by HEATHER KREGER

FULFILLING THE WEB SERVICES PROMISE

12

Web services are no longer just hype—they have been sanctioned by the industry on two fronts: standards and products. IBM is investing time, talent, and money on both these fronts. Web services technologies are being developed as the foundation of a new generation of business-to-business (B2B) and enterprise application integration (EAI) architectures, as well as important parts of such "on demand" components as grid, wireless, and autonomic computing. In order to fulfill the promise of this foundation, a set of technologies must be specified, standardized, realized in language bindings, and supported by interoperable products.

This article reviews the set of technologies necessary for Web services, looks at where they are being specified and standardized, and identifies the current technical challenges the Web services community is attacking. It concludes with recommendations for the use of Web services today and suggestions that prepare for Web services tomorrow.

We can represent the technologies that must be standardized in order to implement the Web services-oriented architecture (see the sidebar by Ferris and Farrell) in the conceptual Web services stack [6] shown in the figure here. The three sections of the stack are corollaries to the roles in the Web services-oriented architecture: *interact, description,* and *discovery agency.*

At the base is the wire section that captures the technologies required to transport messages from a service requester over the network to the service provider. The *transport* layer addresses network connectivity via the ubiquitous TCP-IP base. The *packaging* layer defines how the payload is encoded in the message to be transported. The *extensions* layer defines the extensible set of features expressed as headers on the message. These layers must support the XML information set (infoset). SOAP [11]

The creation and support of standards for Web services is a critical component to their effective functionality and ultimate success.

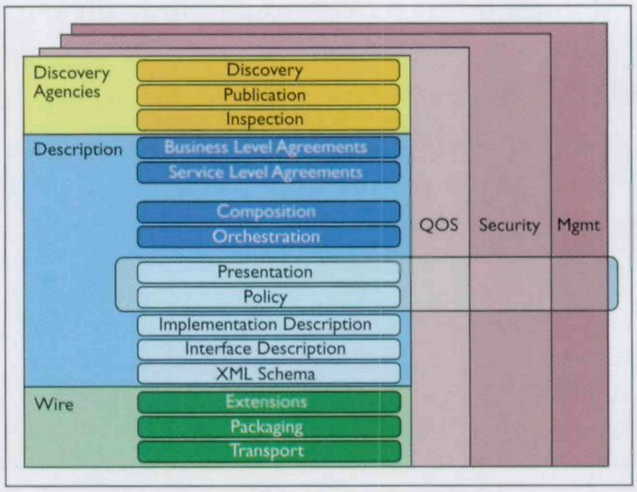
Web services technologies are being developed as the foundation of a new generation of B2B and EAI architectures, as well as important parts of such "on demand" components as wireless, and autonomic computing. In order to fulfill the promise a set of technologies must be specified, standardized, realized in language bindings, and supported by interoperable products.

and HTTP are the most widely supported standards for these layers, but other bindings are possible. The technology choice for this section will determine the potential client base for a service.

The next layer to be standardized is the description layer. All type descriptions are specified and expressed using the XML Schema language. The interface and implementation description define the mechanics of interacting with a Web service, which includes the operations and messages supported, how to serialize those messages onto the wire, and where to send the messages. A *policy description* layer will be used to describe service-specific information beyond mechanics, such as owning business, taxonomy, security requirements, timeouts, costs, and quality of service parameters. The *presentation* layer describes how a user interface is generated for this service. These four layers fully describe a service. The next two layers describe relationships and interactions between services. Related services may be expressed in the *composition* layer. This layer includes groupings, containment, dependencies, and parent-child relationships. The *orchestration* layer encompasses ordering of operations, choreography, workflows, and business processes. The final two layers describe agreements between the service requestor and provider. The *service-level agreement* layer defines the specific performance, usage, costs, metrics, and thresholds to which a service is expected to adhere. The *business-level agreement* layer describes a contractual agreement between the two business partners who will be transacting business using Web services.

Discovery agencies, the next section of the stack, encompass the technologies that enable service descriptions to be published, support the discovery of service descriptions, and provide inspection of sites for the descriptions of hosted services. Publish is very loosely defined as any means to make a service description available to a requester—from email to registries. Discovery is defined just as loosely, ranging from accessing a description in a file system to sophisticated searches of service registries at either development or runtime.

Web services conceptual stack.



The pages behind the stack represent overarching infrastructure concerns that span every layer of the stack. They are *security, management,* and *quality of service.* The solutions for these concerns may be different for each layer of the stack. Security is the most pressing, but also the most mature of these concerns.

