The Technological and the Business Dimensions of the Topic of SOA and Semantic Web Technology

Kim, Tae-hwan*

ABSTRACT

The growing popularity of the Semantic Web and SOA has opened up exciting opportunities not only for the businesses but also for the web developers looking for best results in the e-commerce environment. This research is to investigate both the technological and the business dimensions of the topic of SOA and Semantic Web Technology. The fundamental technologies of each were discussed first and the benefits, advantages, limitations from a business perspective were illustrated.

Key Words : Semantic Web, SOA, E-commerce

Ⅰ. Introduction

Over the past several years, the subject of the Semantic Web and Service Oriented Architecture (SOA) has received a great deal of attention among practitioners as well as academics. Few topics have received as much attention as these subjects in applying Web technology for business application. The growing popularity of the Semantic Web and SOA has opened up exciting opportunities not only for the businesses but also for the web developers looking for best results in the e-commerce environment.

Originally, they designed the Web with the purpose that it should be useful not only for human to human interaction, but also those machines would be able to

* Department of Management Information Systems Dankook University, thkim@dankook.ac.kr

participate and help. They thought that the major obstacles to accomplish this had been the fact that most information on the Web is designed for human consumption. Leaving aside the artificial intelligence problem of training machines to behave like people, the Semantic Web develops languages for expressing information in a machine processable form (Daconta, et al., 2003).

Other concerns noted in the recent IT studies were SOA. The term expresses a software architecture that defines the use of loosely coupled software services to support the requirements of the business processes and software users (Channabasavaiah, et al., 2003). In an SOA, resources on a network are independent services that can be accessed without knowledge of their underlying platform implementation.

The primary purpose of this research project is to investigate both the technological and the business dimensions of the topic of SOA and Semantic Web Technology. First, the fundamental technologies of each will be discussed. Second, the benefits, advantages, limitations from a business perspective will be illustrated.

II. Overview of the Semantic Web and SOA

Semantic Web technologies help separate meanings from data, document content, or application code, using technologies based on open standards. 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 (Antoniou, et al., 2004). The Semantic Web also provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries. In other words, we can think of the Semantic Web as an efficient way to represent data on the World Wide Web, or as a database that is globally linked, in a manner understandable by machines, to the content of documents on the Web. Semantic technologies represent meaning using ontologies and provide reasoning through the relationships, rules, logic, and conditions represented in those ontologies

(Davies, et al., 2003).

In another hand, SOA provides link computational resources on demand to achieve the desired results for consumers' needs. The Organization for the Advancement of Structured Information Standards (OASIS) defines SOA as the following:

> "A paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations."

Enterprise architects believe that SOA can help businesses respond more quickly and cost-effectively to the changing market conditions (Computerworld, 2006). SOA can also simplify interconnection to and usage of existing IT assets.

In other respects, SOA is considered an evolution in architecture, not a revolution. It captures many of best practices or actual use of the architectures that came before it. In communications systems, for example, there has been little development in recent years of solutions that use truly static bindings to talk to other equipment in the network, but by formally embracing a SOA approach, solutions are better positioned to stress the importance of well-defined, highly interoperable interfaces (Krill, 2006).

One area where SOA has been recognized is as a tool for defining business services and operating models. The purpose of using SOA as a business mapping tool is to ensure that the services created properly represent the business view and are not just what technologists think the business services should be (Natis, et al., 2006). SOA is the process of defining architectures for the use of information in support of the business, and the plan for implementing those architectures.

Ⅲ. Semantic Web and SOA Infrastructure

Here we investigate and provide a description of the state of the SOA and Semantic

Web infrastructure globally based on Asia, Europe, and North America.

1. Asia

Public Sector has been one of the leading industries driving the adoption of SOA in Asia, according to Springboard Research. Government IT professionals are using SOA to reduce departmental silos and unify public sector infrastructures. The analysts said government interest in SOA is particularly noticeable in Australia, Hong Kong and Singapore. In addition, many government organizations in China and India have also deployed SOA or are in the process of pursuing SOA projects.

