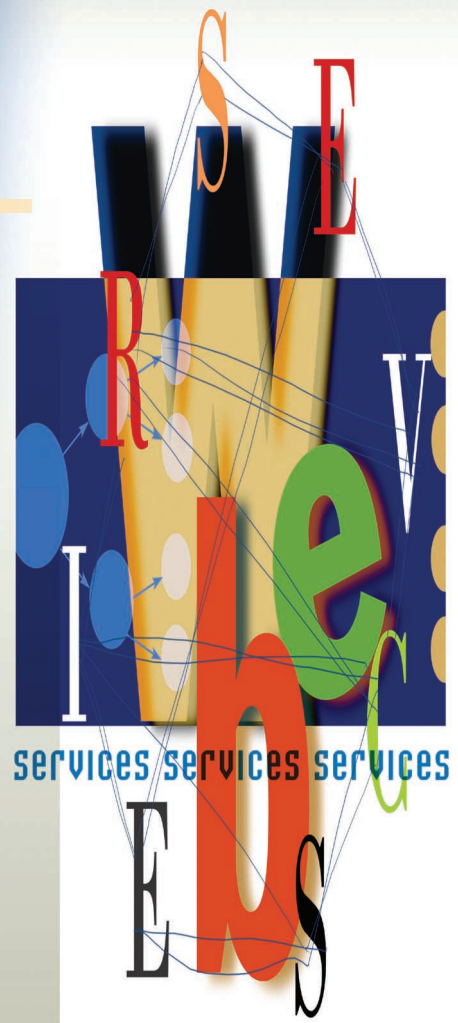


# Web Services Computing: Advancing Software Interoperability

**Web services computing poses significant challenges as developers determine how to leverage emerging technologies to automate individual applications based on cross-organizational, heterogeneous software components.**



*Jen-Yao Chung*  
IBM T.J. Watson  
Research Center

*Kwei-Jay Lin*  
University of  
California, Irvine

*Richard G. Mathieu*  
Saint Louis  
University

**W**eb services have generated considerable excitement in the global computing industry. According to IDC, a global market intelligence firm in the information technology industry, approximately 3,300 Web services-based projects were implemented in North America in 2002 alone.<sup>1</sup> IDC estimates that spending on Web services hardware, software, and services will reach \$15.2 billion by 2007.

For end users, the vast increase in available services can imply either more freedom or more chaos. For developers, Web services computing offers significant theoretical and engineering challenges as they determine how to leverage emerging technologies to automate individual applications.

Three columns recently published in *Computer* have highlighted these challenges. Clay Shirky argued that because Web services disregard the traditional separation between local and global applications, software engineers must carefully consider and anticipate design requirements related to trust, semantics, and coordination.<sup>2</sup> Christoph Langdon described several inhibitors of Web services adoption, including a lack of service provider processes such as metering, accounting, and billing; a lack of semantic consistency in business processes such as ordering, billing, or shipping; and a lack of workflow management mechanisms to orchestrate a

group of specialized Web services in support of a single business process.<sup>3</sup> In addition, the Web services industry faces significant challenges related to security. Martin Naedele provided a detailed framework of seven XML and Web services security standards and suggested that it may be a while before these specifications provide a complete foundation for XML security.<sup>4</sup>

## WEB SERVICES DEFINED

Web services are Web-based applications composed of coarse-grained business functions accessed through the Internet. From a technical perspective, Web services are a standardized way of integrating Web-based applications using open standards including XML, the simple object access protocol, the Web Services Description Language, and the universal description, discovery, and integration specification. XML structures the message, SOAP transfers the message, WSDL describes the available services, and UDDI lists them. XML describes both the nature of each self-contained function and the data that flows to and from it.

Web services must conform to a particular specification format to ensure that each service function will easily integrate with others to create a complete business process. This interoperability allows businesses to dynamically publish, discover, and aggregate a range of Web services through the

**With the strong support that most major market players are providing, Web services may achieve what Corba and DCOM could not.**

Internet to more easily create innovative products, business processes, and value chains. Typical application areas are business-to-business integration, content management, e-sourcing, creating composite Web services, and design collaboration for computer engineering.

While the development and deployment of Web services does not require an underlying technology platform, two different Web services architectures—Microsoft's .NET and Sun Microsystems' J2EE—are competing head to head to become the platform of choice for Web services developers. Although Web services technology is platform-neutral, each camp provides specific development tools and APIs. J2EE is an open standard, but there are several competing J2EE development environments including JBoss+Tomcat, BEA WebLogic, IBM WebSphere, and Sybase EAServer.

### **EVOLUTIONARY OR REVOLUTIONARY?**

Over the years, the goal of achieving software interoperability across computer networks has remained elusive. Standards such as the common object request broker architecture (Corba) and the distributed component object model (DCOM) have met with marginal success. Skeptics are rightfully asking whether Web services will ultimately be viewed as a revolutionary jump or merely an evolutionary step toward software interoperability.

With the strong support that most major market players are providing, Web services may achieve what Corba and DCOM could not. The XML/SOAP approach to Web services offers some significant advantages over Corba and DCOM. While they are both reasonable protocols for server-to-server communications, Corba and DCOM have significant weaknesses for client-server communications on the Internet. Web services use document-style messages that offer the flexibility and pervasiveness that Corba and DCOM cannot provide.

### **NEXT-GENERATION COMPUTING**

Web services will play an increasingly important role in integration initiatives, first within an organization and then between enterprises. Companies that regularly hand off data between systems such as those in the financial services, energy trading, high-tech manufacturing, and telecommunications industries are ideal candidates for Web services technologies. Mike Burner argues that XML schemas will be central to how organizations communicate and that the unique characteristics of Web

services suggest new design patterns.<sup>5</sup> These patterns include messages with a high degree of granularity, asynchronous messaging, bidirectional services, endpoint discovery, and protection against mishandling messages received more than once.

