By Carl Jacobson

Web Services: Stitching Together the Institutional Fabric

Expect to be hearing a lot about *Web services* in the coming year. Web services are modular, self-contained, component applications that can be discovered and used by other applications and other Web services. The most-promising aspect of Web services, however, is their ability to resolve the differences among shared, networked applications. Under this new, rapidly emerging model for Internet computing, various Web applications can be stitched together: applications from different vendors, of various vintages, written in different languages, and running on disparate platforms can easily communicate and cooperate.

Web services differ from the proprietary connectors and interfaces traditionally used to integrate disparate applications. The loosely coupled nature of Web services is more forgiving than the interface requirements of traditionally integrated systems. The sharing of a tagged XML document is more flexible than the sharing of a strictly formatted interface record. Web services are self-describing and self-announcing, meaning that all specifications related to the use and behavior of a service are part of the service itself. Additionally, Web services allow an application to invoke a remote process or application as if it were part of the invoking application.

Naming this model of distributed computing *Web services* is, at worst, an injustice and, at best, confusing. After all, customer services have been provided via the Web for a number of years—"services on the Web." But the new use of the term *Web services* does not refer to the large, complete, discrete business and customer services that happen to be on the Web. Rather the term *Web services* refers to a specific collection of Internet technologies used to build, publish, and relate business and learning components across the network. Although the term *Web services* is confusing, it may be easier to understand than the alphabet soup of acronyms used for the technologies that support this new computing model: XML, SOAP, UDDI, WSDL.

XML, like its cousin HTML, is a markup language. "ML" stands for "markup language"; "X" represents "extensible." Whereas HTML uses text tags to describe how the contents of a document will appear (e.g., tags for bold, header, underline, font, image), XML uses text tags to describe how the contents of a document will be used (e.g., tags for invoice number, item number, price). Whereas HTML is used to describe how a document will be displayed on a browser, XML is used to describe how a document can be processed by a program, application, or system. By separating data from presentation, XML allows information to be passed between applications or systems or devices. For example, many colleges and universities mark up grade information with the formatting tags of HTML in order to deliver grades to students via the Web. However, to share grade information between a learning management system (LMS) and a student information system (SIS), XML tags allow each side of this partnership to understand the information. In fact, LMSs from two competing vendors could share grade information with the SIS by using similar XML "dialects."

SOAP (Simple Object Access Protocol) is a messaging standard used to permit Internet applications to communicate. SOAP allows process control to be passed between two networked applications. SOAP is not a language or a component; it is a communications protocol that defines encoding rules to allow disparate systems to interact. Although the SOAP standard began at Microsoft, it is now supported by all major vendors and allows Web services to be developed in virtually any programming language.

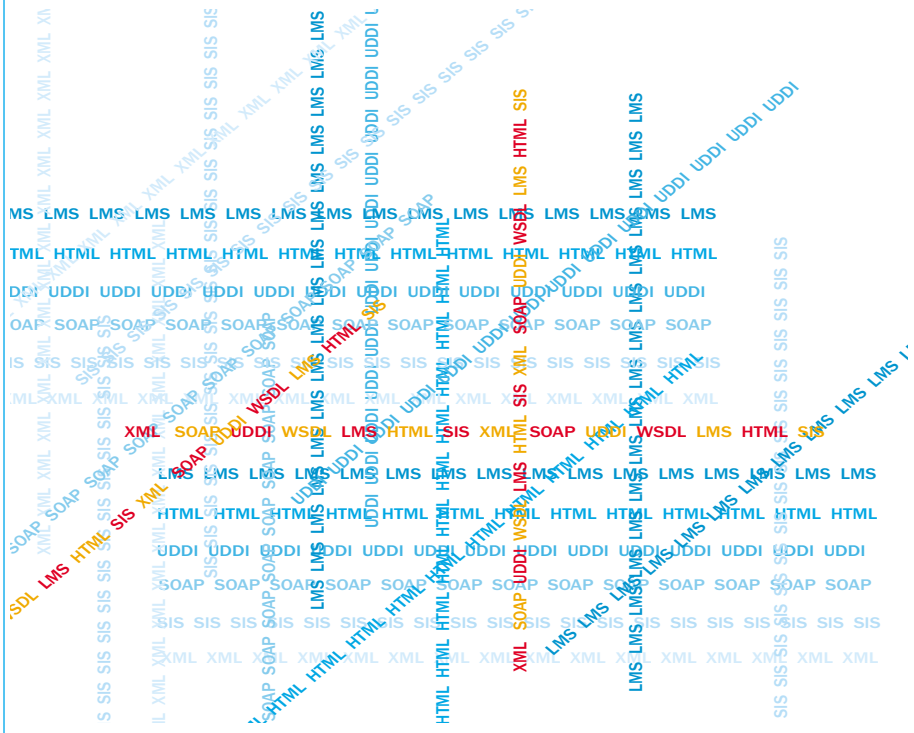
UDDI (Universal Description Discover and Integration) is the "yellow pages" of Web services—a directory used to allow an application to announce its availability and service to other applications. Web services registered in a UDDI directory may be discovered by Web applications hunting for specific services, goods, or partners. An application can use UDDI to find and "call" a partner in order to conduct business.

