TRIDIUM™
revolutionary://software.solutions™
3951 Westerre Parkway, Suite 350
Richmond, VA 23233
804.747.4771 Phone
804.747.5204 FAX

The Case For an Automation Infrastructure

A White Paper

The Case For an Automation Infrastructure

The proliferation of microprocessors has changed all facets of our world. It has resulted in an explosion of "smart devices" or "embedded systems" in the buildings we live and work in. From systems for control of HVAC equipment, access and security, to electrical metering devices, process control systems and on-site backup generators, to smart appliances and automated food-processing equipment, today's facilities contain a panoply of embedded systems.

While these devices provide great utility in their specific area of application, they pose a great challenge in today's increasingly knowledge-based economy. The problem stems from the fact that these different devices and systems do not communicate with each other, do not communicate with enterprise systems, and do not communicate with the Internet.

Why The Problem Exists

Whether we look at thermostats, facilities automation systems, intelligent appliances, security and access control systems, fire and life safety systems, medical devices, automated food service devices, utility metering devices, or industrial automation systems, embedded systems have been developed largely without the benefit of accepted standards. They employ a dizzying array of communications protocols, data formats, and software platforms. The result is that the world of embedded systems is highly fragmented, not interoperable, and more complex and expensive than necessary. In short, embedded systems have not been conceived with the Internet, interoperability, and integration with the enterprise in mind.

While it is true that more recent systems follow some of the emerging standards, this has only exacerbated the problem due to the fact that the standards themselves have not been designed to interoperate. In reality, the recent push towards standard protocols has created more "languages" that need to be integrated instead of fewer. And because devices themselves continue to get smaller, less expensive and more focused on a single application, we cannot expect to see the devices themselves solving the problem by speaking many languages simultaneously.

History has demonstrated that interoperability will not be accomplished through the use of any single "standard". There are simply too many that have achieved viable followings. The ability to move to new standards is also problematic for the owner. Economics do not support wholesale replacement of existing, functioning systems. It is simply not practical to replace all existing devices with ones that speak a new standard, no matter how compelling the potential benefits.

What we are left with is the need to integrate the myriad of smart devices without affecting the devices themselves – a solution that embraces the multitude of new standards and the wealth of legacy systems equally.

An Analogy from the PC Revolution

The desktop computer revolution created a standardizing force that allowed software developers and manufacturers to focus on a single platform and set of technologies. It provided a standard methodology for interfacing to printers, networks, storage systems, etc. This standard platform had the affect of isolating (or freeing) application developers from the details of how those devices worked, thereby allowing them to focus on their applications. The result was dramatic acceleration of application development in the software market.

This fundamental step has not occurred in the world of embedded systems and "smart devices". There is no standard platform for application developers to build on. Every software application has to be written to deal with the vagaries of any system that it will be used with. This creates a tremendous financial burden on developers and has limited the range independent software applications that can be used with embedded systems.

.

An Automation Infrastructure is the Solution

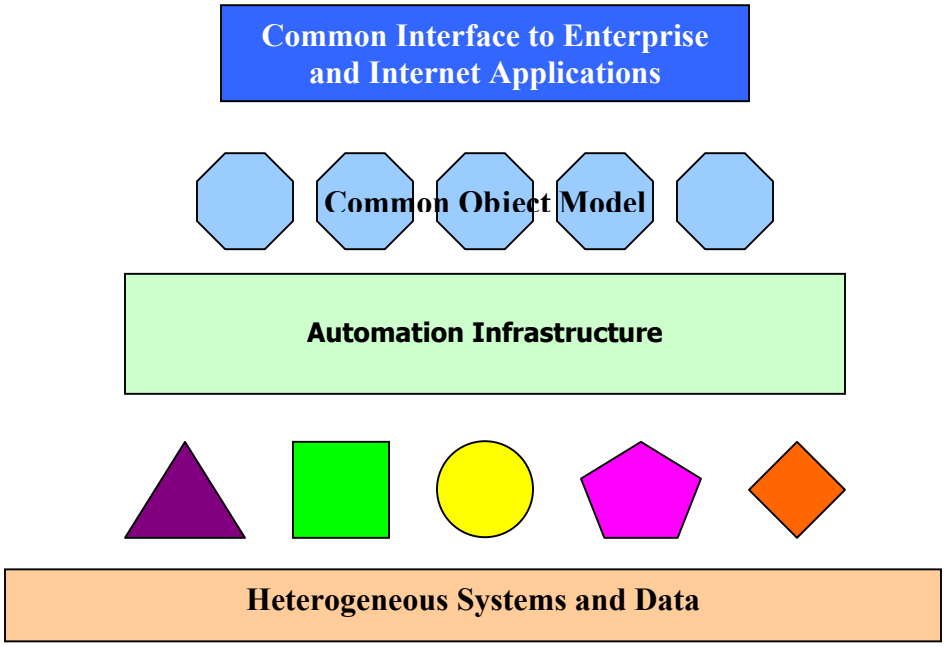
The problem we have described is present throughout the world of embedded systems. What is needed is a solution that brings together all embedded devices, old, new, standard or otherwise, into a single environment that acts to shield the user (and software developer) from the distinctions between different systems. This is the role of an automation infrastructure.

