Internet-based Factory Monitoring

Alfred C. Weaver
Department of Computer Science
University of Virginia
151 Engineer's Way
Charlottesville, VA 22904
weaver@virginia.edu

Abstract - **Monitoring a factory remotely over the Internet introduces a variety of issues: how to capture the factory's data in a web-accessible database, locate the data from anywhere in the world, display it locally using a web browser, protect private data during transmission, and shield it from unauthorized viewing. Solving those problems from scratch would be difficult, but fortunately that is not necessary. Using techniques developed for electronic commerce—including universal data access, a ubiquitous programming environment, data security and user authentication—all of these requirements can be met. We discuss these techniques, then show how they are used in combination to create a data monitoring system for a simulated petrochemical plant.**

I. REMOTE MONITORING

Suppose the year were 1970. If you built a chemical plant in Santiago, how could you monitor it from Tokyo? You would have been faced with a formidable array of problems that would have been extremely difficult and expensive to overcome, as summarized in table 1.

Table 1
Problems and Solutions in the 1970s

Problem	1970s Situation
data representation	different computer vendors used different representations; so use the same manufacturer's computers everywhere in the enterprise
data sharing	for long distance communication, only file transfer was practical
communications	leased telephone lines had high error rates and slow transmission speed
real-time control	don't do it; error rates were too high, latencies too long
security	keep everything secret
programming	use the same programming language everywhere

In contrast, how would you do it today? One approach would be to base the data architecture on the public Internet and the World Wide Web, because collectively they support global data access, exchange, and display. Our premise is that factory automation (FA) shares many needs and requirements with electronic commerce (EC), and that several solutions developed for EC are reusable in an FA context. In this paper we look at four FA requirements—universal data access, ubiquitous programming, data security, and user authentication—that are already supported by extant EC techniques.

II. A SIMPLIFIED ARCHITECTURE FOR INTERNET-BASED MONITORING

While highly simplified, one can conceptualize an Internet-based remote monitoring architecture as shown in Fig. 1. On the factory side, sensor data are programmatically scanned and stored in a database and updated at rates that are appropriate for each individual data type. This database is then made accessible through a web server that will allow (authorized) remote entities to read database entries.

In support of our premise that EC provides some ready-made support for FA, table 2 lists some of the issues that naturally arise in trying to achieve remote factory monitoring, and shows how these generic requirements are already provided by electronic commerce components common to the Internet and World Wide Web.

Table 2
FA Monitoring Issues and Solutions Borrowed from EC

FA Monitoring Issue	Solution Borrowed from Internet-based EC
finding the factory's data from anywhere in the world	the web server's URL (Universal Resource Locator) is unique, permitting unambiguous access to the factory's web server
transporting the data	HTTP (HyperText Transport Protocol) reliably transports data between the server and client
routing the data	IP (Internet Protocol) automatically routes data worldwide
displaying the data	HTML (HyperText Markup Language) provides a common vocabulary for machine-independent data display
Understanding the data	XML (eXtensible Markup Language) allows semantic tagging so that the meaning, as well as the value, of data elements is understood
data access and exchange	wired and wireless Internet access is supported worldwide
programming differences	Java programs are interoperable across all vendor equipment
lookup services	Jini is a service that self-discovers capabilities and functionalities in adjacent areas
privacy and security	encryption renders pirated data useless to the thief
authentication	digital signatures authenticate users and commands
reliability	firewalls and proxy servers help insulate the web server from malicious attack and infiltration
audio/video	webcasting enables the transmission of audio/video bitstreams in addition to conventional text and data

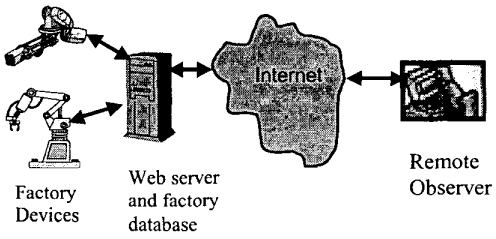

Fig. 1. System Architecture for Remote Monitoring via the Internet

III. REAL-TIME DATA MONITORING EXAMPLE

One example that illustrates the components of a remote monitoring system is the real-time flight information database shown in Fig. 2. Aircraft flying within the U.S. electronically report their flight parameters (heading, speed, altitude, position), and these data are collected in a database attached to a web server. Users access a public portal at *www.trip.com* and download a Java applet that provides a customized user interface to the database. When the user enters an airline name and flight number, the Java applet transmits that information to the web server, retrieves the flight parameters, and converts the raw data into a customized display appropriate for the data type (e.g., a compass to display the heading, an airplane icon on a map to denote geographic position).