Springboard Research's end-user survey results and vendor interviews clearly identified the public sector as a key segment for SOA products and solutions.

> "There is a growing realization among Asian governments that SOA can help to serve citizens better and become more efficient in the delivery of services," said Dane Anderson, Vice President of Research for Springboard Research. "SOA helps governments combat their ongoing struggle of uniting technology platforms across hundreds of departments, and enables them to share information and offer more integrated services across whole of government initiatives," added Mr. Anderson.

Besides, government organizations are also trying to promote SOA by encouraging others to adopt it. Additionally, some of the governments in the region are also looking at SOA to strengthen and retain their competitive edge. For instance, both Hong Kong and Singapore look at SOA as an engine of economic growth. In fact, both of these governments have been taking an active interest in promoting SOA through public-private partnerships.

2. Europe and North America

According to pulse surveys conducted by Capgemini, a global leader in consulting, technology and outsourcing, organizations will, on average, increase the percentage of applications run on SOA by 20% over the next three years. The top reasons for utilizing SOA include innovation (30%), compliance (28%) and the speed of change (26%). The surveys conclude North American organizations expect to increase their adoption of SOA at a faster pace than European organizations. While the majority (67%) of North American respondents suggest they will push for their organizations to run between 20% and 60% of their applications on SOA-based technology in three years, the same percentage (67%) of European respondents look to run only up to 40% of their applications on SOA-based technology.

Ⅳ. Technical Discussion of Semantic Web and Service Oriented Architecture

To properly manage a SOA, enterprise must maintain active representations of the services available to the enterprise. Specifically, to discover and organize their services, the architects must use best practices that model and assemble services using metadata, encapsulate business logic in metadata for dynamic binding, and manage with metadata. Ontologies provide a very powerful and flexible way to aggregate, visualize, and normalize this service metadata layer (Willy & Sons, 2003).

Ontology is a network of concepts, relationships, and constraints that provide context for data and information as well as processes. Ontologies enhance service discovery, modeling, assembly, mediation, and semantic interoperability. They improve the way people browse, explore, and interact with complex metadata information spaces.

Business ontology is a formal specification of business concepts and their interrelationships that facilitates machine reasoning and inference. Business ontology ties systems together using metadata, much as a database ties together discrete pieces of data. This abstraction provides agility and flexibility, as interfaces can be changed and new resources and subscribers added easily, even while the system is running.

Semantics are the future of service-oriented integration. Semantic technologies provide an abstraction layer above existing IT technologies, one that enables the bridging and interconnection of data, content, and processes across business and IT silos. Finally, from the human interaction perspective, semantic technologies add a new level of semantic portals that provide far more intelligent, relevant, and contextually aware interactions than those available with the traditional point-to-point integration approach for portal-based information delivery.

V. Semantic Web Architecture

Humans are capable of using the Web to carry out tasks such as finding the Japanese word for “car,” to reserve a library book. However, a computer cannot accomplish the same tasks without human direction because web pages are designed to be read by people, not machines. The semantic web is a vision of web pages that are understandable by computers, so that they can search websites and perform actions in a standardized way. A computer could, for example, automatically find the nearest manicurist or book an appointment that fits a person’s schedule.

Currently, the World Wide Web is based primarily on documents written in Hypertext Markup Language (HTML), a markup convention that is used for coding a body of text interspersed with multimedia objects such as images and interactive forms.

HTML, as it is generally deployed, has limited ability to classify the blocks of text on a page. For example, with HTML, one can create and present a page that lists items for sale. The HTML of this catalog page can make simple, document-level assertions such as “this document’s title is ‘Superstore’”. But there is no capability within the HTML itself to assert that item number A569172 is an flagnance with a retail price of $199, or that it is a consumer product. Rather, HTML can only say that the span of text “A569172” is something that should be positioned near “flagnance” and “$199,” etc. There is no way to say that “flagnance” is a kind of title or that “$199” is a price. There is also no way to express that these pieces of information are bound together in describing a discrete item, distinct from other items perhaps listed on the page.

