used for describing resources both on and off the Web. RDF makes use of unique identifiers (URI, Uniform Resource Identifier) for describing metadata. URIs are used to describe things, also called resources, which could be people, places, documents, images, databases, etc. All RDF applications adopt a common convention for identifying these things. A subset of URI, the Uniform Resource Locator or URL, is concerned with the location and retrieval of resources, while URI is a unique identifier for things or resources that we describe but that may not necessarily be retrievable. However, RDF provides a consistent, standardized way to describe and query Internet resources, from text pages and graphics to audio files and video clips.

2.2 RDF Schema

RDF Schema is a standard means to describe the properties of resources defined using RDF. For example, if we say that a resource is of a particular type, or has a certain relationship to another resource, or has some specified attribute, we need to uniquely identify these descriptive concepts. RDF uses URIs for these too. Different development communities can invent new descriptive properties (such as person, employee, price, and classification) and assign URIs to these properties. Since the assignment of URIs is decentralized, one can be sure that uniquely named descriptive properties don't get mixed up when we integrate metadata from multiple sources.

Both RDF and RDF Schema are based on XML and XML Schema. The existence of standards for describing data or other resource (RDF) and data or other resource attributes (RDF Schema) enables the development of a set of readily available tools to read and exploit data or other resource from multiple sources. The degree to which different applications can share and exploit data or other resource is called syntactic interoperability. The more standardized and widespread these data manipulation tools are, the higher the degree of syntactic interoperability, and the easier and more attractive it becomes to use the SW approach as opposed to a point-to-point integrated solution.

2.3 Ontology
The term ontology, also known as a domain model, refers to a hierarchical data structure containing the relevant entities and their relationships and rules within a specific domain [11]. Ontologies define data models in terms of classes, subclasses, and properties. Before multiple applications can truly understand data – and treat it as information – semantic interoperability is required. Syntactic interoperability is about parsing data correctly, which requires mapping between terms. Semantic interoperability is about content analysis, which requires formal and explicit specifications of domain models, which define the terms used and their relationships.

Ontologies provide shared semantics to metadata, enabling a degree of semantic interoperability. The challenge for the designer is to identify how to represent, create, manage and use both ontologies as shared knowledge representations, but also large volumes of metadata records used to annotate Web resources of a diverse kind. That is, to achieve knowledge integration through ontologies, semantic metadata and databases of annotations. The aim of building ontologies is to share and reuse knowledge. Since the SW is a distributed network, there are different ontologies that describe semantically equivalent things. Therefore, it is necessary to map elements of these ontologies if one wants to process information across applications or domains.

W3C’s OWL (Ontology Working Language) Web Ontology Language is a poplar language used to represent ontologies on the Web. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema by providing additional vocabulary along with formal semantics. The basic components of OWL include classes, properties, and individuals. Classes are the basic building blocks of OWL ontology. A class is a concept in a domain. Classes usually constitute a taxonomic hierarchy (for example, a subclass-superclass hierarchy). Properties have two main categories: (a) Object properties, which relate individuals to other individuals; and (b) Datatype properties, which relate individuals to datatype values, such as integers, float, and strings. OWL makes use of XML Schema for defining datatypes. Individuals are instances of classes, and properties can relate one individual to another.

Examples of ontologies include catalogs for online shopping sites like Amazon.com, domain-specific standard terminology like UNSPSC (a terminology used for products and services), or various taxonomies on the Web, like the “My Yahoo” categories.

3. Web Services

Enterprises have wanted to move to a more “plug-and-play” business IT infrastructures for some time [13]. They can now realize this vision by adopting WS, which provide them the ability to design, create, and deliver applications more quickly and cost-effectively. The significance of WS is that it moves software and programming to being truly network-centric; the network becomes the heart of the system, linking all WS [19]. Packages of code can be concatenated to produce highly tailored and quickly changed processes. In the past, once software was programmed to handle a process in a specific way, it was essentially “hard wired” because the process could not change until the software was modified. The idea behind WS is that a process is defined at the time it is executed, because each WS decides at that time which of its many options to use to answer the current request. Thus the new-generation e-business systems could compose services dynamically, as needed, by binding several lower-level services. WS-based approach to application development overcomes major limitations of traditional Web/software development and evolution [23]. Though WS are relatively young, a new breed of Web applications based on WS have been successfully deployed in financial, manufacturing, travel, and e-business sectors [15].

