

# The web services revolution

*It's the latest technical revolution—perhaps the technical revolution. As vendors scramble to position themselves as thought leaders, what does a campus user or buyer need to know?*

By RICHARD JACIK

If you sat out the client-server revolution, slept through the object-oriented development revolution, ignored the Java revolution and are dismissing web services as one more technological bullet you can dodge, read on. This revolution deserves your attention.

Granted, there is a lot of hyperbole surrounding web services and the marketing machines are fully buzz-

word-compliant. Meanwhile, many still misunderstand and misuse the term—vendors included. Simply put, web services are computer programs or pieces of computer programs that can be made accessible across a network. A web service often operates behind the scenes providing some data, or manipulating some data or performing a computation that can be used by yet other applications.

So web services may be said to be computer-to-computer programs, or what David Truog, an analyst with Forrester Research calls, “a network of things, not people.” He estimates that the bulk of internet traffic by 2006 “will be digital chatter between software, not email or web pages aimed at human users.” However, it is humans who will benefit.

## How web services work

Think about the FedEx tracking number tied to the delivery of your order that you may be viewing in your own school’s procurement system. It’s not hard to visualize a further display of the actual shipping status direct from FedEx. At a glance you know when the package was picked up from the manufacturer, where it is right now (on a truck downtown), and when it’s expected to show up at your office. Value was added to your procurement system by the FedEx web service described above.

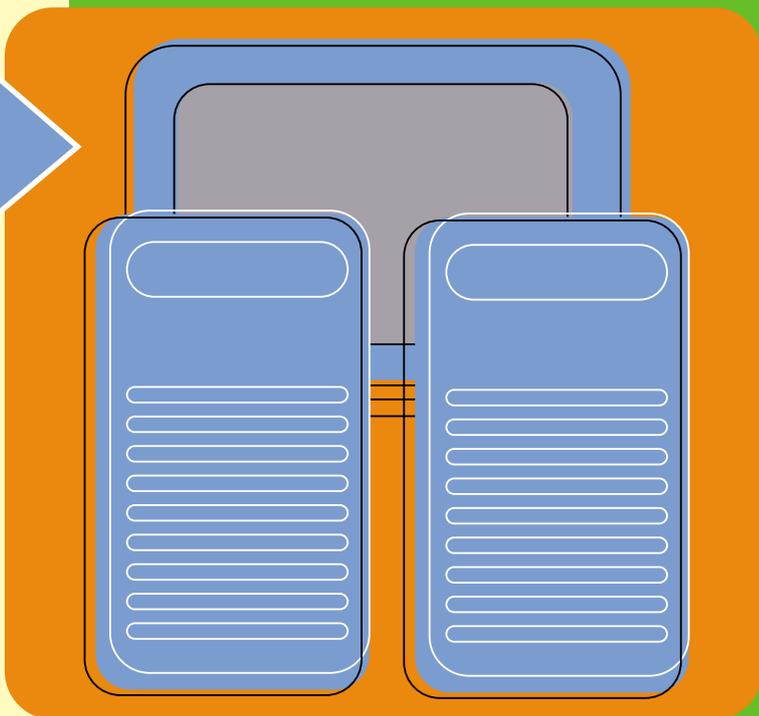
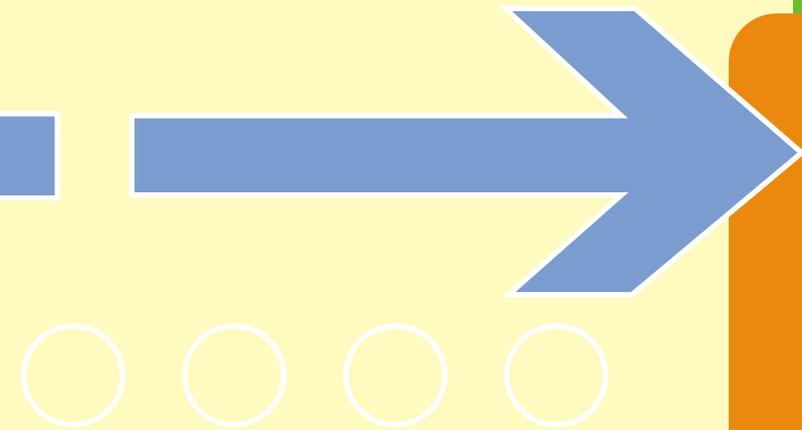
Your local application had a key bit of information—the purchase order number, or the SKU number of the item purchased. But a behind-the-scenes computer-to-computer link provided very useful delivery

## What’s in a name?

You may believe your institution is already providing web services—that is, services over the web to students, faculty, and administrators. That might make them web services in a lexical sense. But they are not really web services in the technical sense. Which is why ‘web services’ may be the most confusing and unfortunately-named technology ever, one that only a techie like Bill Gates could love.

