What is Intelligent Device Management?

NET+Works® Integrated Systems Architecture Saves Valuable Development Time, While Future-Proofing Application Design

White Paper

Abstract

To manage devices over a network, embedded software engineers must build products using a variety of communications protocols, so their network-attached applications meet the needs of their customers. However, as methods of network communications vary, today's time-consuming methods for implementing variables separately for each protocol must change. These methods hinder product development and oppose the "future-proofing" of applications. To meet embedded engineering needs, Digi offers NET+Works with an Integrated Systems Architecture, which uses an embedded database to allow variable implementation or changes to be reflected in all deployed protocols. The result is greater programming ease and flexibility, and faster application development.

Why Manage Devices Over the Network

Today, an increasing number of devices, such as printers and copiers, Ethernet-attached security cameras, retail point-of-sale systems, building access controllers and many more are being attached to corporate and commercial networks. The network and all the value-added services associated with it are increasing the functionality of what were once islands of standalone applications. A device designed with Ethernet/Internet connectivity can communicate with databases, PCs, people and with other devices. This allows a previously unknown level of efficiency and productivity, both for people and businesses.

For example, when a device is network-attached, its status and the information it is gathering can be monitored over the Internet through a web browser. Additionally, a device can be programmed to automatically respond to an event: a networked security system may feature Ethernet-attached security cameras that automatically notify security personnel or alarm companies when doors open after hours, or when motion is detected, eliminating human error or oversight; or, a networked heating sensor can be programmed to shut down an HVAC system if a temperature exceeds a certain pre-defined limit.

Once applications are deployed in the field, existing device networking functionality also enables personnel to diagnose device status via a local network or the Internet. For example, company personnel can use a web browser from a remote location anywhere in the world to upgrade (in real-time) an in-store point-of-sale system's software, and Ethernet-attached airline ticket printers can automatically send warning emails to administrators when their paper or toner levels are low. Such diagnostic and predictive maintenance functionality maintains devices' optimum performance levels while reducing downtime, critical to day-to-day business operations and profitability.

This kind of connectivity, however, raises entirely new challenges, because once a device is attached to a network, it must be managed: it must be installed, given an IP address, configured, monitored, etc. Of particular relevance to design engineers, methods of communicating information over a network vary: in the 1980s, for example, applications primarily communicated using the Telnet protocol (used in communicating over TCP/IP networks and later the Internet). SNMP (Simple Network Management Protocol) versions 1 through 3 were developed in the 1980s and 1990s and are currently used to communicate device status over networks; and HTTP (Hypertext Transfer Protocol), is today a popular and standard protocol enabling HTML pages to be sent back and forth across networks and the Internet, allowing users to manage a device through a web browser.

What protocol is deployed depends on which protocol a customer uses for their networks. Most often, a device is built and designed to handle all of the above protocols, as well as others, to meet the widest market demand and sell as many devices as possible.

As shown above, not only do customers' needs vary, but these communications protocols are also evolving. What are popular protocols today will almost certainly change tomorrow. Future networking protocols deployed at the device level may include XML (Extensible Markup Language) and SOAP (Simple Object Access Protocol), but no one can predict the future with any great accuracy. Device management technologies, meaning the actual communications protocols and methods people want to use over the network, change over time. Many of these changes occur because system administrators want to manage their networks differently and with more sophistication, as better technologies emerge. Electronic product OEMs and their design teams therefore have to adapt their product offerings accordingly.

What About Variables?

Now that we know that most devices have more than one way of being managed, it is important to understand how variables impact a networked product's design. A variable is simply a symbol or a field in a line of code that stands for a specific value. In the case of the networked heating sensor mentioned above, a variable could be "80 degrees Fahrenheit," the temperature at which the device "knows" to turn itself off, or to perform any other action it was programmed to do in response to registering that temperature. So, rather than entering data directly into a program, a programmer can use variables to represent the data.