WS is a very hot topic for many reasons. First, WS development people see it as the next generation of distributed services because it allows the building of large, complex systems by linking together any number of modules that each performs one or more tasks. Second, it makes the Internet the hub of computing, which aims to ease inter-enterprise computing as required in B2B e-business environments. Finally, it permits flexible systems not possible in the past, because a Web Service can choose which other WS to utilize at the time it needs one. Thus WS could also leverage the creation of business network through which aggregations of products and services can flow freely. They have the potential for transforming how businesses and enterprises interact within themselves and with other enterprises to carry out B2B transactions. Besides being a catalyst for transformation, WS offer businesses the agility to configure and implement IS in pace with the business growth and demands. The flexibility, the ability to evolve and grow, and potential cost savings make WS very attractive for a range of enterprise and e-business applications. WS is expected to fuel a new wave of electronic business, application integration, and business-to-business (B2B) interactions, as the industry moves towards application and service integration, rather than dedicated system development that require extensive design, deployment and integration efforts [7].
WS provide IS designers and builders the ability to design, create, and deliver applications more quickly and cost-effectively [5, 14]. The three fundamental methods in a WS system are: register; discover; and bind. When an organization exposes services on its Web site, the services are ready to use, but only by clients that know in advance where the service is and how to access it. To increase the number of potential customers, WS providers will register their services in a public registry. Registries provide the equivalent of a phone book for WS. Once services have been registered, would-be service consumers can locate systems that provide relevant operations and information. Registry specifications include interfaces for various kinds of query, including industry type, name of service, key word descriptions, etc. The registry provides enough information for service consumers to discover WS dynamically and to learn how to access them. Once a service consumer has located a relevant service in the registry, the consumer has the information it needs to use that service. Connecting to the service provider, passing correctly formed input, and receiving the results, is called binding to the service. Thus WS can bind together autonomous heterogeneous applications, data services, and components residing in distributed environments. They facilitate interoperability and integration, and hence the common intersection with the SW technologies.

WS represent a standards-based model for creating and connecting distributed applications across the Internet. They include simple object access protocol (SOAP), Web services definition language (WSDL), and universal description, discovery, and integration (UDDI).

4. Semantic Web Services

Although SW concepts and WS specifications were developed by substantially separate groups of researchers and organizations, there is now a growing understanding of the benefits of combining the two to form “Semantic Web Services” (SWS).

SWS [18] will combine the meaningful representation of information, relationships and context from the SW community with the dynamic discovery and binding of the WS specifications to create a system that will allow contextual discovery and use of WS [1, 8].

The current WS specification for discovery and registration defines registry queries that are based on the use of keywords for service descriptions. Although the registry does allow a service provider to state the (standard) taxonomy from which descriptive terms are drawn, taxonomies cannot represent the full range of relationships possible with an ontology.

In a SWS environment, on the other hand, service registration includes a semantically rich description of the service model [12]. Service consumers, then, can discover service providers based on reasoning about the context within which the service is defined. Moving to a semantics-based discovery process will further loosen the coupling between service consumers and service providers, allowing for greater flexibility across the system.

SWS will also improve on current methods of binding a consumer to a service [22]. When service operations are registered using current WS specifications, the parameter and return types must be exactly defined. This is reasonable (even necessary) when the services are based on standard or well-known definitions; for instance, when an industry or organization has created interface specifications for all operations are registered using current WS specifications, the parameter and return types must be exactly defined. This is reasonable (even necessary) when the services are based on standard or well-known definitions; for instance, when an industry or organization has created interface specifications for all operations in a given domain. However, the type definitions usually mean that a would-be consumer must include classes that can only be created at design time. This mitigates against the total flexibility of run-time discovery.

If, however, the service is defined using an ontology, it is possible for would-be consumers to learn the context in which the parameter and return classes are defined. This will allow the consumer to relate those classes to classes in its own information model, potentially allowing the consumer to construct and populate objects that can be passed to the service, even though the classes themselves are not part of the consumer’s internal representation.

The DARPA Agent Markup Language [2] project is defining DAML-S, a DAML vocabulary for service definitions. As the DAML-S specification matures, it should provide a standard way to define services based on an ontology. This will help to allow context-based service discovery. In addition to defining the service model, ontology representation in the SWS world must have the ability to describe the domain model of the service.