Clearly, this web system is analogous to web-based factory monitoring. Whereas this system records real-time flight parameters, the factory monitoring system would record real-time process parameters. Where this system displays raw data in a user-friendly way (e.g., speed on a speedometer), the Java-based factory monitoring system would display process data in a format appropriate for that data. Both systems would utilize: a unique URL for locating the data; HTTP over IP for reliably moving the data; XML for encoding data semantics and values; Java applets to control user interaction and to create data-specific displays; and worldwide access via the Internet and WWW to the database.

Because this example involves only monitoring of public data, neither data encryption nor user authentication were used. For factory monitoring, however, encryption would be used to keep the monitored data private, and digital signatures would be used to authenticate the identity of viewers. These are discussed in the sections that follow.

IV. REAL-TIME CONTROL EXAMPLE

In times past when electron microscopes were rare, Oak Ridge National Laboratory allowed academicians to use its microscope by reserving time and then physically traveling to the Oak Ridge , Tennessee site to use the machine at the scheduled time. In modern usage, a scientist still reserves time on the microscope and mails the material to be scanned,

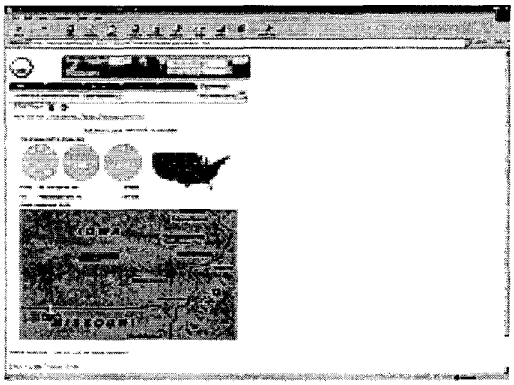

Fig. 2. Real-Time Aircraft Position Monitoring over the Internet

but by utilizing the Internet and a Java applet he or she can now run the microscope remotely from the desktop.

As shown in Fig. 3, the Java applet provides a user-friendly interface that allows the user to shift the sample's position (up, down, left, right), change the magnification, change the focus, and retrieve a real-time low-resolution (256x256x8) gray-scale image from the microscope. Once the control parameters are properly set, the user can then download a high-resolution (1024x1024x14) gray-scale image to a local file.

Here again the analogy to factory control is clear. The Java-enabled desktop becomes a surrogate for the physical machine's control panel, and commands and data originating from the Java applet are transmitted to the remote web server; there they are relayed locally to the machine being controlled.

V. INTERNET ACCESS

The commercial Internet supports worldwide access via the telephone, cable TV, fiber optic, and wireless connections. The most interesting of these are the new wireless capabilities. Geosynchronous satellites can be used for data distribution as has been done with the DirecPC [1] system developed by Hughes Network Systems. The satellite downlink provides a channel with a capacity near 400 kbps; a conventional telephone connection provides the reverse channel for keypresses and mouse clicks.

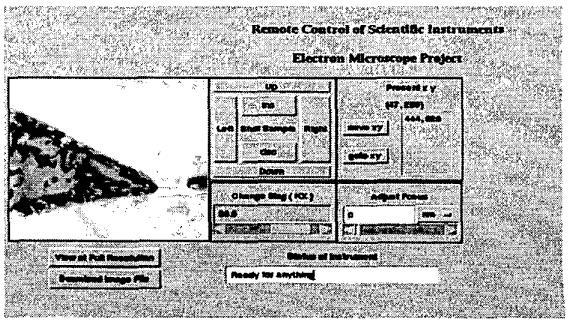

Fig. 3. Remote Control of Electronic Microscope

Although Motorola's Iridium project failed, Teledesic [2] plans to place 300+ satellites in low earth orbit by 2005. If successful, Teledesic will provide both voice and data (Internet) access to cell phones, laptops, and other Internet appliances located anywhere on the surface of the globe (see Fig. 4). Because Teledesic is a data infrastructure that can also accommodate voice, it is ideally suited to support mobile data access worldwide via laptops, palmtops, cell phones, and other mobile devices.

WAP (Wireless Applications Protocol) [3] is an emerging standard that will permit wireless telephones and PDAs to interact with Internet appliances ranging from parking meters and drink dispensers to household devices, automobiles, and factory equipment. WAP-enabled devices are just now appearing in the marketplace. Bluetooth [4] is an emerging standard for short-range (10 m) wireless communications. Bluetooth implements both infrared and RF communication and will provide the physical communications channel among communicating devices. IEEE 802.11b [5] is the new standard for wireless Ethernet; it operates at 11 Mbps and is especially useful for connecting mobile laptops to a resident computing infrastructure using a low-cost PCMCIA adapter card.

