THE QUEST FOR THE WEB SERVICES STACK

Flávio O. Silva
União Educacional Minas Gerais, Uberlândia, MG, Brasil

Pedro F. Rosa
Faculdade de Computação, Universidade Federal de Uberlândia, Uberlândia, MG, Brasil

Keywords: Web Services, Protocols, Standards, Specifications, Layers.

Abstract: The strong use of distributed applications is due to the fact that Internet and the Web are based on common and widely accepted standards. Web services can be the basis of the next generation of applications. Unfortunately if someone asks for the layers of the Web Services protocol stack, the answer cannot be given right away. Considering that this technology is emerging and its bases are being defined, different companies and working groups propose standards that come and go with the same speed. Another problem is that these groups provide their own vision about the stack and, finally, the widely available standards proposals makes this environment a standard's Babel. In this paper we will run a quest in order to find the Web Services protocol stack, by putting together different standards and visions and finally providing an independent Web Services protocol stack.

1 INTRODUCTION

Internet and, particularly, the Web have generated deep transformations in all knowledge fields. The impact provided by the Web is changing the way applications are being constructed. The strong use of distributed applications is due the fact that the Web is based on common and widely accepted protocols like Hyper Text Transfer Protocol (HTTP) (Tanenbaum, 2003).

Right now the Web is being prepared to switch the way users interact with it. The user-to-machine relationship will be extended by a machine-to-machine relationship. This is the role of Web Services.

To accomplish this, Web Services applications might be based on common and widely accepted standards, as well.

Unfortunately if one asks what are the layers of the Web Services protocol stack this answer cannot be given right away. Although this can be accepted in an emerging technology, that is under development, this can be a great risk the for the Web Services approach.

Right now different companies and working groups are proposing protocol specifications and standards. These protocols reflect their vision about Web Services protocol stack. In this environment the protocol proposals come and go with the same speed. The applications that are being created right now may lack of compatibility in the future because of a selected protocol suite.

Another problem is that software architects, engineers, developers may be victims of this standard's Babel.

In this paper we will run a quest in order to find the Web Services protocol stack.

This will be accomplished first by analyzing the multiple visions of this stack provided by different groups and the myriad of protocols and standards available; this will be done on section 2.

Section 3 will propose the Web Services protocol stack and its layers. Finally in section 4 some concluding remarks.

O. Silva F. and F. Rosa P. (2006).
THE QUEST FOR THE WEB SERVICES STACK.
In *Proceedings of the International Conference on e-Business*, pages 105-112
DOI: 10.5220/0001426501050112

2 THE STACK - SEARCHING FOR THE CLUES

To reduce their complexity, network software is organized as a stack of layers. Each layer offer services to the higher layers and use the services provided by the lower ones. The same concept must be applied to Web Services.

The search will begin at World Wide Web Consortium (W3C), considering that this standard organization is responsible by SOAP (Mitra, 2003) protocol, one of the main Web Services standards, adopted de facto, as the message format.

This organization created a working group (W3C, 2004) in order to define Web Services Architecture. Although the final document contains a conceptual view of the Web Services stack, it did not achieve its objectives, as noted by Steve Vinoski (Vinoski, 2004) a charter member of this group.

Another place to look for the Web Services architecture is the work done by the Organization for the Advancement of Structured Information Standards (OASIS). At February, 2005 a technical committee (TC), called OASIS SOA Reference Model TC (OASIS, 2005). This OASIS TC has a different view about the Web Services protocol stack. They claim that the correct is to create a reference model which will be used as a guideline to create specific architectures. A working draft (MacKenzie, 2005) was created, but, at this time no implementations of an architecture using this reference model were proposed.

Some books about the subject could be another interesting place for this quest. Most of them, like (McGovern, 2003) are concerned about the basic and well stated Web Services protocols like SOAP, Web Services Description Language (WSDL) (Chinnici, 2006) and Universal Description Discovery & Integration (UDDI) (Clément, 2004)0 and does not mention the new standards available. An interesting book from Sanjiva Weerawarana (Weerawarana, 2005) and others go deeper in order to propose the Web Services stack and its protocols, but it does not mention other protocols and standards available from other development groups.