The Semantic Web make up this shortcoming, using the descriptive technologies Resource Description Framework (RDF) and Web Ontology Language (OWL), and the data-centric, customizable Extensible Markup Language (XML). These technologies are combined in order to provide descriptions that supplement or replace the content of Web documents. Thus, content may manifest as descriptive data stored in Web-accessible databases, or as markup within documents interspersed with XML or, more often, purely in XML, with layout/rendering cues stored separately. The machine-readable descriptions enable content managers to add meaning to the content, thereby facilitating automated information gathering and research by computers.

VI. Service Oriented Architecture

Service Oriented Architecture (SOA) is a design for linking computational resources on demand to achieve the desired results for service consumers. The Organization for the Advancement of Structured Information Standards (OASIS) defines SOA as the following:

> "A paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations."

Enterprise architects believe that SOA can help businesses respond more quickly and cost-effectively to the changing market conditions (Balzer, 2004). This style of architecture promotes reuse at the macro level rather than micro levels. It can also simplify interconnection to and usage of existing IT assets.

Ⅶ. Business value of Semantic Web and SOA

IT systems organize meanings using relational data models, flat files, object-oriented models, or proprietary data models. The demands of changing business needs require that new entities and relationships to the relational data models or object-oriented models.

Moreover, if an organization uses many applications provided by various vendors, we might replicate the same model across application databases. For instance, a banking firm can provides a variety of products to serve various types of customers. Normally, several vendors provide the applications for the bank, but each application replicates the same common information in an application-specific database. As the organization adds products to meet ever-growing business needs, the same redundant information becomes scattered around the enterprise.

If the bank adapts an ontology-driven approach, it can capture and represent its total product knowledge in a language-neutral form and deploy the knowledge in a central repository. With this shared, adapted ontology, the organization can provide a single, unified view of data across its applications. This unified view allows for precise retrieval of information and seamless enterprise integration, as business processes and various data sources can map to each other through a common meta-model. Thus, the shared ontology eliminates point-to point integration and simplifies application integration, reducing data redundancy and providing the same semantic meaning across applications, which eases the bank's maintenance and upgrades.

1. Benefits of the Semantic Web to the World Wide Web

The World Wide Web is the biggest repository of information ever created, with growing contents in various languages and fields of knowledge. But, in the long run, it is extremely difficult to make sense of this content. Search engines might help you find content containing specific words, but that content might not be exactly what you want.

The Semantic Web can provide the ability to tag all content on the Web, describe what each piece of information is about and give semantic meaning to the content item (Antoniou, 2004). Thus, search engines become more effective than they are now, and users can find the precise information they are hunting. Organizations that provide various services can tag those services with meaning; using Web-based software agents, we can dynamically find these services on the fly and use them to our benefit or in collaboration with other services.

2. The Benefits of a Service-Oriented Architecture

Service-oriented architecture has services that developers create in a service layer. The services that they develop have published interfaces (Krill, 2006). These interfaces support a distinct business domain. Organizations that focus their development effort around the creation of services, will realize many benefits.

The most common scenario for development organizations is to have some experience with component-based development. The use of application servers such as .NET for hosting applications is becoming more common. If our organization is using component-based development practices and application servers for business logic, then we are already service-oriented. By following the service-oriented mind set with even more rigor, combined with the component-based approach to software development, our organization will realize many benefits.

The creation of a service layer has the benefit of a better return on the investment made in the creation of the software. Services map to distinct business domains. For example, a company might create an inventory service that has all of the methods necessary to manage the inventory for the company. By putting the logic into a separate layer, the layer can exist well beyond the lifetime of any system it is composed into. For example, if our organization needs to create a credit card authorization service, there are basically two options. Developers will either develop the functionality as part of the application that needs it, or they will develop it as a separate component. If credit card authorization is developed as a separate component and used as a service, then it is likely to outlive the original application.

A service-oriented architecture will force an application to have multiple layers

(Pollock, et al., 2004). Each layer has a set of specific roles for developers. For instance, the service layer needs developers that have experience in data formats, business logic, persistence mechanisms, transaction control, etc. A client developer needs to know technologies such as SWING, JSP or MFC. Each layer requires a complex set of skills. To the extent that developers can specialize, they will excel at their craft in a particular layer of the application. Companies will also benefit by not having to rely on highly experienced generalists for application development. They may use less experienced developers for focused development efforts.

