Interoperability of building security systems: Protocols and standards

Published on 4 Mar 2010

XML, API and web server offer the ability for interoperability between proprietary software.

Today's facility executive wants to purchase systems and components from various manufacturers and have those components work together as one system. They want "open systems" that allow the manufacturers to offer new features and add value without locking the owner into a single vendor's proprietary protocols. In short, they want Interoperability.

Hirsch Electronics examines the technologies that are now being embraced that allow information to be shared between different manufacturers' proprietary application software. These technologies are available to both the application software supplier and to the IT department that provides the information infrastructure to the enterprise. Suppliers who create these technologies to make it easy for their system to interface and interoperate with another's, add value to the enterprise. These technologies are the API, XML, and web servers.

Let's take an example of a building with an existing XML-enabled access control system. In a standalone mode, a reader transaction might provide the following data for other building and
However, the facilities department wants to use entry/exit reader information from the anti-pass back program to provide better modulation of outside air. Also, a presidential task force is being established to evaluate actual building occupancy patterns of both employees and visitors to reduce cost by setting better working hours and sharing resources among telecommuters. They want to include information indicating which zone the occupant is travelling from and to. This could easily be added (eXtended) by including two new elements in the XML message as follows:
Only the access control system that implemented the anti-pass back control with occupancy counting, the HVAC system, and the presidential task force database application using this information would need to be aware of the "From" and "To" elements added to the XML message. Other non-target applications could retain their legacy interfaces. XML extensibility allows tremendous modifications without incurring any of the re-development costs associated with traditional binary interface protocols.

**Transformation of data using XML**

One of the advantages of XML is that any software developer can define their own tags. However, this can also be a problem if two developers use different tags to mean the same thing. For instance one manufacturer might use the tag "" and another might use the tag "".

Fortunately, there are two solutions. One is the development of industry standards (addressed later). The other solution is to utilize XML's ability to support "Transformations." XML transforms can
be used today while formal industry standards are in development.

XSLT (eXtensible Style sheet Language Transformations), which is also conveniently written in XML, provides a way to "map" or create a template that cross-references one developer's conventions to another's. So, if one manufacturer calls an entryway a and another calls it a, the XSLT transform would make the differences transparent when the two applications exchange data.

---

**XML standards for building systems interoperability**

Today, just about every industry is forming working groups, technical committees, etc. to develop the XML conventions to make it easy to use XML to exchange their industry specific data.

Significant standards activity is occurring in the oBIX (Open Buildings Interface Exchange) technical committee of OASIS (Organization for the Advancement of Structured Information Standards). CABA (Continental Automated Buildings Association) served as an incubator for oBIX before it affiliated with OASIS for a closer association with the IT industry. The ASHRAE (American Society for Heating Refrigeration and Air conditioning Engineers) Standing Standard Project Committee 135, is actively developing an XML interface for their Data Communication Protocol for Building Automation and Control Networks (BACnet). SIA (the Security Industry Association) is also actively developing standards for interoperability.
While official standards for each industry will take time to develop, many vendors will start using the preliminary templates and constructs proposed from the committee work prior to that. The demand is high, the benefit is great, and changes are easy to make. It's eXtensible.

Interoperability allows disparate systems to work together to deliver application solutions not possible with separate, proprietary offerings.

Also of note, Microsoft is spending billions of dollars annually on XML and web services because of the business opportunities these tools open up. Their Office 2003 package stores data natively in XML, and Internet Explorer is XML enabled.

Many software companies are developing new applications, including shareware and freeware, to simplify creating and using XML documents.

The interoperability vision

Customers are now free to choose individual building control systems. As long as they choose systems that are truly open - like those based on XML and APIs with SDKs and professional services available - they will enjoy interfaces that are flexible, customer controlled, and yield a strong ROI.

In the not-so-distant future, end-users and systems integrators will be able to obtain off-the-shelf interoperability applications for specific manufacturers. Those applications will provide standard templates and mapping tools to join the silos of information that exist among the various systems...
within the enterprise. New solutions will emerge that result from interoperability among two or more applications.

Key terms defined

Users want to purchase components from various manufacturers and have them work together as one system

Application Programming Interface

An Application Programming Interface - or API - is essentially a set of conventions that allows a third party application to interoperate with the target application. The third party program need not know the details or native programming concepts of the target application.

Without an API, it can be virtually impossible for programmers or developers to create an interface that connects to or controls any aspect of the target application. With an API, interoperability is readily available to skilled developers to the extent the manufacturer exposes their functionality. The responsibility for providing an API belongs with the application software vendor.

The Software Development Kit
Ideally, a third party programmer or the end user can develop interfaces to a target application without dependence on the supplier. It is incumbent on the application software supplier to provide documentation and sample code to facilitate development of interfaces by others. The API, the documentation, and the sample code collectively make up a Software Development Kit (SDK). Without an SDK, an API can be difficult or impossible to implement.

**eXtensible Markup Language**

eXtensible Markup Language (XML) is fast becoming the worldwide standard for exchanging data because it is human readable, self describing, flexible, platform-independent and widely supported.

XML is a simple, very flexible text format derived from a language originally designed to meet the challenges of large-scale electronic publishing. It is written in plain text, usually English, making it easy to read, understand, and implement.

XML is also a data format. Like other common data formats (such as CSV, .dbf, .mdb, etc.), XML can be used in programs for calculations, sorts, queries, reports, displays, mergers of data, et cetera.

One of the advantages of XML is that any software developer can define their own tags
XML is not a programming language, per se. Nor is it a protocol. By itself, it does nothing. However, when data is stored and communicated in XML's text based format, programmers gain a powerful tool for the development of interoperability between applications.

**Binary Protocols vs. XML**

Today, most controllers communicate with devices or other controllers using a binary protocol. So, integration between two vendors means one has to acquire the binary protocol of the other and adapt to their proprietary, ever-changing rules. Industry standard protocols have overcome this problem to some extent.

Binary protocols (based on ones and zeros) have been around for decades. Although computers still talk in ones and zeroes, the gigabit Ethernet speeds available today make it practical to use human readable XML text for intranet and machine-to-machine (M2M) communications of data. After all, the Internet is built upon TCP/IP using HyperText Transport Protocol (HTTP), which is optimized for "text."

More importantly, binary protocols are not extensible. Once a binary protocol is established, it is not easily changed. Whether to describe the data on the magnetic stripe of your credit card or to define communications between controllers, a change would probably mean reissuing all credit cards, and updating or replacing controllers. With XML, another "element" can be added as easily as another item can be added to an outline in a Word document.