Summary of Web Services:
Based on Various Articles and White Papers

Lindsay D. Grace
October 8, 2001

The article entitled *The Brave New World of Web Services* is a comprehensive introduction to the theory of Web Services implementation. The Web Services model provides a standard for creating functional business components that are accessible using standard Internet protocols. This model is part of the next generation of distributed computing, in which distribution of core operational logic occurs between computers linked only by the Internet.

The Web Services model is currently based on three core technologies. These technologies are the Simple Object Access Protocol (SOAP), the Web Description Languages (WSDL) and the Universal Descriptions, Discovery, and Integration (UDDI) mechanism. Each has a moderate relationship to the Extensible Mark Up Language commonly known as XML.

SOAP is the technology most closely related to XML. This technology "uses HTTP to carry messages that are formatted using XML"(Ben-Natan). HTTP, or Hypertext Transfer Protocol operates at the server level. It is one of the core technologies responsible for file sharing on the Internet. SOAP extends HTTP's abilities by providing a means of conveying logic. Instead of simply transferring data as HTTP does, SOAP affords the ability to transfer the logic that operates on data. Despite some analogies, SOAP and HTTP are not siblings. SOAP is a standard for object representation that may be transferred over HTTP.

WSDL also has roots in the XML recommendation. It "defines an XML grammar for describing network services as collections . . . capable of exchanging messages"(Ben-Natan). WSDL simply prescribes the way messages and protocol specific headers are described. The language used is XML. The methodology allows for unlimited message types, protocols, services, or other such data elements. The flexibility provided by this methodology and the language provides WSDL's required extensibility.

The final element of Web Services is UDDI. UDDI encourages interoperability by specifying a standard for systems to discover, describe, and integrate services. This means that systems may interact with each other to understand what other systems offer. UDDI is actually a hybrid of developed technologies. As the author reminds us, "the SOAP messages defined by UDDI allow Web Service providers to register themselves with UDDI registries" (Ben-Natan). XML once again provides the language of description. HTTP provides the distributed object management system required for UDDI registries.

The article also describes the fundamental layers of UDDI. These are the "businessEntity layer," "the businessService later," "the bindingTemplate layer," and the "tModel layer" (Ben-Natan). It does not benefit a 2-page description of the article to explain these layers in detail, but it is useful to know that they are organized into a hierarchy and that their contents are roughly reflected by their names.

Ben-Natan continues the article by describing a survey of related technologies. These include the WebSphere Application Server from IBM and the .Net initiative guided by Microsoft.

WebSphere is a cohesive production environment that supports all three core Web Service technologies. WebSphere includes tools first introduced by IBM. Some of this core technology for SOAP has been donated to the Apache organization.

The .Net initiative is a tremendous effort by Microsoft to push the world of development from web applications adapted to desktop and server environments (e.g. the current Distributed Computing model) to an entirely web based model. Microsoft hopes that the new .Net technologies (including C#, Visual Basic .Net, et al.) will have the same

effect that "the move from DOS to Windows" (Ben-Natan) had on the industry. Microsoft's technology reflects a slight derivation from the standard Web Services model. .Net uses SOAP and WSDL, but introduces the Microsoft construct called, DISCO. The author explains," DISCO is a simple discovery format in XML that allows users . . . to automatically discover Web Services"(Ben-Natan). Microsoft's DISCO is a response to what they consider the "overly complex" UDDI standard.

Web Services present an interesting organizational anomaly in the competitive world of networking. This set of technologies is not being separately developed by warring software companies, but is being forged by a cohesive group of competitors. Unlike the battle between Web Services predecessors CORBA (Common Object Request Broker Architecture marketed by OMG) and DCOM (Distributed Component Object Model marketed by Microsoft) the Web Services model is being developed from "both the Microsoft camp and the non-Microsoft camp." IBM, Microsoft, and others are working to develop this standard. By sharing resources between these companies, they can develop with the authority of ISO and the working speed of business. This should translate into the efficient development of one standard for the industry. It may also prevent the industry schisms produced by technologies that offer the same benefits, but different approaches, exemplified by CORBA and DCOM or UNIX and Windows NT.

Beyond the political example, Web Services is in itself a fascinating entrance into a new realm of distributed computing. It encourages the goals of some software Vendors that suggest the industry should move from PC centered computing to a more centralized model where end users use dumb terminals in the form of web appliances, and servers to do the actual computing. The proponents of this architecture suggest that a more centralized model will offer some core benefits. First, customers can receive the benefits of upgrades more easily, because upgrades will not need to be distributed to desktop computers. This should in turn lower the cost in end user computing. It may also bring end user costs down because dumb terminal web appliances should be simpler machines than desktops, and should be priced accordingly.

Web Services technology provides a greater opportunity to share core business logic between disparate systems. Since business processes are more than the data they generate or consume the ability to share these process should create a more tightly coupled network of systems that faithfully integrate the business logic of the participating companies. Web Services will simply allow developers to better network systems.

While this article clearly touts the enormous planning and benefits of the standards being developed, it fails to mention any inherent flaws. One might ask why HTTP is the assumed groundwork for Internet component transfers. Why would a new component transfer protocol not be part of this change? Does it make sense to adapt a file transfer protocol for component needs? The author hints at the attempt of flexibility by stating that "SOAP ... doesn't care about operating systems, programming languages, and other such issues." It will continue, however, to require HTTP.

This article also fails to discuss security issues involved with such implementations. There are of course the typical problems resolved by firewalls, encryption, and other tricks pulled from a system administrator's magic hat. More interestingly, there are the economic and sociological issues of security.

Business are not typically interested in exposing their business models to the general public. Nor are they interested in detailing the processes that they charge services for using. A quantitative money manager, for example, would sacrifice its competitive edge by exposing the factors or ranking model it uses to choose its stocks.

Today's businesses offer their process in compiled components that cannot easily be disassembled to reveal the reveal the models that went into forming them. However, with the transparency of XML, the dissection of these Intellectual properties may be as trivial as taking apart a child's toy. It then seems that the logical components that business will provide to the public, or through loose partnerships, are simple and non-proprietary. On the other hand, perhaps, these new technologies will produce new economies where the *service* provided by each module will incur a cost per transaction.

On the sociological front, there is little consideration for the effects of introducing such technology into society. With many new technologies there seems to be an initial excitement that clouds both judgment and caution. Like children in a candy store, developers buy into the technology, because it is available, and do not evaluate the consequence. Could such technologies be used in unscrupulous ways? Will Web Services provide people access to advanced technologies whose access to technology is limited by law? It seems logical that powerful networking technology may encourage powerful networking for both beneficial and adversarial uses. In some ways, the current Internet is experiencing the same dilemma.

Although often the theme of science fiction writing, there is a danger that technology may precede society's maturity. Hackers and malicious users have schooled the web community on the loopholes and inefficiencies that were created in a rush to acquire new technologies. Will we ask for these lessons again?

The article is not intended as a complete topography of the newly arising world of Web Services. It serves as a simple introduction. This introduction more or less educates its readers. While it is useful to discuss the topic as it pertains to the larger world around it, it would be difficult to do so in such a limited space. Perhaps the benefit of this level of introduction is its propensity to encourage new lines of thought. Whether Web Services become as ubiquitous as web browsers, the advent of such technology is worth some investigation.

Works Cited

Ben-Natan, Ron. "The Brave New World of Web Services: SOAP,WSDL, and UDDI part 1."

XML Journal (Sept 2001):

< www.sys-con.com/xml/article.cfm?id=248

10/8/01