Considering that companies like IBM, Microsoft, Sun, Oracle and others are leading the standards proposals a good source of information for this quest is look for their vision.

Microsoft's vision of the Web Services stack (Microsoft, 2006) contains information about well established and new standards. The stack considered in this document does not show how a specification, like BPEL4WS (Weerawarana, 2005), fits on it, although this standard is cited in the document.

The same happens to management related specifications, like WS-Management (Arora, 2005). They are very new specifications, but the Web Services stack does not show their layer.

Compared to Microsoft's stack, IBM view contains additional layers (IBM, 2006)0. Most of the protocols are similar, considering that both companies have a work together proposing Web Services specification. But there are differences. The security layer from IBM contains the standard Security Assertion Markup Language (SAML) (Cantor, 2004), while it is not present at Microsoft's security layer.

This clearly indicates that there is no consensus in some areas and this may lead to incompatibilities between applications. In this case the Web Services interoperability can turn in a far promise.

Continuing with the quest for the stack, an important reference is Sun. The vision (SUN, 2006) is very different when compared with IBM and Microsoft. An important observation is that security specifications and standards are not mentioned. It means that someone that is searching for the Web Service protocol layers will not have a complete vision of it.

All this information and different points of view produce new visions, and some researchers are proposing custom Web Services architectures, which are related to some specific domains like Semantic Web (Turner, 2003), creating vertical Web Services protocol stacks.

Besides different views of the stack, protocol specifications come and go with the same speed. Considering this, Savas Parastatidis (Parastatidis, 2004) proposed a method to evaluate the risks assessments for Web Services protocols specifications.

3 WEB SERVICES STACK – THE HOLY GRAIL

Section 2 shows that there is no consensus about the Web Services protocol Stack. In order to propose a protocol suite, all the actually available views were considered and more than fifty protocols at hand until this moment where analyzed.

Figure 1 shows the layers present in the Web Services protocol stack:

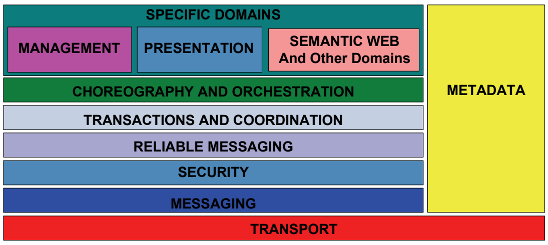

Figure 1: Web Services Protocol Stack.

The model shown at Figure 1 considers the main aspects related to Web Services taking into account their current development state and the proposed protocols right now.

Some layers may not be present in some Web Services Architecture, it will depend on the application field and the assumptions made. One may say that security is not a problem into an intranet, where only basic services are needed, or maybe that he is relying in the Secure Hypertext Transfer Protocol (HTTPS).

The layers presented at Figure 1 will be analyzed in greater depth. A red dashed line shows redundant standards which offer a similar solution to the same problem with small differences. There are cases that another standard superseded a standard and in this case, it will be on a rectangle inside the newer standard.

The standards present on each layer are related to the domain of each layer. Some of them are recent and are in different stages of development. Some, like SOAP (Mitra, 2003), are standards accredited by a standard organization like the W3C or OASIS

Other standards, like Web Services Choreography Description Language (WSCI) (Kavantzas, 2005), were submitted to these organizations and are in a standardization process.

Another group has been just published by a company or a development group but was not submitted to any standard organization, within this group there are proposals like SOAP-Over-UDP (Gudgin, 2004).

Finally, some standards have been not published yet, but are scheduled or planned to be. The protocols in this category are based on vision of a need for Web Services; an example is WS-Authorization (Microsoft, 2002), which was present as this white paper and until now was not even published and was superseded before its publication.

3.1 Transport Layer

The transport layer is responsible for the message exchange between two endpoints. This layer is transport and application layers based protocols as defined in the Internet architecture, where the main protocol here is Hyper Text Transfer Protocol (HTTP). By the way, this is the transport protocol defined in the Basic Profile (Ballinger, 2006), published by WS-I.