The creation of a service layer by definition means that developers have created an additional network interface that can be used by multiple applications (Mashups, 2005). When systems were built using monolithic or client-server methods, security was normally handled on the front-end. Companies often did not even implement database security because it is too hard to maintain multiple security lists. Services on the other hand are used by multiple applications, so they have their own security mechanisms. An application will therefore have multi-level authentication at both the client level and at the service level.

The services that developers create will evolve into a catalog of reusable services. Customers will come to understand this catalog as a set of reusable assets that can be combined in ways not conceived by their originators. Everyone will benefit from new applications being developed more quickly as a result of this catalog of reusable services.

Code reuse has been the most talked about form of reuse over the last four decades of software development (Hunt, 2002). Unfortunately, it is hard to achieve due to language and platform incompatibles. Component or service reuse is much easier to achieve. Run-time service reuse is as easy as finding a service in the directory, and binding to it. The developer does not have to worry about compiler versions, platforms, and other incompatibilities that make code reuse difficult.

One of the requirements of a service-oriented architecture is location transparency. This feature promotes scalability since a load-balancer may forward requests to multiple service instances without the knowledge of the service client. Also because of location transparency, multiple servers may have multiple instances of a service running on them. If a network segment or a machine goes down, a dispatcher can redirect requests to another service without the client's knowledge.

A service layer takes little extra effort in the beginning of a project, but the benefits pay off in the long run. From higher quality to more reuse to better scalability and availability, the benefits of implementing a service layer far outweigh the cost and extra effort involved.

Ⅷ. Conclusions

First, one obvious and common challenge faced is managing services metadata SOA-based environments can include many services which exchange messages to perform tasks. A single application may generate millions of messages. Managing and providing information on how services interact is a complicated task.

Another challenge is providing appropriate levels of security. Applications, particularly those external to company firewalls are more visible to external parties than traditional applications. The flexibility and reach of SOA can compromise security; the WS-Security suite of specifications is being developed to provide appropriate security.

Interoperability is another important aspect in the SOA implementations. The WS-I organization has developed Basic Profile (BP) and Basic Security Profile (BSP) to enforce compatibility. Testing tools have been designed by WS-I to help assess whether web services are conformant with WS-I profile guidelines. Additionally, another Charter has been established to work on the Reliable Secure Profile

There is significant vendor hype concerning SOA that can create expectations that may not be fulfilled. SOA does not guarantee reduced IT costs, improved systems agility or faster time to market. Successful SOA implementations may realize some or all of these benefits depending on the quality and relevance of the system architecture and design.

Another criticism of SOA is that the services are just stateless. According to www.xml.com, stateful service is difficult to avoid in a number of situations. One situation is to establish a session between a consumer and a provider. A session is typically established for efficiency reasons. For example, sending a security certificate with each request is a serious burden for both any consumer and provider. It is much quicker to replace the certificate with a token shared just between the consumer and provider. Another situation is to provide customized service.

Stateful services require both the consumer and the provider to share the same consumer-specific context, which is either included in or referenced by messages exchanged between the provider and the consumer. The drawback of this constraint is that it may reduce the overall scalability of the service provider because it may need to remember the shared context for each consumer. It also increases the coupling between a service provider and a consumer and makes switching service providers more difficult.

For the Semantic Web applications, critics question the basic feasibility of a complete or even partial fulfillment of them. Some approach the critique from the perspective of human behavior and personal preferences, which ostensibly diminish the likelihood of its fulfillment. Other commentators suggest there are limitations that stem from the current state of software engineering itself.

Enthusiasm about the Semantic Web could be tempered by concerns regarding censorship and privacy (Fensel, 2002). For instance, text-analyzing techniques can now be easily bypassed by using other words, metaphors for instance, or by using images in place of words. An advanced implementation of the Semantic Web would make it a lot easier for governments to control the viewing and creation of online information as this information would be much easier for an automated content-blocking machine to understand. In addition, the issue has also been raised that with the use of software files and meta-data, there would be very little anonymity associated with the authorship of articles on things such as a personal blog.