WSDL (Web Services Description Language) is an XML format used to describe the capabilities and interoperability required of a Web service. WSDL provides an abstract definition of the communication details between two applications. For example, WSDL would define the parameter data types to be exchanged between two cooperating applications.

The use of these standards allows applications to be built and deployed so that data and business logic can be provided to partner applications for their own use. This means that vendors, information providers, and government agencies can use Web services to expose data to partners, allowing those partners to use the

data for their own business purposes. Since Web services can be written in any language on any platform, they can be used to reveal information locked away in legacy applications. Existing applications can be "opened" to the network using Web services, and of course, new applications can be developed using the Web services model.

Examples of useful Web services in higher education might include the following:



- A class roster service that provides class rosters to online gradebooks and campus-wide learning management systems
- A credit card service that accepts credit card and payment information and returns bank authorization indicators
- An interface service that permits students, faculty, and researchers to use an online art collection on their own terms
- A student loan tracking service that allows students to monitor the status of guaranteed student loans

Web services may be internal to the institution, enabling the applications of the electronic campus to communicate effectively and work in concert. The class roster example shows how faculty spreadsheets, an institutional SIS, and campus-wide LMS systems can all share class enrollment information within the confines of the institutional Web. Or Web services can be handled by external service providers. In the credit card example, the service might be licensed from the banking industry so that an institution can allow all Web-based payment applications to use a single point-of-payment authorization. In addition, information providers, perhaps providers of scholarly information, can make learning resources available directly for pedagogical applications developed by faculty and researchers. The online art collection interface, for example, can overcome the limits of a single-user interface, allowing valuable learning resources to be used in many different and creative ways. Finally, government agencies can more easily communicate with several thousand disparate campuses and several million individual students using Web services. In the last example, student loan guarantee agencies can be brought into the communications loop to provide a single information service that can be "plugged in" to a wide variety of campus information systems.

Major vendors are moving to support Web services. Most notably, Sun Microsystems' Sun ONE (Open Net Environment) and Microsoft's .NET both address the needs of distributed computing with support for the Web services paradigm. Products are rolling out to facilitate the construction and management of Web services. But standards, tools, and products for Web services are still under development. Some existing weaknesses that need future attention include security, state management, and service monitoring.

The promise of Web services will be clouded with much hype. What should colleges and universities do now to be prepared for this new model? Closely follow the development of Web services and associated industry standards and products. As Web services relating to the business, teaching, and learning activities of the institution become available, take advantage of these offerings. Determine how Web services might improve interapplication communication of Web applications built on campus. Encourage information providers and software vendors to learn about and adopt the Web services model to expose their data to the business need of the higher education institution. Loosely couple the college or university Web applications to provide a tightly consolidated institutional image. Look for Web services to help formulate a single, coherent, well-branded image of the institution on the Web—a Web image that appears, to each member of the community, as a consolidated, personalized view of the institution.

In the past, the promise of integrated systems could not be realized. Now, however, the Web becomes the "great resolver" as Web services are providing the technologies, tools, and standards needed to stitch together network applications into what will become the fabric of the higher education institution.

Carl Jacobson was one of two recipients of the 2001 EDUCAUSE Award for Leadership in Information Technologies. Director of Management Information Systems at the University of Delaware, Jacobson has been a pioneer in the use of the Web to provide access to institutional administrative information resources. He was an early adopter of the Java language for administrative Web development and was instrumental in the formation of JA-SIG, the Java in Administration Special Interest Group. Jacobson is the recipient of a grant from the Andrew W. Mellon Foundation in support of the uPortal open-source software effort.