A web-page response sends simple data sets to your browser intermixed with display tags so that your browser knows how to format the display. A web service response returns just XML data sets. Figuring out how the data should be displayed or used is the responsibility of the requesting program. This “headless” nature of web services can make understanding them difficult without a detailed technical look under the covers.

Given the murky definition of web services and the technical embarrassment of lagging behind, less scrupulous vendors can and do find it easy to claim that their applications include fully integrated web services capabilities.



information, perhaps across several time zones and certainly across several network connections.

Meanwhile, FedEx cares only that it delivers the correct delivery data to your system. It doesn't really care about the nature of your system or what you'll be doing with the data. Your system receives the data and optimizes the display and format. It might just as easily have been logged to a database, or fed into a GPS or mapping system.

### One for all

The technical steps FedEx follows to provide the web service to one school is exactly the same as for any other interested web service consumer. In fact, web services standards insure that FedEx doesn't have to worry about whether the receiving system is PeopleSoft, Oracle, SCT or home-grown. It is equally unimportant whether you run your procurement application on a desktop, a UNIX server or a mainframe. Under the web services paradigm, all systems are created equal.

Such things were technically possible before the advent of web services, but they were highly dependent upon the internal technologies and data formats of the applications involved. Only key trading partners were willing to undertake the significant effort and resources required to make disparate systems "talk" nicely with each other.

The promise of web services is to make such system-to-system interactions simple, technologically neutral and exceedingly commonplace. So far, vendors are actually doing a good job making web services work in a standard way. Software companies, for their part, are planning on web services to provide new revenue opportunities.

### Service for a price

It is possible that your next student information system will calculate GPAs by accessing that functionality on a remote server as a web service. Or perhaps your payroll system will rely on third-party tax rate calculations from a server that specializes in just that web service.

In the early days of web services, access will likely be free or very cheap. As the marketplace and the business model mature, however, software as services isn't likely to save a school any money. Charging per use will become essential to making a web services business model work.

## Web services in the marketplace

Here's what some higher education vendors say they're doing with web services.

Warning! Separating the vaporware from the software is more difficult than it should be.

**Microsoft** ([www.microsoft.com](http://www.microsoft.com)), arguably the web services leader, has adopted web services as a key part of its .Net initiative. Microsoft also has an early lead in tool-building. Microsoft's effort and focus on web services may be likened to its internet drive in the 90s.

**IBM** ([www.ibm.com](http://www.ibm.com)) is another strong proponent of web services and has, through its WebSphere infrastructure technologies, begun to enable web services interaction. IBM, Microsoft and others are cooperating on defining standards to make web services interoperate well.

**Oracle** ([www.oracle.com](http://www.oracle.com)) is supporting web services through its development tools (JBuilder) and through its 9i Application Server, upon which Oracle administrative applications rely.

**Sun** ([www.sun.com](http://www.sun.com)) supports web services through its development tools and the Sun ONE (Open Networking Environment) initiative. Sun ONE, like Microsoft's .Net, is a collection of technologies that include support for XML and web services. In Sun's case, these will be consistent with the expected upcoming J2EE (Java 2 Enterprise Edition) specification for web services.

**PeopleSoft** ([www.peoplesoft.com](http://www.peoplesoft.com)) has added web services integration capabilities to its products via Integration Broker software, which acts as a common web services hub for system-to-system integration.

**SCT** ([www.sct.com](http://www.sct.com)) has announced support for web services through its use of the Oracle 9i Application Server and Microsoft .Net technologies (for the SCT Matrix application).

**Blackboard** ([www.blackboard.com](http://www.blackboard.com)) is developing a version of its Blackboard Building Blocks integration APIs that are .Net compliant and will allow integration with Blackboard using web services.

# Count on KHEAA



**For Exceptional  
Services to Students,  
Schools, and Lenders!**

**A balanced mix of need-based and merit-based aid.**

**The lowest cost student loans in Kentucky with BestStart and Best in Class.**

**Online services:**

- eSign processing
- Account access
- Consolidation application
- itsmoneybaby.com counseling site
- Disbursement/maintenance
- KHEAA Partners news



Kentucky Higher Education  
Assistance Authority

800.928.8926  
www.kheaa.com

KHEAA does not discriminate on the basis of race, color, national origin, sex, religion, age, or disability in employment or services and provides, upon request, reasonable accommodations to afford individuals with disabilities an equal opportunity to participate in all programs and activities. Printed with State Funds KRS 57.375.

## 90 Seconds on XML...

While HTML is designed to display data, XML is designed to focus on the data and its structure. Using special tags to demarcate where, in long data streams, certain data elements are to be found, the recipient of the data streams can parse for individual elements, groups of elements or sets of groups of elements.