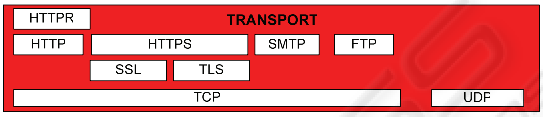

Figure 2: Transport Layer

The protocol HTTP specifies how the messages are exchanged between the client and a server. Although, this protocol HTTP is base for the Web, it cannot deal with situations like: lost messages, duplicated messages; long messages and message acknowledgement.

IBM proposed and specified a protocol called Hypertext Transfer Protocol Reliable (HTTPR) (Banks, 2002) that uses HTTP and allows a reliable message exchange between the client and the server, in order to provide reliable Web Services. This specification was published on 2002, but this specification will remain as another “proposed only” standard.

Although there are proposals this layer is well defined and protocols like HTTP, Simple Mail Transfer Protocol (SMTP) are the underlying protocols that are specified for the transport of SOAP messages.

Figure 2 shows the protocols defined for this layer.

3.2 Messaging Layer

This layer defines the formatting of the messages and the way it will be delivered, independently from the programming language, the message processors or the platform. SOAP protocol is the basis of this layer. This protocol is a standard and it is widely accepted as the core of Web Services. Above de SOAP protocol there are other specifications related to message addressing, message routing, message acknowledge, sequencing and other message exchange mechanism different from the basic request-response, like notification.

As can be seen in figure 3, in this layer there are competing standards created by different group of companies, represented by the red dashed line and there are some standards that were superseded by other even before being part of a standardization process, this is shown by the standards within a rectangle inside another rectangle which represents the main standard by this time.

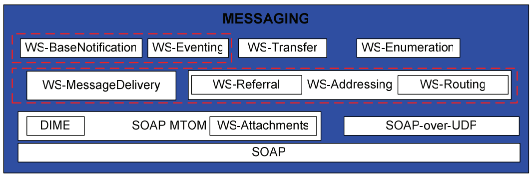

Figure 3: Messaging Layer.

Figure 3 shows the relation between all the protocols and how they are layered and are related to each other.
Although SOAP is well stabilized there are proposal like SOAP Message Transmission Optimization Mechanism (SOAP MTOM) (Gudgin, 2005), which is a W3C recommended standard, which basically defines a mechanism to optimize the transmission by coding some parts of the message in order to compact its data, using special character sequences that will be reconstructed in at the receiver side.

3.3 Security Layer

On loosely coupled systems, security is a very important issue that might considered. Web Services applications are related to key information that may be transported over Internet. On this environment is important to guarantee an end-to-end security, not a point-to-point security. The conversation between two applications endpoints using Web Services usually is performed between many different nodes and the security requirements must be present through all these nodes.

The security protocols available at the transport layer, like Secure Sockets Layer (SSL) (Tanenbaum, 2003) or Transport Layer Security (TLS) (Tanenbaum, 2003) are focused on a point-to-point communication

The security layers defines the protocols in order to guarantee properties like message integrity, it's confidentiality, it's non-repudiation and services like authentication and authorization.

WS-Security (Nadalin, 2004), a well defined and accepted OASIS standard, is the basis for this layer. It provides SOAP message security in order to guarantee its integrity and confidentiality and basically it uses XML-Signature (Eastlake, 2002) and XML-Encryption (Eastlake, 2002) to achieve its objectives.

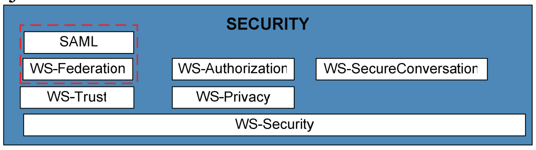

Figure 4: Security Layer.

Above WS-Security there are other standards that are concerned to other security issues like authentication, authorization. These requirements are related to policy; the exchange of security tokens between trusted domains and single sign-on. On this area the standards are growing are not in a mature state.

3.4 Reliable Messaging Layer

The use of SOAP over HTTP, as an example, does not guarantee that the messages will correctly delivered. This layer contains protocols in order to provide this requirement.

Reliable messaging is related to guarantee of deliver; absence of duplication and ordered deliver.

