Web Services—A New Energy Management Tool

Steve T. Tom, P.E., Ph.D.
Automated Logic Corporation

ABSTRACT

Integrating building systems has long been touted as an emerging technology, a development which will provide energy engineers with a powerful tool for optimizing energy use. The technical ability to integrate building systems has existed for many years, but the difficulties involved in writing custom interfaces to all the computer systems involved made it impractical in all but a few "showpiece" applications. Recently a new technology called web services has gained acceptance, and this technology promises to make building integration practical. This article takes a non-programmer's approach to web services, describing what they are and how energy engineers can use them. A discussion of the forthcoming ASHRAE standard for web services is also presented.

INTRODUCTION

Many years ago I worked for a controls company that built a pneumatic "computer" out of its HVAC control components. We dragged this contraption around to trade shows, challenging all comers to play tick-tack-toe against a brain that ran on compressed air. This was a great demonstration of what was possible with pneumatics, but it definitely was not a demonstration of what was practical. Inside the machine there was a veritable rat's nest of tubing, relays, and controllers, all of which would periodically break, leak, or disconnect themselves. We never felt comfortable demonstrating this computer unless there was a cigar smoker with us, because we never knew when we'd need someone to blow smoke across the tubing while we watched for leaks.

In some ways, the concept of an "integrated building" is similar to that pneumatic computer. Certainly it's possible to integrate HVAC, fire, security, utility, accounting, HR, and other computer systems within a building, but how often is it actually done? For years the trade press magazines have carried stories about the executive who swipes his access card as he drives into the parking garage, secure in the knowledge that this will also turn on the lights in his office, adjust the HVAC to his preference, start the coffee, and send the executive elevator to the first floor so it's waiting for him when he enters the building. Great concept, but has anyone ever done this on anything other than a "showpiece" project? Once you start digging into the details of what it takes to get all these systems to talk to one another, you quickly decide it's just not practical for most projects. Standard communication protocols like BACnet and LonWorks help, but their influence doesn't extend beyond the world of building automation. For instance, if you want your human resources (HR) system to automatically cancel a fired employee's login on your fire, security, and HVAC control systems, BACnet and LonWorks aren't going to help because the HR system doesn't use these protocols.

Recently a new technology called "web services" has entered the field, and it promises to change everything. Web services are not a replacement for any of the existing protocols; they're a standard way for computers to exchange data regardless of any other protocol the computer may use. Web services started in the business world, so there's a good possibility your HR system already supports it. Web services are platform independent, so even if your HR system is running on a Mac, on Linux, or on a UNIX mainframe, it can still support web services. While previous technologies made it possible to communicate with these systems, the cost of the custom programming required made it impractical. Web services make building integration practical.

WHAT ARE WEB SERVICES?

Web services are a standardized method for machine-to-machine data exchange over an IP data network. While the term "web" indicates this exchange can take place over the world wide web, it doesn't have to. Web services work just fine over a local area network or a

dedicated network connection. As the term "services" implies, a web service can do more than just regurgitate data. Computers can be programmed to run complex applications when requested by a web service and send the results back to the requesting computer. Indeed, some experts are predicting that vendors will offer specialized "sub-applications" on a time-share basis, and commercial software products will use web services to call on these routines as required and tie them all together in a seamless user interface. A word processing application, for example, could automatically use web services to execute functions such as spell checking, language translation, exporting to other formats, etc. While the services likely to be used to integrate building systems will probably not be nearly as elaborate as these, functions such as unit conversion and interpolating trend samples to match samples from another system.

Web services use XML as the basic file structure for the data exchange. XML stands for "eXtensible mark-up language." You don't need to understand XML to use web services, but a few general characteristics may be of interest. First, XML is a structured way of creating an ASCII text file that contains both the data itself and at least a limited amount of self-documentation. The structure allows a computer to quickly search the file and find the data of interest, *if* it's been programmed to know what to look for. The fact that it's written in ASCII text and is self-documented means a human can read the file, study the documentation, and program the receiving computer to know what to look for. Often there are other resources available to help him interpret the data, such as an XML schema or a web services description language (WSDL). The "extensible" aspect of web services comes from the fact that the XML file structure is not rigidly defined at the moment the file is created, but in fact can be added to (extended) over time if the need arises. Computers that read this new file ignore any sections that don't apply to them as they search for data of interest, rather than rejecting the file because the structure has changed.

Web services use a protocol called SOAP (simple object access protocol) to package information for transport over the web. The requesting computer will send a SOAP request for data, and the host computer will use SOAP to send an XML file in response. Again, you don't need to understand SOAP to use web services, but it's good to at least be aware of the acronym because you will sometimes hear the term "XML/SOAP" used to describe a web services data exchange.

EXAMPLES OF WEB SERVICES

Web services were developed to facilitate business to business (B2B) transactions over the web. As businesses started using the web as their primary means of communication, they realized they could do more than simply transmit an order form from one human to another. They could use the web to connect the computers directly, placing orders, transferring funds, tracking shipping, etc. *if* they had a standard way for the computers to communicate. Web services provided this standard, and the first web services applications were written for B2B transactions. Several vendors developed programming tools to help create web services (Microsoft .NET, IBM WebSphere, SunOne, etc.), but the services themselves were platform independent.