VI. JAVA

Java is a programming language designed from the outset to be portable, thereby minimizing the effort required to port an application from one computer to another. Java borrows heavily from the popular C and C++ languages, and thus is reasonably easy to learn. Java is portable because the language produces an intermediate code (bytecodes) that is then interpreted at run-time, and the run-time interpreter is provided transparently in all modern browsers.

Suppose that the purpose of a Java function was to compare the last-known values of pressure readings against their respective threshold values and to signal an alarm if any values are over-limit. That task could be accomplished with code such as this:

```
// Check temperature limits
public static bool Alarm
 (Pressure[], Threshold[]){
   bool AlarmStatus = false;
   for (i=0; i<Pressure.length; i++;)
      if (Pressure[i] > Threshold[i])
        AlarmStatus = true;
   return AlarmStatus;
}
```

VII. DATA SECURITY VIA ENCRYPTION

No person would use the Internet for electronic commerce if he thought that private information such as credit card numbers or bank account numbers could be viewed or copied by non-authorized persons. Similarly, no company would use the Internet to transmit its proprietary information if it thought that its data could be copied, diverted, or altered

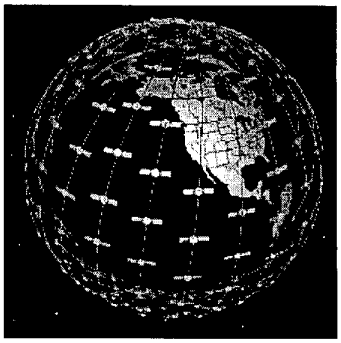

Fig . 4. Teledesic Low Earth Orbit Satellite System

enroute. The solution to data security is encryption, in which data to be transmitted over the Internet is first transformed (encrypted) in such a way that anyone who intercepts the encrypted data is powerless to reverse-transform (decrypt) it, whereas an authorized receiver of encrypted data can easily decrypt it.

There are two major classes of encryption techniques in use: symmetric key and public key. With symmetric key techniques, the transmitter passes his original data (plaintext) through an agreed encryption algorithm whose output is a function of the plaintext and the symmetric key (some large number, typically 56 to 128 bits in length), and the value of the key must remain a secret forever between the sender and receiver. When the receiver of the encrypted data runs it through the agreed decryption algorithm using the same secret key, the plaintext is reproduced. Without access to the value of the secret key, an eavesdropper who intercepts the encrypted data could only attempt to decrypt it by trying all possible values of the secret key. A key of length n bits would thus have 2^n possible values, requiring on average 2^{n-1} attempts to discover the secret key by brute force search.

The best known of the symmetric key algorithms is DES, the Data Encryption Standard, first proposed in 1976. Its 56-bit key was adequate then, but the relentless increase in computing power (Moore's Law says that processor power doubles every 18 months) has made that algorithm insecure. Today's safest symmetric key algorithms are triple-DES (which can use either 112-bit or 168-bit keys) and IDEA that has a 128-bit key. The weakness of symmetric key schemes is not in the algorithms (which have been widely studied for 25 years), but in the key distribution. Both sender and receiver must use the same key, and thus even if the sender randomly generates the encryption key, just before use, there must be a way to transmit it securely to the receiver. This problem is solved by public key cryptography (PKC).

PKC was first developed in the 1970s and is now in widespread use. The underlying mathematics is elegant, and the keys can be made arbitrarily long so that the protective power of PKC can be increased whenever necessary. Each entity that transmits or receives data is assigned a pair of keys, one public and one private. The public key is widely distributed and certification authorities assure the binding between an entity's name and its public key's value; the

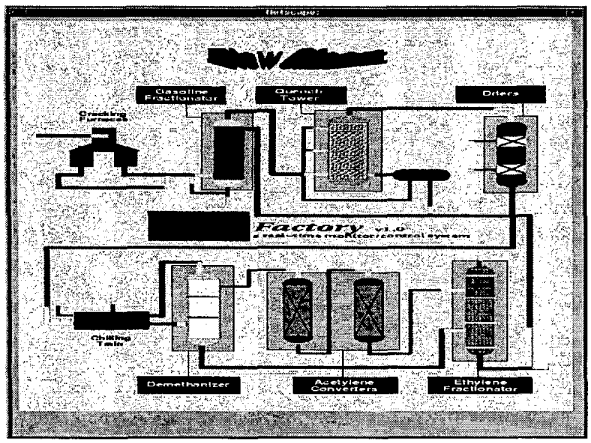

Fig. 5. Process Flow Diagram for Petroleum Distillation Plant

private key must be a secret forever, but the private key is never shared so this is less of an issue than with symmetric key systems.