In order to provide these functionalities the protocols of this layer usually adds information to the message like: a message identification number in order to guarantee its uniqueness; a sequence number; a time to live value. Besides this they have to provide a method in order to provide positive or negative acknowledgment of the received message in order to confirm or deny the success of the delivery.

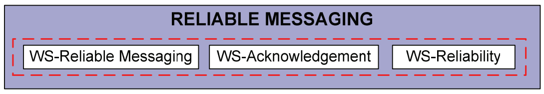

Figure 5: Reliable Messaging Layer

As can be seen in figure 5, there are other competing standards, but until this time WS-Reliability (Iwasa, 2004)0 is the only of them which is an accredited standard by OASIS. This standard specifies how reliable SOAP messages can be sent over HTTP.

3.5 Transactions and Coordination Layer

Some applications have in its requirements the need to be fully completed to consider it correct. The classical example is the money transfers between two accounts, two operations has to be completed successfully in order to guarantee that the transfer was correctly.

This sample shows the requirement of a transaction which represents a series of operations that has to be executed integrally and that are seen as unique operation by others systems. If some of these activities could not be completed the processing of the transaction might be canceled or another action has to be done. Another issue regarding this kind of the problem is that a transaction can have a short run or a long run time. The above example is a short run, a example of long run would be a product delivery where the customer will only be billed after a successful deliver of the products.

This layer implements these requirements. On a global view, this is done by creating a operation context that can be shared between many different Web Services. This context defines the conditions of operation of each Web Services and within this context it's possible to exchange information about the processing of each activity that has to be executed.

On this environment is necessary to have a coordination in order to change the state of this context and propagate this information to all Web Services that are working together to perform a common objective.

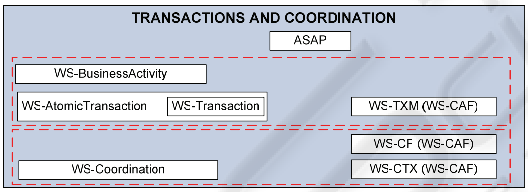

Figure 6: Transactions and Coordination Layer.

Figure 6 shows some protocols specifications in order to provide transaction modeling and coordination. Web Services Composite Application Framework (WS-CAF) (Litte, 2005), which is under a standardization process, provides the functionalities present in this layer.

In this area there are competing standards in different stages of development.

3.6 Choreography and Orchestration Layer

Loosely coupled endpoints uses Web Services to provide the communication and functionality for different applications by making this technology suitable for many different processes present at the e-business world.

These processes can contain many different requirements and some examples are: the existence of a sequence of operations, usually variable that might be performed; the necessity to handle transactions that might be executed in a synchronous or asynchronous way; the need of the monitoring these processes and others.

This layer has the high level protocols to perform these common e-business requirements.

Figure 7: Choreography and Orchestration Layer.

This layer can be divided in two sub layers. One devote to the orchestration and the other concerning with the choreography between different Web Services.

The concept of orchestration is related to a group of Web Services that interact in order to perform a business rule and in this group it's necessary to have a maestro which is responsible to conduct the process in order to achieve the desirable results.

Web Services Business Process Execution Language (WSBPEL) (Arkin, 2005) is about to be a standard from OASIS and represents a proposal for Web Services orchestration.

Choreography sub layer is over the orchestration sub layer. The concept of choreography is larger than orchestration and it is related with different parts which interact on a public way in order to perform a common task. This concept is near the P2P concept where each part has a vision of the whole context and executes its particular role. A part of this choreography can be executed, for example, by an orchestration.

A W3C working group is working on a standard called Web Services Choreography Description Language (WS-CDL) (Kavantzas, 2005), which specifies how to perform choreography between Web Services.

3.7 Specific Domains Layer

Web Services can be applied to offer computing as a distributed service to a wide range of application areas.

On the top layer is located the Specific Domains layer which contains protocols intended to be used in particular areas like: Healthcare (OASIS, 2004); e-Commerce (OASIS, 2001) e-Procurement (OASIS, 2003); Service Distributed Management (OASIS, 2003); User Interface (Kropp, 2003); Supply Chain (OASIS, 2003)0; e-Government (OASIS, 2002); Voice over XML (McGlashan, 2004) and Semantic Web (RuleML, 2006); among others.