In time, the use of web services expanded to include services available to general internet users. Microsoft Passport was an early example. Using Passport, internet shoppers could enter their credit card and shipping information once, on the Passport computer, and this computer would then supply the information as required via web services whenever the customers made purchases on a website that supported Passport. Amazon.com was another early supporter, and it freely supplied code to its business partners to enable them to build web services links to the Amazon host computer. As a result, when you shop Amazon.com, you may view and order products which are being offered for sale by an Amazon partner rather than by Amazon itself. Similarly, when you visit the website of an Amazon partner, you may be offered Amazon products which you can order directly through the partner's website.

The state of New Mexico used web services in a manner that is of particular interest to building integration. Like many large organizations, the state government includes multiple bureaus and branches, each with its own specialized website designed to best suit its needs. This can be a problem for the common citizen, who needs information such as "how do I register my car" but who doesn't even know which branch of the government handles car registrations, let alone its website address. The solution adopted by New Mexico was to create a portal, a central website which uses web services to integrate information from many other websites. Users can go to the portal site and see information from multiple government websites, get answers to basic questions, view statistics, download forms, and perform many similar functions

without ever being aware that the information they're getting is being supplied from a different website via a web service. Then, if they need more detailed information, they can click on a link that will take them to the other website.

WEB SERVICES AND ENERGY ENGINEERS

The preceding examples show how web services can help shoppers on the internet, but how can they help an energy engineer? Once building automation systems (BAS) manufacturers started making their software support web services, one of the first applications was tenant billing systems. Utility meters are often integrated into the BAS, and with web services it's fairly simple to pass this consumption information to the landlord's accounting system. Even if baseline utilities are included in the rent, after-hour and weekend use is often a billable item. Following the principle that "you don't conserve what you don't pay for," even a simple energy accounting system can make a significant impact.

The King Street Wharf project in Sydney, Australia, shows another way in which web services can be used by an energy engineer. This building was required to stay within strict energy use limits by SEDA (Sustainable Energy Development Authority, since incorporated into the Department of Energy, Utilities and Sustainability.) In some ways this is like our LEEDS program in that the building must not only meet these guidelines when new, but must document continued compliance year after year. The compliance guidelines cover several areas of environmental responsibility, not just energy conservation, so the project engineers created a compliance report that uses web services to gather information from multiple building systems, including the BAS.

The fact that web services support is being built into standard applications like Microsoft Excel makes it easy for energy engineers to create their own reports. A school system in Texas is doing just that, gathering information from the BAS installed in multiple schools, normalizing the data to provide energy use/square foot-degree day, and creating an "energy report card" for their schools. Comparing schools in this manner lets them quickly pinpoint their greatest energy problems and spend limited facility money where it will do the most good.

A commercial building in Texas provides an interesting example of

the practicality of web services. In this application, the landlord not only wanted a standard tenant billing program, he also wanted a "virtual thermostat," a small application that would run on a user's computer and enable them to adjust the temperature and schedule of their office to best meet their needs. (When you're metering and billing a tenant's utilities, you can give them flexibility like this and still save energy.) The user interface for the virtual thermostat was something the local BAS vendor couldn't develop in Excel, so they hired a programming firm to create the program. The initial estimate was high, because the programmer was assuming he'd have to learn a proprietary protocol to link with the BAS. When he found out the BAS supported web services, he cut his estimate in half.

More exciting uses of web services are still being developed. Several pilot projects have demonstrated the practicality of using web services to implement real-time energy pricing. Under this scheme, a utility company prices its energy according to its actual costs. When demand is high, standby generators are running, and transmission systems are running at close to capacity, the price is high. When demand is low and the utility can take full advantage of alternative fuels or other cost-saving strategies, the price is low. Customers use web services to monitor this price and automatically implement cost saving measures (adjust setpoints, turn off non-critical loads, bring standby generators online, etc.) as appropriate. Similarly, some engineers are experimenting with using web services to bring weather forecasts into a BAS and use this information to optimize control strategies. Ice storage systems are a natural for this type of control, as to correctly calculate the amount of ice to generate at night the system needs to know how hot it will be on the following day. Similarly, if a system is going to automatically start a boiler that takes six hours (or longer) to come up to capacity, it's important to know in advance how cold it will get during the night.

Energy engineers often need information from multiple systems, not just the BAS, and this makes the idea of a facilities portal attractive. Using web services, information from a number of sources can be integrated into a single website that provides the energy engineer with the information he most needs to know. Utility costs, utility usage, alarms from the BAS, maintenance work-orders for the central chiller plant, budget reports—all can be integrated into a single facilities portal.