The infrastructure is a layer of software that resides above the individual systems and their specific protocols. Its' role is to provide a uniform method of accessing data from, and issuing commands to, the various smart devices. The infrastructure must be protocol agnostic and vendor neutral supporting all devices equally.

How It Works

The infrastructure enables seamless integration of the diverse systems by introducing a new factor into the equation. That new factor is the concept of a "common object model". Simply put the infrastructure takes the data elements from all of the various devices – inputs, outputs, setpoints, schedules, control parameters, etc., and "morphs" these items into virtual objects. This creates a virtual model of the actual systems – a model that supports all of the functions and features of the end devices. The result of this conversion is a uniform, normalized, database of objects for the user or application developer to work with.

Figure 1: Representation of an automation infrastructure.
The framework takes in heterogeneous data from different systems and creates standardized software objects that represent them. These virtual objects are fully interoperable. On top of this object database the infrastructure provides a set of general services such as a real-time control engine, scheduling, alarming and Internet connectivity.



Infrastructure Provides a Foundation for Value Added Applications

The common object model, which has access to all of the data and actions supported by the diverse systems, can now serve as a foundation for other software applications that provide value to the operation of the facility. Examples include: real time energy data collection and analysis, execution of global control strategies such as schedules and alarming, aggregation and control of energy consumption across multiple facilities, etc..

With these common objects, we can easily build browser-based displays, reports, alarms and even supervisory control logic that works across the multiple systems. The result is true interoperability without the need for users to get mired in the details of competing protocols and without the need to disturb the installed systems. All of this is made possible by the concept of an infrastructure, which forms the foundation for higher level functionality.

It is important to note this approach allows system expansion through the addition of more of the existing smart device type(s) while at the same time enabling expansion with emerging new "best of breed" smart devices and systems previously not used due to incompatible field protocols.

Summary – Why It Matters

Much has already been written about the need for interoperability. The control industry has debated the virtues of various standard protocols for a number of years now. These efforts have created new protocols, but no comprehensive solution to the interoperability issue.

Today, the needs of our increasingly information driven economy demand seamless operation across diverse systems. The challenges unleashed by energy deregulation require enterprise-level applications to be able to interact with diverse systems for applications such as metering, demand management, curtailment programs, real time pricing and procurement. These applications demand a solution that works not only with new systems based on emerging standards but also with the huge installed base of legacy systems. An automation infrastructure provides the solution to these challenges.

The great diversity of systems also makes it difficult for facilities personnel to manage their facilities. Owners often purchase (or inherit) systems from different manufacturers. The complexity of needing to use a variety of manufacturer-specific software applications to interact with different systems often results in systems being ignored or forgotten over time.

The rapid advance of technology has created a situation where even device manufacturers themselves cannot keep up with IT technology and the resulting requirements to upgrade their products. It is no longer feasible for manufacturers to attempt to try to build all of the IT-level functionality required into proprietary hardware and software.

Similarly, the continued proliferation of new standards has created a situation where manufacturers cannot develop for all of them. They typically choose to support one major standard or the other, but not both. This limits market access and customer choices. An automation infrastructure provides the solution to these challenges.

In each of these cases and automation infrastructure is the solution – a framework that allows companies to easily build applications to access and control smart devices of all types over the Internet regardless of protocol, operating system or age.