An XML data set that is meant to contain the contents of a magazine article, for instance, might look something like:

```
<publication>The Greentree  
Gazette</publication>  
<title>The Web Services  
Revolution</title>  
<author>Richard Jacik</author>  
<pubdate>Jan 1, 2003</pubdate>  
<paragraph>  
This is the text of Paragraph 1  
</paragraph>  
<paragraph>  
This is the text of Paragraph 2  
</paragraph>  
<sidebar>  
This is the Sidebar text  
</sidebar>  
</article>
```

XML is extremely useful, but by itself, it doesn't really do anything. Programs must parse the data and

then act on it. And as far as web services go, XML is only half the battle. To be specific, it is the half of the battle that standardizes how data will be delivered to and received from some remote system that is performing the web service.

The other part of the equation is SOAP (Simple Object Access Protocol) which itself is based on XML. SOAP defines the standard way to use XML to request that a remote system *do something*. That something might include a request to calculate a GPA, authorize a credit card, or complete a travel booking. Again, because the base language, XML, is simple text, it theoretically won't matter whether Visual Basic on your Windows system is talking with Java on Solaris or COBOL on a mainframe.

It's not a stretch to imagine all sorts of university data, from student transcripts and class schedules to purchase orders being represented as XML data sets. The *lingua franca* of XML data sets is plain text, not binary formats used up to now in machine-to-machine communications. Therefore it shouldn't matter who produces the data sets. Therefore, a lot of integration problems will find solutions.

Charging and collecting micro-payments (tiny charges for tiny services) is still a sticky issue standing in the way. How about a desktop version of Microsoft Word that uses web services to spell-check your document for a dime each time, or maybe a penny for each misspelled word?

### All together now

Microsoft's vision of web services is one in which the data and programs in your personal database on your PC interact seamlessly with processes and data on a server. For example, contact

and schedule information in Outlook on your PC might automatically synchronize and share certain pieces of data with enterprise systems, perhaps to validate and update email addresses or campus phone numbers.

Information stored in Word might actually embed data elements from your administrative systems into documents stored on a file server. In that way, policy and procedure manuals could be automatically and consistently updated to reflect changes in your administrative systems.

While most vendors have started

**"Web services become important to the extent that they facilitate rich capabilities like the campus portal," says Richard Kogut, CIO of University of California, Merced.**



**“Plan** for system architectures that don’t rely on hubs and spokes for data movement. Rely on a confederated set of applications,” Sandy Senti, executive director of technology strategy and support at Stanford University.



“Start by leveraging XML and its supporting tools as much as possible,” says Senti. “It begins an education process that can end with the full implementation of web services. In the meantime, plan for system architectures that don’t rely on hubs and spokes for data movement. Rely on a confederated set of applications, each interacting intelligently with the others.”

supporting or are at least voicing support for web services, the first uses won’t stretch the imagination too much. They are likely to be deployed on your campus by your IT group as a more expedient way to get your financial system to talk to your student system. Don’t let this ultimately unexciting, but important, first deployment dull your senses to the larger implications of the web services revolution.

Things that have been difficult, like getting disparate data sources to feed commonly formatted information to a campus portal, will become easier as product vendors make their systems more accessible. Costly or complicated things will become less so. For example, system-to-system interaction with each high school that supplies freshman to you, or with every employer that hires your graduates, will be possible.

### Upside potential

In fact, web services may yield unparalleled opportunities to take advantage of a closer technical rela-

tionship with trading partners and peer institutions. Those closer relationships may yield efficient uses of administrative dollars on campus or improved marketing.

“Web services become important to the extent that they facilitate rich capabilities like the campus portal,” says Richard Kogut, CIO of University of California, Merced. “XML technology will make that integration easier. If SOAP succeeds as a protocol that separates applications from their underlying protocols, then it will truly prove to be a compelling technology.” (Of course, in the U.S. any new technology breeds new acronyms. For more on XML and SOAP, see the sidebar in this article.)

Sandy Senti, executive director of technology strategy and support at Stanford University, is examining how web services and XML might be used to replace some home-grown infrastructure applications developed before web services became mature. She offers some advice to campuses that are struggling with next steps.

### Other concerns

If you’re already thinking about the security problems from systems all over the internet having frequent interactions with the systems on your campus, you’re not alone. Indeed security is a key issue in the cautious adoption of web services. Several IT industry groups, including one with both Microsoft and IBM as members, are working on defining standard security approaches.

So far, industry leaders are playing well together. This collaboration between competitors is one reason that web services is *the* technical trend to watch, learn from and leverage. When IBM, Microsoft, and Sun all pretty much agree on the technical future of interoperability, it is time to pay attention. ■



**RICHARD JACK** is president and co-founder of Information Methodologies Inc., a leading enterprise web integrator for higher education. Contact him at 703.435.0370 or [jack@itsuccess.com](mailto:jack@itsuccess.com).

## Friendly.



95% of Campus Door customers would recommend us to a friend and 97% of our borrowers rate their experience with Campus Door as Outstanding or Favorable. Find for yourself by contacting us today at: 1.800.786.0002 or [www.campusdoor.com](http://www.campusdoor.com)