Let the public key be two numbers (d, n) and the private key be two numbers (e, n), where e, d, and n are very large integers that have special mathematical relationships to each other (see [6] for details). Then for entity A to send a secure message to entity B, A encrypts the message with B's public key (which is well known), and now only B can decrypt it because only B has access to B's private key. While PKC can be made arbitrarily secure by increasing the length of the keys, the downside is that the resulting mathematical manipulations of very large numbers can be quite time consuming.

In commercial practice, these two techniques are combined. The transmitter of secure data first chooses an appropriate symmetric key algorithm, then generates a random symmetric key (SK). The SK is then encrypted using PKC and the public key of the receiving entity, and the encrypted key is then sent to the receiver. The data to be protected is then encrypted using the chosen symmetric key algorithm and SK and then transmitted. The receiver decrypts the first message using its private key; that reveals the SK, which is then used to decrypt the subsequent message. This is the basis of Secure Sockets Layer (SSL) [7] that is now embedded in every commercial web browser and server.

VIII. AUTHENTICATION

Authentication is easily achieved using the digital signature [8] component of PKC. In this scheme, the sender encrypts a specific message with its private key and transmits it; if the receiver can recover the expected message by decrypting the message with the transmitter's public key, that proves that the message must have originated from the entity whose public key was used for decryption.

IX. VIRTUAL FACTORY

To show that these components do work together, we created a simulation of a petroleum distillation plant. A web user loads a Java applet, the applet creates the display shown in Fig. 5 that illustrates the process flow diagram for the cracking furnace, gasoline fractionator, quench tower, drier, demethanizer, acetylene converter, and ethylene fractionator. The simulated plant maintains a database of temperatures, pressures, and flow rates for each input and output port in the diagram. By clicking on any of the process components, say the gasoline fractionator, the Java applet creates an enlarged flow diagram labeling the input and output points. Clicking on any of these displays the real-time temperatures, pressures, and flow rates, and allows the user to watch the data change over time as shown in Fig. 6.

X. PUTTING IT ALL TOGETHER

The components of electronic commerce provide a partial solution to the task of supporting remote factory monitoring. Connecting the factory devices to a local web server allows them to export whatever data is useful to observe; worldwide Internet access is provided by wired and wireless means; local data transfer within the factory will be eased by the emergence of WAP, Bluetooth, and 802.11b; data security is supported by modern encryption techniques; and authentication of data viewers is provided via digital signatures.

After monitoring, the next challenge is control. This is a much more difficult problem that must deal with data latency and data loss, control loop timing, and system reliability. While EC has techniques to support these operations as well, they remain a research topic for now.

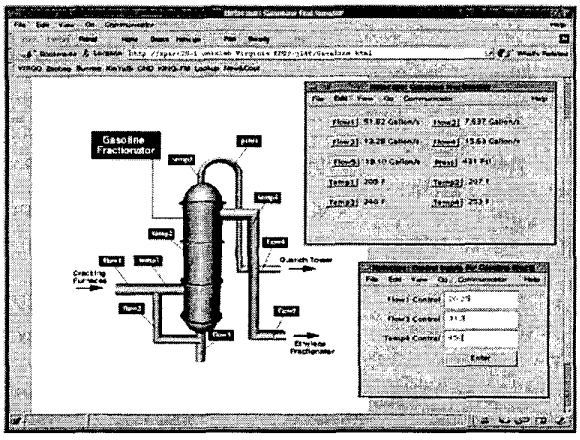

Fig. 6. Real-Time Data Display

XI. REFERENCES

[1] DirecPC, *http://www.direcpc.com/index1.html*
[2] *http://www.teledesic.com/*
[3] *http://www.wapforum.org/*
[4] *http://www.bluetooth.org/*
[5] *http://standards.ieee.org/*
[6] Bruce Schneier, "Public Key Cryptography," **Applied Cryptography**, John Wiley & Sons, Inc., 1996, pp. 31-34.
[7] Secure Sockets Layer, *http://home.netscape.com/security/techbriefs/ssl.html*
[8] Bruce Schneier, "Digital Signatures," **Applied Cryptography**, John Wiley & Sons, Inc., 1996, pp. 483-494.