The technologies for these layers and towers combine to create a complete infrastructure for use by B2B applications, EAI solutions, as well as the emerging grid infrastructure. This broad use of Web services requires the technologies of the conceptual stack and their language bindings be standardized, and that development tools and middleware support them in a verifiably interoperable manner.

Standardization

Given the figure here, how far along the road are we toward standardization of that stack? Standardization actually occurs in four stages: specification, submission, working group, and approval.

Architecture. The overall terminology and architecture, including interaction models and the stack for Web services, are being specified by the W3C Web Services Architecture Working Group. The reference architecture [6] and glossary developed by this

Web services in it current state of support is an excellent application integration technology. Web services can glue together applications running on two different messaging product platforms, enable database information from one application to be made available to others, and enable internal applications to be made available over the Internet.

group is targeted for availability this year. However, since all of the work of this group, including interim drafts, is being conducted in public, the definitions and supporting graphics are already available.

The Wire Stack. For the network and messaging layers, we use the TCP-IP, HTTP, and the SOAP v1.2 specification [11] from the W3C. These protocols are widely supported by development and middleware products and have interoperable runtimes available for many languages and platforms that include Microsoft's .Net (www.microsoft.com/net/) platform as well as the Java technology platforms that use JAX-RPC (JSR101) and Web services for J2EE (JSR109) as defined by the Java Community Process. Indeed, SOAP has evolved into an excellent technology for B2B applications. Additionally, bindings to existing messaging technologies, like message queues and message-oriented middleware, have been developed for use within the enterprise for improved reliability and performance while still preserving the loose coupling and interoperability offered by XML and SOAP.

The Description Stack. For the description section, the Web Services Description Language (WSDL) [7] provides the standard interface (portType) and implementation (binding) description, along with the addressability (location) for a service. The W3C Web Services Description Working Group is developing the next version of the WSDL specification that should be completed this year. WSDL is already supported by multiple programming languages (such as Java and C), as well as by most development tool and middleware vendors. As with any other Web standard, they will have to track the WSDL specification as it finalizes. The policy layer should be satisfied by the WS-Policy specifications [3], but this must still be standardized. The user interface layer is being specified jointly by the OASIS Web Services for Remote Portlets (WSRP) and Web Service for Interface Applications (WSIA) technical committees. The Business Process Execution Language For Web Services (BPEL4WS) [8], WS-Coordination [5], and WS-Transactions [4] specifications together form a proposal for both the orchestration and composition description layers. These specifications are at the beginning of the standardization process. A preview implementation, BPWS4J, is available at IBM's Alphaworks site (www.alphaworks.ibm.com). The Web Service Level Agreement specification [9] details a service-level agreement description, but no standardization process has begun. The contract description is just beginning to get some attention and prototype development, while there were specifications in ebXML on this, there are no specifications or standards in this area yet specifically for Web services.

The Discovery Agencies Stack. The discovery agencies section may be mapped to Universal Description Discovery and Integration (UDDI) for publication and discovery. Web Services Inspection Language (WS-Inspection) [1] defines an inspection document format and methodology that allows active discovery by registries. Although both UDDI and WS-Inspection are supported by development environments and middleware, only UDDI has begun the standardization process at OASIS while WS-Inspection—a joint specification of both IBM and Microsoft—has not yet been submitted to a standards body.

The Overarching Concerns. For the security concern, the specifications outlined in the Web Services Security Roadmap [12] define the end-to-end Web services security strategy. The roadmap defines a foundation specification—WS-Security—that defines a message security model along with six additional security specifications: a Web service endpoint model (WS-Policy) [3], a trust model (WS-Trust), a privacy model (WS-Privacy), secure conversations (WS-SecureConversation), federated trust (WS-Federation), and authorization (WS-Authorization) specifications. WS-Security is being standardized at OASIS. The Management concern is being defined in both the W3C and at OASIS. The W3C Web Services Architecture Working Group is defining the set of components of the Web services architecture that are to be managed in addition to the types of manageability information they will need to support. Meanwhile, the OASIS Web Services Distributed Management Technical Committee is defining how to access management data for any managed resource using Web services technology. Specifications are being developed for different aspects of the QoS concern. The most significant efforts center around reliable messaging approaches, like HTTP-R [2] and WS-Reliability [10], as well as service-level agreements, like WSLA [9], however, there are no agreed-upon standards to date.