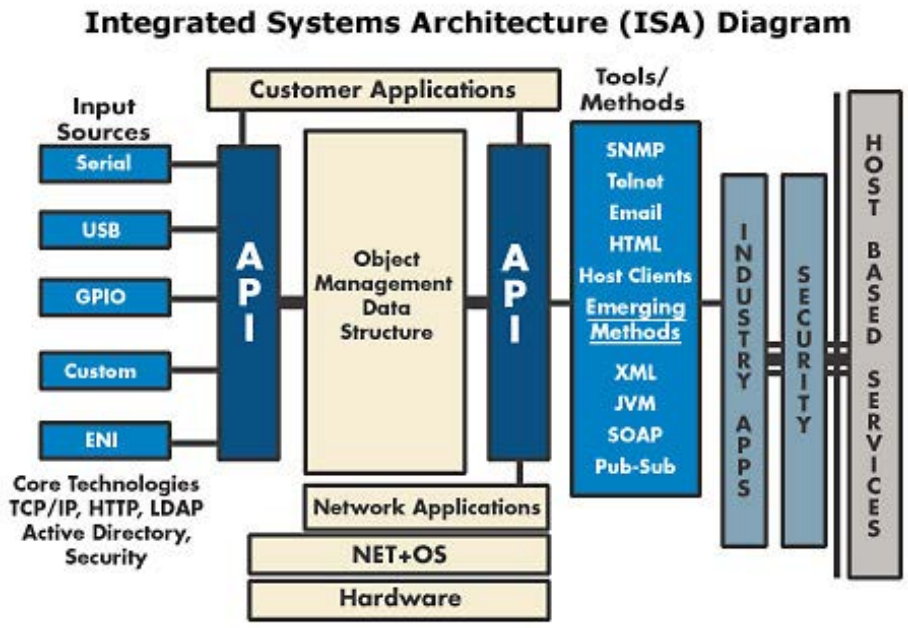
Most devices today are built with Telnet, a version of SNMP, and an HTTP server. Currently, most embedded software engineers implement all of these into a single application is by writing three independent code paths for each protocol. The problem, from the perspective of efficiency, is that if a programmer wants to update or change a variable in one protocol, such as SNMP, the changes made will not show up in other protocols: in this case, in Telnet or in the HTTP server. The programmer then has to make all other code path changes separately. The use of the ISA greatly simplifies this problem.

Clearly, what the market needs is a mechanism that allows programmers to define variables and the terms for them in such a way that any application program will be able to understand it. In other words, software engineers can clearly benefit from a common set of global variables such that if a programmer makes a change to a variable in one protocol, similar changes are made automatically to all other protocols that communicate data related to that variable. The market needs a technology

that "ties" protocols together, from an access and programming perspective. Understanding this, Digi offers NET+Works, a suite of embedded networking software that features an ISA. The ISA allows management variables to be accessed from any communications protocol (i.e., Telnet, SNMP, HTTP, etc.).

Decouple Applications from Your Data
The NET+Works Integrated Systems Architecture (ISA) saves software engineers valuable development time while future-proofing application design to allow for new networking technologies.

In the ISA, Input Sources and Tools/Methods are decoupled from the data structure. To add a new source or tool, the programmer need only interface it to the ISA API. Therefore, the programmer can compile in only those sources or tools needed, and add upgrades or new sources or tools even after the design is complete and the device has been deployed. Customer application software is similarly transportable to new ISA-based device designs. Because the data structure is common, normally incompatible tools have "effective" compatibility: such as remote XML access to SNMP objects, or vice versa.



The Global Store

In the rush to get to market, many embedded engineers often purchase tools from several vendors – for example, an engineer may purchase an embedded web server from one vendor; an SNMP agent from another vendor; and possibly Telnet from a third vendor – and then they have to write the code that allows them to access device management variables for each, and they try to make them all work together. With Digi's ISA, a "global store" model is deployed. The global store is an embedded database of management variables: system variables that reflect configuration values, and values that reflect the state of the device.

For example, suppose a device is designed to be accessed and managed using SNMP, Telnet and HTTP. An engineer would define an SNMP MIB (Management Information Base – in SNMP terminology, a MIB is a database of objects that can be monitored by a network management system) that defines how you control the device and how you get status from it. An engineer would use the tools that come with the NET+Works (SNMP MIB Compiler and Code Generator) to convert the MIB into a set of management variables. Once the MIB has been converted, a person can access the management variables through an SNMP MIB browser. Because the MIB objects are now in the global store, programmers can use them in Web pages, too, allowing someone to deploy another device management interface. Additionally, an engineer can create Telnet menus that access the variables from yet another device management interface. With the ISA, the communications protocols are integrated together, saving the programmer time, while the customer gets their choice of what device management interface they want to use for their network.

And not only does the ISA provide value today: in terms of future-proofing an application, consider JVM (Java Virtual Machine) as an example of how extensible an architecture NET+Works is. While JVM is not widely deployed in embedded networking projects today, conventional industry wisdom suggests that it, or something very much like it, will be deployed in the near-to mid-term. By providing methods for the JVM to use the ISA APIs (Application Programming Interfaces), the Java program will be able to be written to access and control all the variables that were previously controlled by C programs. With the ISA, the same approach can be used for whatever programming language or protocols that emerge in the future.

Therefore, from the above one can see that through the ISA's global store model, when a variable is changed using one protocol, the new value is set in the global store and that new value is visible to the other protocols. This way, the NET+Works ISA allows software engineers to save valuable product development time.