Another criticism of the Semantic Web is that it would be much more time-consuming to create and publish content as there would need to be two formats for one piece of data (Geroimenko, 2004). One format would need to be specialized for human viewing and the other would have to be specialized for machines. With this case, it would be much less likely for companies to adopt these practices as it would only slow down their processes. However, many web applications in development are addressing this issue by creating a machine-readable format upon the publishing of data or the request of a machine for such data. The development of micro formats has been one reaction to this kind of criticism.

Yet another criticism stems from the fact that the Semantic Web is based on traditional client-server architecture, which ultimately is not scalable (Walton, 2006). URIs within RDF is still machine-specific references, which calls the persistence of these documents into question. Many have pointed to MAYA's Information Commons as a more practical implementation of the basic RDF schema that is both distributed and ultimately scalable.

References

Antoniou, G., & Harmelen, F.V., 2004, *A Semantic Web Primer*. The MIT Press, April.

Balzer. Y., 2004, *Improve your SOA project plans*, IBM, July.

Channabasavaiah, Holley & Tuggle, 2003, Migrating to a service-oriented architecture, IBM Developer Works, Dec.

Hunt, A. & Thomas, D., 2002*, Software Archaeology, IEEE Software*, Vol. 19, No. 2, March/April , 20-22.

Daconta, M.C., Obrst, L.J., & Smith, K.T., 2003, *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management*. John Wiley & Sons, May.

Davies, J., Fensel, D., & Harmelen, F.V., 2003, *Towards the Semantic Web,* January.

Fensel, D., Wahlster, W., Lieberman, H., & Hendler, J., 2002, *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. MIT Press, November.

Fowler, M., 2002, Public versus Published Interfaces*, IEEE Software*, Vol. 19, No. 2, March/April, 18-19.

Geroimenko, V., & Chen, C., 2004, *Visualizing the Semantic Web*. Springer Verlag.

Hinchcliffe, D., 2005, *Is Web 2.0 The Global SOA?* SOA Web Services Journal, October.

Is There Real Business Value Behind the Hype of SOA? Computerworld, June.

Krill, P., 2006, *Make way for SOA 2.0*, InfoWorld, May.

Koch, C., 2005,A *New Blueprint For The Enterprise*, CIO Magazine, Mar.

Mashups, J.B., & SOBAs: *Which is the Tail and Which is the Dog?* Zapthink.

McKendrick, J., 2006, *Anti-SOA 2.0 petition nears 400*, ZDNet.com, June.

Natis, Y., & Schulte, Y., 2006, *Advanced SOA for Advanced Enterprise Projects*, Gartner, July.

Ontology-Driven Knowledge Management. John Wiley & Sons, 2003.

Pollock, J.T., & Hodgson, R., 2004, *Adaptive Information: Improving Business,* July.

Through Semantic Interoperability, Grid Computing, and Enterprise Integration.

Staab, S., & Studer, R., 2004, *Handbook on Ontologies*. Heidelberg: Springer Verlag. January.

The Technological and the Business Dimensions of the Topic of SOA and Semantic Web Technology

김 태 환*

▌요 약

나날이 관심을 받고 있는 어의 웹 (Semantic Web)과 서비스지향구조 (Service Oriented Architecture : SOA)는 업계뿐만 아니라 전자상거래 환경에서 좋은 결과를 얻기 위해 노력하는 웹 개발자들에게 새로운 기회를 창출하고 있다. 본 연구는 서비스지향구조와 어의 웹의 기술적 그리고 비즈니스 측면을 조사하고 있다. 먼저 기본적인 기술적 배경이 언급되었으며 비즈니스 측면에서 본 효용, 장점, 그리고 한계들이 기술되었다.

핵심주제어 : 어의 웹, 서비스지향구조, 전자상거래

* 단국대학교 상경대학 경영학부 부교수