Interoperability. We have presented an overview of the specifications and standards in progress for each layer. However, this discussion has not addressed interoperability, the other critical success factor. The Web Services Interoperability Organization is the consortium charted to enable and promote Web services interoperability. WS-I.org defines "profiles" for sets of specifications, provides implementation guidance, sample applications, testing tools, and test materials for use by Web service's practitioners to assess a Web service's conformance with the interoperability profiles. The initial profile, the WS-I Basic Profile, requires support of WSDL 1.1, SOAP 1.1, HTTP1.1, SSLv3, and UDDI 2.0. The work of WS-I.org will have a great deal of influence over what the final Web services architecture looks like.

Challenges

The vision of fully dynamic, ad hoc business partnerships is not yet viable for a number of reasons. First, the infrastructure standards outlined here must be finished, productized, and widely deployed. Second, industry standard Web services interfaces, or portTypes, must be defined for the various aspects of business-to-business relationships. Finally, XML languages that can describe legally binding business and service-level agreements must be defined and standardized. This is more than a technical challenge; it may be a cultural challenge as well because business relationships often span legal, cultural, language, and national boundaries.

Of the infrastructure standards that must be completed, the most important set is security. Currently four of the six specifications in the Web Services Security Roadmap are available along with sample implementations. Product support from its authors should be following closely. WS-Security is already working its way through the OASIS Security Technical Committee. The other security specifications will need to follow in some standards organization.

Business process automation is the next most important infrastructure element to be finalized. The BPEL4WS specification, authored by IBM, Microsoft, and BEA, is the specification its authors will productize. Sample applications are already available from IBM. BPEL4WS has dependencies on the WS-Transaction and WS-Coordination specifications. Therefore they will also need to go through the standards process.

The Management and QoS infrastructures are the least developed. Unlike the pattern for the security and business process specifications, an initial Web services management specification has not been developed prior to chartering the technical committees working on its standardization. The management infrastructure is currently being specified in working groups in the W3C Web Services Architecture Working Group, OASIS Web Service Distributed Management technical committee, and Global Grid Forum (www.ggf.org) with the involvement of the major management companies, including IBM Tivoli, Computer Associates, and Hewlett-Packard.

There are several aspects to QoS, depending on the layer of the stack at which you are looking. For the network and messaging layer, reliable messaging over HTTP is needed to guarantee once and only once message delivery between business partners. The HTTPR specification has been available since July 2001. The Web Service Reliable Messaging technical committee is just beginning to develop a WS-Reliability specification in OASIS which brings much more industry focus, but no interoperable implementations at this time. For the description and service-level agreements layers, specifications like WSLA are being developed. However, there has been no industrywide attention or product adoption of a particular specification in this area yet.

Legally binding service-level agreements and business level agreement XML documents are just beginning to be discussed and developed.

Applying Web Services Today

Despite the fact that some important layers of the stack need to be standardized and productized, Web services in it current state of support is an excellent application integration technology. Web services can glue together applications running on two different messaging product platforms, enable database information from one application to be made available to others, and enable internal applications to be made available over the Internet. Web services can also be used between two business partners who already have business agreements in place. For point-to-point applications, SSL is already being used for security functionality. Web services are being used to offer programmatic interfaces to applications once browser-driven, like Google (www.google.com/apis) and Amazon.com (associates.amazon.com/exec/panama/associates/join/developer/resources.html). Web services are also being developed as utilities, or pay-per-use program components, like the auto buyers VIN history service that returns the registration and salvage history of a vehicle.

One benefit of this model is that its infrastructure can borrow from the experience of the telephony utilities industry, especially on user-driven service provisioning, usage tracking, and billing. Web service user provisioning is already being worked on in OASIS. Web service deployments have shown that existing assets used within a company can readily become revenue-generating assets.

Conclusion

Web services standards creation is moving up the stack steadily. Product support is outpacing the standards processes and vendors are working from draft specifications in a race to provide important functionality. Therefore current products and implementations may need to evolve as the industry comes to agreement on the final standards. Even so, it is now safe to use Web services for integration projects, since most development and middleware products will shield customer implementations from specification changes. Web services should be leveraged today for application integration and programming model unification. New product plans and business connections being developed should consider Web services as part of the solution.

If Web services are not appropriate for current development, keep in mind this Web-friendly component model is rapidly maturing and steps should be taken to ensure a smooth transition into Web services technologies. Monitoring its progress closely, planning for its integration, and adopting a service-oriented architecture design style will help ensure a strong position for moving into the Web services direction in the future. In addition, looking for potential services and developing those components now so they can be exposed when it is appropriate to make them available to the Web services community will accelerate Web services rollout. Finally, making sure new software investments, including tools and middleware, support the use of Web services across multiple operating system platforms will minimize disruption to environments in order to gain the benefits of Web services technologies.