Web services standards permit application-to-application interoperability, but the coordination of a set of Web services working toward a common end is an open issue. For example, developing transaction-based applications requires taking steps to ensure that each transaction is completed in its entirety (or not at all), to permit the rollback of interrupted transactions, and to efficiently broker distributed processes within the transaction. As a result, competing process definition languages such as the Business Process Modeling Language (BPML) and the XML Process Definition Language (XPDL) have emerged. In addition, separate business collaboration languages have been introduced, including the XML-based business transaction protocol (BTP), Electronic Business using Extensible Markup Language (ebXML), Business Process Specification Schema (BPSS), Business Process Execution Language for Web Services (BPEL4WS), and the Web Service Choreography Interface (WSCI). These standards are still developing, but they clearly indicate the momentum behind the Web services computing community.

### **IN THIS ISSUE**

The articles selected for this special issue highlight four important contributions to the Web services computing community.

In "Turning Software into a Service," Mark Turner, David Budgen, and Pearl Brereton describe a demand-led software market in which businesses can assemble systems from component Web services when needed to address a particular requirement. According to this software-as-a-service concept, rather than developing software components and binding them together to form a single, rigid solution, future developers will offer applications as a federation of Web services that are only bound together at the point of execution. This article provides an excellent introduction to the promise of Web services computing and in the process points out existing gaps in the technologies.

"Standards for Web Services Orchestration and Choreography" by Chris Peltz highlights the many challenges in integrating Web services in a collaborative manner. Orchestration refers to an executable business process in which interactions with both internal and external Web services occur at the message level. These interactions include business logic

and task execution order, and they can span applications and organizations to define a long-lived, transactional, multistep process model. Orchestration always represents control from one party's perspective. This distinguishes it from choreography, which is more collaborative and allows each involved party to describe its part in the interaction. This article provides an overview of emerging specifications currently available to meet the requirements for orchestrating Web services.

"CPXe: Web Services for Internet Imaging" by Timothy Thompson, Rick Weil, and Mark D. Wood describes the Common Picture eXchange environment, an industry initiative that leverages the Web services paradigm to serve the electronic photographic services market. CPXe is an interoperable service delivery framework that combines open standards for exchanging digital images and order information with a central directory to give providers access to an expanded market while offering consumers a broad range of digital imaging services.

In "Ontology-Mediated Intranet Web Services," Tse-Ming Tsai and his colleagues describe a Web services-based smart office task automation framework. SOTA uses Web services, an ontology, and agent components to create an integrated information-service platform that offers user-centric support for automating intranet office tasks. System developers can use this application to attach the shared ontology's concept and property names onto Web service methods and parameters, using additional ontological information to allow increased interoperability.

**T**he Web has become the user interface of global business, and Web services now offer a strong foundation for software interoperability through the core open standards of XML, SOAP, WSDL, and UDDI. The interaction models that fully realize the potential agility of Web services computing are just beginning to emerge. We hope the articles in this special issue illuminate both the promise and engineering challenges now facing us in this arena. ■

---

#### Acknowledgments

We acknowledge the support provided by the IEEE Task Force on Electronic Commerce (<http://tab.computer.org/tfec>). The IEEE Computer Society established the TFEC in May 2001 to promote e-commerce research and education and to

support the implementation of e-commerce systems and solutions. We also thank all those who contributed to this special issue, especially the reviewers, whose diligent and critical review of the submitted papers is sincerely appreciated.

---

#### References

1. R. McMillan, "IDC: Web Services to Enable \$4.3B Hardware Market by 2007," *Computerworld*, 23 May 2003; [www.computerworld.com/developmenttopics/development/webservices/story/0,10801,81496,00.html](http://www.computerworld.com/developmenttopics/development/webservices/story/0,10801,81496,00.html).
2. C. Shirky, "Web Services and Context Horizons," *Computer*, Sept. 2002, pp. 98-100.
3. C.S. Langdon, "The State of Web Services," *Computer*, July 2003, pp. 93-94.
4. M. Naedele, "Standards for XML and Web Services Security," *Computer*, Apr. 2003, pp. 96-98.
5. M. Burner, "The Deliberate Revolution: Transforming Integration with XML Web Services," *ACM Queue*, Mar. 2003, pp. 28-37.

*Jen-Yao Chung is a research staff member and senior manager at IBM T.J. Watson Research Center. His research interests include business process integration and management, e-commerce, electronic marketplaces, and Web application systems. Chung received a PhD in computer science from the University of Illinois at Urbana-Champaign. Contact him at [jychung@us.ibm.com](mailto:jychung@us.ibm.com).*

*Kwei-Jay Lin is a professor in the Department of Electrical Engineering and Computer Science at the University of California, Irvine. His research interests include real-time and distributed computer systems. Lin is an associate editor for IEEE Transactions on Parallel and Distributed Systems. He received a PhD in computer science from the University of Maryland, College Park. Contact him at [klin@uci.edu](mailto:klin@uci.edu).*

*Richard G. Mathieu is an associate professor in the Department of Decision Sciences and MIS, Saint Louis University. His research interests include intelligent decision support and distributed enterprise management. Mathieu is the IT Systems Perspective column editor for *Computer* and is an associate editor for IEEE Transactions on Systems, Man and Cybernetics—Part A. He received a PhD in systems engineering from the University of Virginia. Contact him at [mathieur@slu.edu](mailto:mathieur@slu.edu).*