This layer is correlated with the Application Layer of the ISO OSI model and for sure a new generation of exciting applications will arise from this layer.

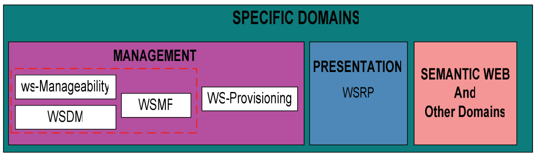

Figure 8: Specific Domains Layer.

The protocols present at this layer are not related with the Web Service technology itself but are concerned to IT applications and focus particular needs of such domains

The protocols and specifications defined for this layer is beyond the scope of this work.

3.8 Metadata Layer

The Metadata Layer can be used even as on execution mode as on a development mode. This layer is related to definition, discovery and policies for Web Services. The base of this layer is WSDL, which is a de facto standard for the description and definition of Web Services, as shown in figure 9.

This layer contains protocols like UDDI (Clément, 2004)0, which can be used for Web Services discovery as defined in the SOA architecture.

Protocols related to policies of using are present in this layer, such as WS-Policy (Bajaj, 2004), which express requirements that might be satisfied for the use of Web Services like authentication schemas; transport protocol that should be used; QoS indicators; security policies and others.

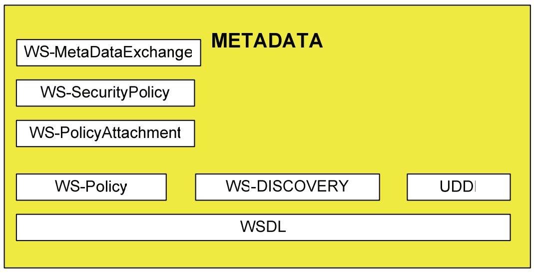

Figure 9: Metadata Layer.

4 CONCLUDING REMARKS

The concept of a protocol stack is an important abstraction. The actual development of computer networks is based on this concept and in the fact that these protocols specifications are accepted and implement by different software vendors.

One of the key benefits of Web Services is the interoperability between different applications constructed over well defined and accepted protocols.

This subject is new and is under development, in this context, the presence of different views about the Web Services protocols stack is natural and under certain conditions, healthy to turn it a mature and reliable architecture.

The benefits of Web Service cannot come true if these different views start to compete with each other instead of contributing.

The different Web Services protocol stack and specifications will turn on different implementations, it may result in a lack of compatibility and the soul of Web Services, which is about providing computing between endpoints over the Internet, will be just a tech dream.

Then, the protocol stack, a basic element, remains obscure and far from software architects, engineers and developers. The dark side of the competition is that usually, different groups, working on the same subject, and thus using their resources in different directions, by producing not the desired development, but confusion.

Besides this, within each layer, there are a bunch of Web Services protocols that provides the same services, usually with minor conceptual differences.

An important contribution is that although you can find a lot of clues about the layers, usually their just conceptual and does not show the protocols inside them and how they are related in order to provide the desirable results.

Even the groups or companies that provides specifications does not show how they can fit together, the impression is that each specification solve a particular problem.

Another point is that the companies and development groups only talk about the specifications and standards they proposed. Again, someone that starts looking for the Web Services standards and layers will become more confused than before.

Another important contribution is that in this work more than fifty Web Services specifications and standards were analyzed. Thus the protocol on each layer shows the superseded and competing protocols in a very straight full view, making easy to compare these specifications.

The Web Services stack presented and proposed is an independent view of the Web Services architecture and can contribute to merge different efforts performed by the research community.

Although there are different views, some standards are becoming de facto standards clearly at lower layers.

Comparing to ISO OSI model the same phenomenon happened: the protocols at the lower layers are well defined and accepted while the higher layers are basically a vision, and becomes optional in most situations, such as the session layer.

Considering that a new network technology is under development problems like this can be avoided, and consistent and well done protocol stacks can de constructed.

The Web Service protocol stack is a key component that will guide the development and the implementation of this technology.