5. Future Research

WS have added a new level of functionality to the current Web by taking a first step towards seamless integration of distributed software components using Web standards. Nevertheless, current WS technologies, that include SOAP, WSDL, and UDDI, operate at a syntactic level, therefore, although they support interoperability (i.e., interoperability between the many diverse application development platforms that exist today) through common standards, they still require human interaction to a large extent: the human programmer has to manually search for appropriate WS in order to combine them in a useful manner, which limits scalability and greatly curtails the added
economic value envisioned with the advent of WS. For automation of tasks, such as WS discovery, composition and execution, semantic description of WS needs to be developed, implemented, and tested.

In order to provide the basis for SWS, a fully fledged framework needs to be developed – starting with a conceptual model, continuing with a formal language to provide formal syntax and semantics (based on different logics in order to provide different levels of logical expressiveness and reasoning) for the conceptual model, and ending with an execution environment, that glue all the components that use the language for performing various tasks that would eventually enable automation of service as envisioned. One of the primary objectives of this framework would be to narrow the understanding and interpretation gap between the semantic and the syntactic descriptions of services [24].

Research on the granularity of WS needs to be examined within the context of the above framework. WS applications include ordering books, buying travel tickets, booking rental cars, buying market research information, making restaurant reservations and others. Some of these WS are elementary and some are composite. Elementary WS do NOT rely on other WS to fulfill external requests. In contrast, composite services integrate multiple component services (either elementary or composite) to fulfill a request. For example, a travel assistant service may combine car rental, hotel and possibly flight reservation services. Composition of elementary services is essential to the growth of the WS model of application interaction [21].

In summary, the heterogeneity, autonomy and dynamic characteristics of WS make the development of an integrated framework for WS composition and execution a formidable task.

SWS require an ontological underpinning. High level service ontologies such as OWL-S are essential but not sufficient to exploit the full potential and the claimed benefits of the SW. Along with the technical means (e.g., programming tools such as J2EE and .NET, or description, discovery and messaging technologies such as WSDL, UDDI and SOAP), it is necessary to have a complete integration between knowledge bases (in the form of ontological domain models) and functional offerings (in the form of SWS). In other words, WS must be described in relation to the classes and individuals modeled in shared and commonly agreed web ontologies. Currently the Web Ontology Language (OWL) and its predecessor DAML are the languages in which most ontological models are represented.

There exists a need for developing various SWS applications by the IS research and development community. These applications could represent pilot projects and research prototypes to demonstrate how SWS technologies can strengthen and add benefits to the current Web-based IS and applications.

For the development of complex application-based WS integration scenarios, individual WS must appropriately understand the information that is being shared based on their intended meaning. To achieve this purpose, WS must carry a variety of metadata, such as descriptions of the interfaces of a service – the kinds of data entities expected and the names of the operations supported – format of request and response messages, order of interactions, and so on. The use of WS-MetadataExchange would be essential in this context. WS-MetadataExchange is a specification that discovers and retrieves WS metadata from a specific Internet address. For a given URI, WS-MetadataExchange defines how to query the network endpoint for WSDL definitions and associated policy information. WS-MetadataExchange is a complete messaging protocol to be carried out independently of and prior to any requestor-provider interaction.

6. Conclusion

The Web is the biggest repository of information ever created, but it is difficult for the computer systems to make sense of this information because they do not understand the meaning of the content. To build the next generation of the Web, there is a need to add more semantics to content and interfaces on the Web. To capture this XML-based semantics, it is important to capture all aspects of meaning creations, such as process modeling, ontology modeling, standard ways of defining meaning of languages, translation between different kinds of notations for people and applications with different preferences and needs, reasoning about Web resources and so forth. The enabling Semantic Web technologies provide the potential to play a critical role in various application areas, such as e-business, e-learning, e-government, and many more.

Web services technology, on the other hand is considered to be a computer-to-computer use of the Internet. One computer program or Web service makes a request of another Web service to perform its task or set of tasks and pass back the answer. Web services offer a completely new IT architecture, one based on the Web and the open standards. Rather than build proprietary systems, companies can obtain the functionality they need from the Internet. Thus some Web services can be proprietary, some public; some can be subscription based and others on demand.

Semantic Web services, the union of Semantic Web and Web services, expand the capabilities of a Web service by associating a semantic description of the Web service in order to enable automatic search,
discovery, selection, composition, and integration. Semantic Web services technology is gaining momentum. However, new Web engineering techniques should be defined to enable the systematic development and promote widespread adoption of Semantic Web services.

7. References