Device Management: The Road to Widespread Adoption

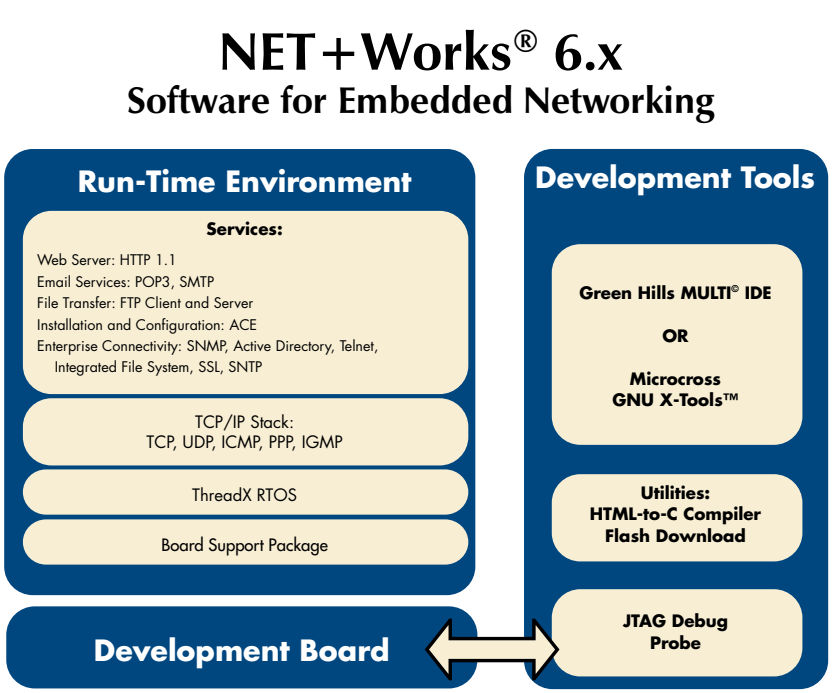
As network-connected devices proliferate, engineers must build each new application with sophisticated configuration, control and management capabilities. In other words, engineers need to design future-proof architectures, and plan for a device-centric, extended web today. But how will a device-centric network unfold? There are four critical enabling factors needed for this transformation to prevail:

Low Overall Cost

Low cost is critically important and is a primary obstruction to deployment and innovation. Cost points include cost of silicon, development tools, software (particularly royalty-based offerings), engineering costs to integrate multiple networking components, and other costs.

Ease of Installation

The ease of device installation, particularly the ability to configure a device remotely, will be a necessity in the next three to five years. Technology is already evolving toward this trend with protocols such as DHCP (Dynamic Host Configuration Protocol), LDAP (Lightweight Directory Access Protcol), UPnP (Universal Plug and Play) and JINI, and almost certainly will impact device networking. And as device manufacturers begin adding these device management protocols to their applications, the NET+Works ISA becomes more and more useful, and will save countless hours of programmers' time, as ultimately they will have to add support for these protocols into their devices.



Security

As an ever-increasing number of intelligent, networked devices connect to corporate LANs and the Internet, new issues and concerns emerge for dealing with device security. Applications used in distributed environments, for example in industrial applications or in retail systems, are becoming critical to the day-today operation of some enterprises. Leaving these devices exposed to potential accidental changes, to sabotage, or to eavesdropping will become too great a risk for most businesses. But, as engineers examine such risk, they will have to make some determinations: at what price do they want security? What are they trying to protect, and from whom? What are the greatest security concerns for device connectivity? What are the performance issues associated with adding security to a networked device? What are the security risks versus an application's Bill-of-Materials (BOM) costs? For devices, "just enough" security means implementing limited but effective security elements, like secret key cryptography, and using them sparingly.

Securing a connected device presents an unusual challenge. Whereas a PC is loaded with resources like memory and computational power, a device that has been retrofitted for network access – such as medical monitoring equipment or a point-of-sale system – has few spare resources. So, how should design engineers secure a resource-limited device without increasing its resources and cost?

Distributed Processing

To build larger, more sophisticated service applications (all of which depend on real-time data) intelligent processing is moving from a centralized computing model (i.e., client-server) to a distributed model (i.e., peer-to-peer communications), where devices communicate with each other at the edge of the network, without human intervention. Real-time or near real-time data

access is a significant factor in creating value in a network-connected device. As networking is driven down to the device level, new technologies for the delivery of services will evolve, that we cannot anticipate. Devices and businesses must be prepared, to the best of their ability, with appropriate technologies in order to do business in this emerging business environment. Digi has released NET+Works 5.0 with the Integrated Systems Architecture to save software engineers valuable development time, while helping to future-proof application design to accommodate emerging networking technologies. To learn more about Digi, NET+Works or the Integrated Systems Architecture, or to order a NET+Works evaluation kit, please see the contact information below to reach your local Digi representative.

Summary

As networking is driven down to the device level, new technologies for the delivery of services will evolve that we cannot anticipate. Devices and businesses must be prepared, to the best of their ability, with appropriate technologies in order to do business in this emerging business environment. With NET+Works technology and ISA, software engineers can save valuable development time, while helping to future-proof application design to accommodate emerging networking technologies.

91001181
D1/1115

DIGI INTERNATIONAL WORLDWIDE HQ
877-912-3444 / 952-912-3444 / www.digi.com

DIGI INTERNATIONAL FRANCE
+33-1-55-61-98-98 / www.digi.fr

DIGI INTERNATIONAL JAPAN
+81-3-5428-0261 / www.digi-intl.co.jp

DIGI INTERNATIONAL SINGAPORE
+65-6213-5380

DIGI INTERNATIONAL CHINA
+86-21-50492199 / www.digi.com.cn