This paper brings this important issue to the discussion and proposes an independent view of the Web Services stack, resulted from the comparison and reasoning about a myriad of protocols and personal protocol stacks actually available.

REFERENCES

Tanenbaum, Andrew S., 2003. *Computer Networks*, Fourth Edition. Prentice Hall, New Jersey

Mitra, Nilo, 2003. *SOAP Version 1.2 Part 0: Primer*. W3C Recommendation 24 June. Retrieved May 2, 2006, from <http://www.w3.org/TR/2003/REC-soap12-part0-20030624/>.

W3C, 2004. *Web Services Architecture Working Group*. Retrieved May 2, 2006, from <http://www.w3.org/2002/ws/arch/>.

Vinoski, Steve, 2004. *WS-Nonexistent Standards*. IEEE Internet Computing, vol. 8, no. 6, pp. 94-96. Retrieved May 2, 2006, from <http://dsonline.computer.org/0412/d/w6towp.htm>.

OASIS, 2005. *OASIS SOA Reference Model TC*. Retrieved May 2, 2006, from <http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm>

MacKenzie,C. MatthewR et al, 2005. *Reference Model for Service Oriented Architectures. Working Draft 11, 15 December 2005*. Retrieved May 2, 2006, from <http://www.oasis-open.org/committees/download.php/15966/wd-soa-rm-11.pdf>

McGovern, James et al., 2003. *Java Web Services Architecture*. Morgan Kaufmann, San Francisco.

Chinnici, Roberto et al., 2006. Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. W3C Candidate Recommendation 6 January 2006. Retrieved May 2, 2006, from <http://www.w3.org/TR/wsdl20/>.

Clément, Luc et al., 2004. *UDDI Version 3.0.2. UDDI Spec Technical Committee Draft*. Retrieved May 2, 2006, from <http://www.oasis-open.org/committees/uddi-spec/doc/spec/v3/uddi-v3.0.2-20041019.htm>.

Weerawarana, Sanjiva et al., 2005. Web Services Platform Architecture : SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More. Prentice Hall PTR

Microsoft, 2006. *Web Services Specifications*. Retrieved May 2, 2006, from <http://msdn.microsoft.com/webservices/webservices/understanding/specs/default.aspx?pull=>

Arora, Akhil et al., 2005. *Web Services for Management (WS-Management June 2005)*. Retrieved May 2, 2006, from <http://msdn.microsoft.com/ws/2005/08/ws-management/>

IBM, 2006. *Standards and Web services*. Retrieved May 2, 2006, from <http://www-128.ibm.com/developerworks/webservices/standards/>

Cantor, Scott et al., 2004. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0. Retrieved May 2, 2006, from <http://www.oasis-open.org/committees/download.php/9455/sstc-saml-core-2.0-cd-02.pdf>.

Sun, 2006. Web Services Standards and Technologies. Retrieved May 2, 2006, from <http://developers.sun.com/techtopics/webservices/standards.html>

Turner, Mark; Budgen, David; Brereton, Pearl, 2003. *Turning Software into a Service*. Computer, vol. 36, no. 10, pp. 38-44, October.

Parastatidis; Savas; Webber, Jim, 2004. *Assessing the Risk and Value of Adopting Emerging and Unstable Web Services Specifications*. SCC, pp. 65-72, Services Computing, 2004 IEEE International Conference on (SCC'04), 2004.

Kavantzas, Nickolas et al., 2005. *Web Services Choreography Description Language Version 1.0*. W3C Candidate Recommendation 9 November 2005. Retrieved May 2, 2006, from <http://www.w3.org/TR/ws-cdl-10/>

Gudgin,Martin et al., 2004 *SOAP-over-UDP*. Retrieved May 2, 2006, from <http://msdn.microsoft.com/webservices/default.aspx?pull=/library/en-us/dnglobspec/html/soap-over-udp.asp>

Microsoft; IBM, 2002. *Security in a Web Services World: A Proposed Architecture and Roadmap*. Retrieved May 2, 2006, from <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwssecur/html/securitywhitepaper.asp>.