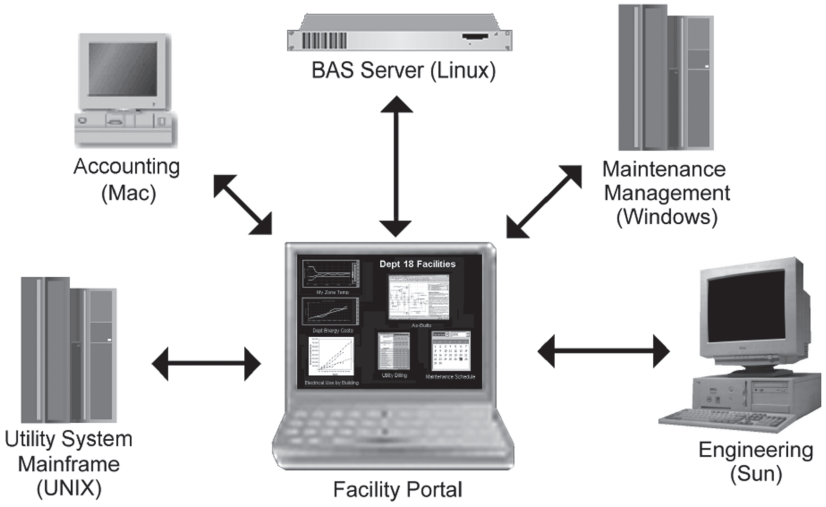

Figure 1. A Facilities Portal

EMERGING STANDARDS

Web services provide a standard way for computer systems to exchange data, but this still requires someone to write the code to make this happen. The acceptance of web services and programming tools like .NET make this much simpler than it was when every connection required proprietary data calls, but the job could be simpler still if there was a standard way to present the data. Imagine a large college campus with thousands of points in the BAS. How do you locate the information you need? How do you tell if a particular piece of data is valid, or if that point has been "locked" due to a faulty sensor? How do you pull up the trend data from a utility meter, say at ten-minute intervals over the last two weeks? Now imagine you were trying to consolidate and compare data from 20 state universities, each of which had a different BAS? Clearly there is a need for a standard way to organize data and a standard set of web services to read or write the data.

As with other aspects of web services, the need for this type of standardization originated in B2B applications. Programmers who were trying to consolidate data from multiple accounting programs faced a challenge no less daunting than that of integrating multiple BAS. There was a similar need to standardize the way shipping systems and other

business applications supported web services, and so the concept of "vertical standardization" developed. It was impractical to develop one standard way of presenting data which would meet the needs of every industry which needed to use web services, so efforts developed within each industry to standardize the way their data would be presented. Within the building automation industry, there are three independent efforts to create a standard. ASHRAE began developing a web services standard several years ago, under the direction of its BACnet committee. More recently, CABA initiated a program (since turned over to OASIS) to develop a web services standard for building automation called oBIX. And the LonMark association is developing a web services standard for LonWorks systems. The ASHRAE standard has already completed its first round of public comment and appears to be well on its way to acceptance.

The ASHRAE Web Services Standard is being developed by the BACnet committee because that's the committee that deals with building automation. An advantage of this sponsorship is that when adopted, the ASHRAE standard will also become an accepted ANSI and ISO standard. A disadvantage of this sponsorship is that people assume it's somehow tied to the BACnet communication protocol. In reality, ASHRAE is publishing a web services standard which has been designed for use by any building automation system, regardless of protocol. In essence, it takes interoperability to a higher level. ASHRAE is providing a standard to provide interoperability between the BAS and the enterprise applications most needed by energy engineers. The standard can also be used to provide interoperability between different building automation systems. The committee looked at the most common communication protocols in use today, solicited input from multiple BAS manufacturers, and carefully developed a web services standard that was "protocol neutral." This standard is being developed as Annex M to ASHRAE Standard 135. This document establishes standard object types and services, defines the properties of each, describes the path structure that will be used to locate the data, and defines the application program interface (API) which a programmer will use to initiate these web services. Annex M has already been posted for public review and the review comments are being addressed. Committee members are hopeful the revised standard will be approved and published in the fall/winter 2005 time frame.

References

1. Craton, Eric and Robin, David, "Information Model, the Key to Integration," AutomatedBuildings.com, January 2002.
2. American Society of Heating, Refrigerating, and Air Conditioning Engineers (ASHRAE), BSR/ASHRAE Addendum *c* to ANSI/ASHRAE Standard 135-2004, "*Web Services Interface.*"
3. Gray, Earl D., "LonMark Association Sets Sights on Larger Goals," HPAC Networked Controls, July 2003.
4. Open Building Information Exchange, "*History of oBIX*," *www.oBIX.org*.

ABOUT THE AUTHOR

Steve Tom, PE, PhD, is the director of technical information at Automated Logic Corporation, Kennesaw, Georgia, and has more than 30 years experience working with HVAC systems. At ALC Steve has coordinated the training, documentation, and technical support programs, and frequently works with the R&D engineers on product requirements and usability. Currently Steve is directing the development of *www.CtrlSpecBuilder.com*, a free web-based tool for preparing HVAC control system specifications.

Prior to joining Automated Logic, Steve was an officer in the U.S. Air Force where he worked on the design, construction, and operation of facilities (including HVAC systems) around the world. He also taught graduate-level courses in HVAC design and HVAC controls at the Air Force Institute of Technology. Mr. Tom can be contacted at *st@automatedlogic.com*.