References

1. Ballinger, K., Brittenham, P., Malhotra, A., Nagy, W., and Pharies, S. Web Services Inspection Language (WSIL) 1.0. (Nov 2001); www.ibm.com/developerworks/webservices/library/ws-wsilspec.html.
2. Banks, A., Challenger, J., Clarke, P., Davis, D., King, R., Witting, K., Donoho, A., Holloway, T., Ibbotson, J., and Todd, S. HTTPR specification. (Apr. 2002); www.ibm.com/developerworks/library/ws-httprspec
3. Box, D., Curbera, F., Hondo, M., Kaler, C., Langworthy, D., Nadalin, A., Nagaratnam, N., Nottingham, M., von Riegen, C., and Shewchuk, J. Web Services Policy Framework (WS-Policy), (Dec. 2002); ftp://www6.software.ibm.com/software/developer/library/ws-policy.pdf
4. Cabrera, F., Copeland, G., Cox, B., Freund, T., Klein, J., Storey, T., and Thatte, S. Web Services Transaction (WS-Transaction), (Aug 2002); www.ibm.com/developerworks/library/ws-transpec/
5. Cabrera, F., Copeland, G., Freund, T., Klein, J., Langworthy, D., Orchard, D., Shewchuk, J., and Storey, T. Web Services Coordination (WS-Coordination), (Aug. 2002); www.ibm.com/developerworks/library/ws-coor/
6. Champion, M., Ferris, C., Newcomer E., and Orchard D. Web Services Architecture, working draft. (Nov. 2002); www.w3.org/TR/ws-arch/, Web Services Architecture Working Group; www.w3.org/2002/ws/arch/
7. Chinnici, R., Gudgin, M., Moreau J., and Weerawarana, S. Web Services Description Language (WSDL) version 1.2 working draft. (Jan. 2003); www.w3.org/TR/wsdl12/, W3C Web Services Description Working Group; www.w3.org/2002/ws/desc/
8. Curbera, F., Goland, Y., Klein, J., Leymann, F., Roller, S., Thatte, S., and Weerawarana, S. Business process execution language for Web services, version 1.0. (July 31, 2002); www.ibm.com/developerworks/library/ws-bpel
9. Dan, A., Franck, R., Keller, A., King, R., and Ludwig, H. Web service level agreement (WSLA) language specification. (Feb. 2003), IBM Web Services Toolkit; www.alphaworks.ibm.com/tech/webservicestoolkit and dwdemos.alphaworks.ibm.com/wstk/common/wstkdoc/ services/utilities/wslaauthoring/WebServiceLevelAgreementLanguage.html
10. Evans, C., Chappell, D., Bunting, D., Tharakan, G., Shimamura, H., Durand, J., Mischkinsky, J., Nihei, K., Iwasa, K., Chapman, M., Shimamura, M., Kassem, N., Yamamoto, N., Kunisetty, S., Hashimoto, T., Rutt, T., Nomura, Y. Web Services Reliability (WS-Reliability) version 1.0, (Jan. 2003); www.hitachi.co.jp/Prod/comp/soft1/wsrm/ WS-ReliabilityV1.0.pdf
11. Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J., and Nielsen, H. SOAP Version 1.2, (Dec. 2002); www.w3.org/TR/soap12-part1, http://www.w3.ort/TR/soap12-part2, W3C XML Protocol Working Group, www.w3.org/2000/xp/Group/
12. IBM, Microsoft. Security in a Web services world: A proposed architecture and roadmap, (Apr. 2002); www-106.ibm.com/developerworks/webservices/library/ws-secmap/

Heather Kreger (kreger@us.ibm.com) is Web Services Lead Architect for IBM's Emerging Technologies, IBM Software Solutions Division, Research Triangle Park, NC. She is the author of *Java and JMX: Building Manageable Systems.*

STANDARDS ORGANIZATIONS

Java Community Process (www.jcp.org) for more information about expert groups and specifications.

Organization for the Advancement of Structured Information Standards (OASIS; www.oasis-open.org) for more information on technical committees and specifications.

Universal Description, Discovery, and Integration Information (UDDI; www.uddi.org) for more information on specifications.

Web Services Interoperability Organization (WS-I; www.ws-i.org) for more information on working groups, specifications, profiles, testing, and samples.

World Wide Web Consortium (www.w3.org) for more information on working groups and specifications.