Ballinger, Keith et al., 2006. *Basic Profile Version 1.1*. The Web Services-Interoperability Organization. Retrieved May 2, 2006, from <http://www.ws-i.org/Profiles/BasicProfile-1.1.html>

Banks, Andrew et al., 2002 *HTTPR Specification*. IBM Software Group. Retrieved May 2, 2006, from <http://www.ibm.com/developerworks/library/ws-httprspec/>.

Gudgin, Martin et al., 2005 *SOAP Message Transmission Optimization Mechanism. W3C Recommendation 25 January 2005*. Retrieved May 2, 2006, from <http://www.w3.org/TR/soap12-mtom/>

Nadalin, Anthony et al., 2004. *Web Services Security: SOAP Message Security 1.0 (WS-Security 2004). OASIS Standard 200401, March 2004*. Retrieved May 2, 2006, from <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf>

Eastlake, Donald et al., 2002. *XML Encryption Syntax and Processing. W3C Recommendation 10 December 2002*. Retrieved May 2, 2006, from <http://www.w3.org/TR/xmlenc-core/>

Eastlake, Donald et al., 2002. *XML-Signature Syntax and Processing. W3C Recommendation 12 February 2002*. Retrieved May 2, 2006, from <http://www.w3.org/TR/xmldsig-core/>

Iwasa, Kazunori et al., 2004. *WS-Reliability 1.1. OASIS Standard, 15 November 2004*.Retrieved May 2, 2006, from <http://docs.oasis-open.org/wsrm/ws-reliability/v1.1/wsrm-ws_reliability-1.1-spec-os.pdf>

Litte, Mark; Newcomer, Eric; Pavlik, Greg, 2005. Web Services Coordination 3 Framework Specification (WS-CF). Retrieved May 2, 2006, from <http://www.oasis-open.org/committees/download.php/15042/WS-CF.zip>

Arkin, Assaf et al., 2005. Web Services Business Process Execution Language Version 2.0. Committee Draft, 01 September 2005. Retrieved May 2, 2006, from <http://www.oasis-open.org/committees/documents.php?wg_abbrev=wsbpel>

Kavantzas, Nickolas et al., 2005. *Web Services Choreography Description Language Version 1.0. W3C Candidate Recommendation 9 November 2005*. Retrieved May 2, 2006, from <http://www.w3.org/TR/ws-cdl-10/>

OASIS, 2004. OASIS International Health Continuum (IHC) TC. Retrieved May 2, 2006, from <http://www.oasis-open.org/committees/ihc/charter.php>

OASIS, 2001. OASIS Universal Business Language (UBL) TC. Retrieved May 2, 2006, from <http://www.oasis-open.org/committees/ubl/charter.php>

OASIS, 2003. OASIS Electronic Procurement Standardization (EPS) TC. Retrieved May 2, 2006, from <http://www.oasis-open.org/committees/eps/charter.php>

OASIS, 2003. OASIS Web Services Distributed Management (WSDM) TC. Retrieved May 2, 2006, from <http://www.oasis-open.org/committees/wsdm/charter.php>

Kropp, Alan et al., 2003. Web Services for Remote Portlets Specification. Approved as an OASIS Standard August 2003. Retrieved May 2, 2006, from <http://www.oasis-open.org/committees/download.php/3343/oasis-200304-wsrp-specification-1.0.pdf>

OASIS, 2003. OASIS Product Life Cycle Support (PLCS) TC. Retrieved May 2, 2006, from <http://www.oasis-open.org/committees/plcs/charter.php>

OASIS, 2002. OASIS E-Government TC. Retrieved May 2, 2006, from <http://www.oasis-open.org/committees/egov/charter.php>

McGlashan, Scott et al., 2004. Voice Extensible Markup Language (VoiceXML) Version 2.0. W3C Recommendation 16 March 2004. Retrieved May 2, 2006, from <http://www.w3.org/TR/voicexml20/ >

RuleML, 2006. The Rule Markup Initiative. Retrieved May 2, 2006, from <http://www.ruleml.org/>

Bajaj, Siddharth et al., 2004 *Web Services Policy Framework (WSPolicy)*. Retrieved May 2, 2006, from <http://ftpna2.bea.com/pub/downloads/WS-Policy.pdf>