



How will XML

impact industrial automation?

BY PAOLO PINCETI

Yet another IT trend is set to push plant progress.

To exchange information, most of the Internet applications use files written with hypertext markup language (HTML). We can say, without any overstatement, that it was HTML that made the Internet boom possible.

Tim Berners-Lee created HTML in the early '80s at CERN Laboratories in Geneva, and its main features are simplicity, standardization of data format, and the possibility of defining multimedia documents.

The downside of HTML is that no data reuse is possible. If the same data is to display in both video and print, a user must create two separate documents with the specific format instructions for video and printer.

In addition, HTML documents do not discriminate between actual data and the instructions to visualize or format it.

To overcome these difficulties, ISO 8879, standard generalized markup language (SGML) for the definition of texts and documents, came into being in 1986. HTML is an application of SGML in the sense that it allows only a limited portion of the enormous amount of functions SGML defines.

SGML clearly separates data from the instructions to present it and offers powerful programming tools. Unfortunately, this flexibility represents the main limit of SGML: its complexity.

No browser today or in the future will use SGML. Some companies build their Web sites using SGML code and then convert it to HTML to make the sites accessible over the Internet. Such a conversion eliminates the possible advantages of the SGML code.

A working group of the World Wide Web Consortium (W3C), an organization with the mission to specify Web technologies to guarantee the interoperability of the Internet, overcame the limits of both HTML and SGML with the definition of the extensible markup language (XML).

Fourteen leading companies are involved in the working group (Microsoft, Adobe, HP, Sun Microsystems, and others), so the success of the project is guaranteed.

XML is going to modify all of the usual programming practice for Internet applications. The question we are considering is: "Will XML impact the world of industrial automation in any way?" We can anticipate that the answer will be "Yes, deeply."

Here's why.

NO SANE MANAGER INTERESTED

The structure of automation systems for industrial plants was strictly application oriented until a few years ago: programmable logic controllers (PLCs) for manufacturing and distributed control systems for continuous processes.

The introduction of standard PC-based platforms is canceling this difference by introducing a new paradigm in automation.

The new paradigm control system splits into several levels, according to the operating requirements of the plant:

- i *Level 0* is the base of the automation system, made by conventional and/or intelligent instruments and actuators in the field.
- i *Level 1* represents local control and leverages local control units (LCUs) using different technologies such as PLCs, intelligent I/O, controllers, or PC-based units. The scope of local control is to interface with field instruments and actuate the control logic. The communication with Level 0 is either by conventional I/Os (4–20 mA or dry contacts) or on fieldbus-based networks. LCUs often have local displays for process supervision or data input.
- i *Level 2* is area control (AC), where several LCUs coordinate operations. The AC units normally sit on a PC or workstation network with special attention to the operator interface.
- i *Level 3* is the communication and information system (CIS), where the different functions of the company interact to play their specific roles, from sales to research and from purchasing to financial.

You say to-MAY-to, I say to-MAH-to

These two XML documents describe a single pressure transmitter. Two different programmers wrote the code. The informative content of the two documents is identical, though the data structure and the names of the elements are different.

Data exchange between two applications written by the two programmers is impossible unless they agree on a common document type definition to define the data scheme. To make a software application transportable and devices interoperable, the interface between devices must be standard.

XML documents for a pressure transmitter

DOCUMENT #1	DOCUMENT #2
<pre><device>transmitter <identification> <tag>PT-002</tag> <variable>relative pressure</variable> <manufacturer>XYZ</manufacturer> <model>ABC</model> </identification> <measure> <unit>bar</unit> <value>14.4</value> <quality>good</quality> </measure> <service> <setting date>25.12.99</setting date> <operator>John Doe</operator> <precision>0.45</precision> </service> </device></pre>	<pre><object>transmitter <identity> <tag>PT-002</tag> </identity> <data> <parameter>relative pressure</parameter> <measurement>14.4</measurement> <e.u.>bar</e.u.> </data> <maintenance> <brand>XYZ</brand> <type>ABC</type> <status>good</status> <start-up>25.12.99</start-up> <technician>John Doe</technician> <accuracy>0.45</accuracy> </maintenance> </object></pre>

The practice of separating the automation system into levels was due, in the recent past, to actual technological needs of the devices used for different functions.

Today, when intelligent field devices are used, communication is theoretically possible from the CIS down to the last transmitter or actuator, although no sane manager will ever be interested in the operating data of a flow transmitter or a valve positioner (especially on the weekend!).

Actually the level-based structure of the automation system arises from requirements of the decision chain of the company, as clearly stated in ISA-95.00.01-2000,

Enterprise-Control System Integration Part 1: Models and Terminology.

HTML INEFFICIENT, DIFFICULT

The communication networks used in automation systems are today based on two main techniques:

- IEC 61158, a fieldbus standard for use in industrial control systems, which conducts data exchange at Levels 0/1 and partly at Level 2
- Ethernet, IEEE 802.3, plus TCP/IP for communication at Levels 2 and above.

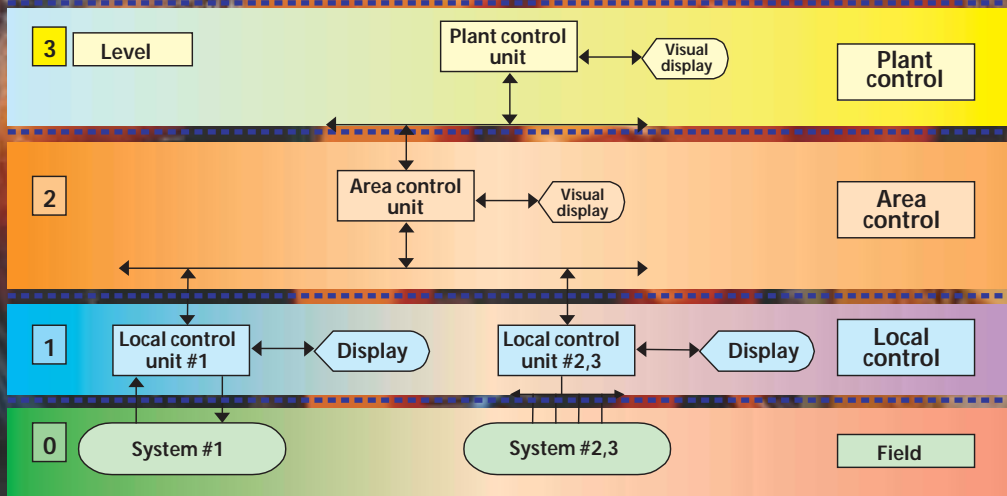
Ethernet is widely used at the CIS level, and this fact justifies the push toward the

use of Ethernet at the upper levels of the automation systems likely to interface with the company CIS.

The information exchange among devices linked with Ethernet is today widely based on the use of OLE for process control (OPC), which allows the high-level applications to access data at lower levels.

OPC is a technique sponsored by a group of 250 members (www.opcfoundation.org). It employs component object model (COM) technology and works and develops in the Microsoft environment, which allows it to integrate documents created by heterogeneous applications.

Generic architecture of an automation system



When network communication is involved, one needs distributed COM techniques. The basic idea of COM is that any software piece is a component, and interactions among components happen through interfaces.

An alternative approach, independent of the Microsoft environment, is common object request broker architecture (CORBA, www.corba.org). Technically, CORBA and OPC are very similar.

The major limit both OPC and CORBA present is that these techniques are specific for process control and do not exploit the progress of general-purpose software applications.

In other words, programmers all over the world wish to use communication techniques based on the "Internet languages" (read HTML), even though they weren't created for industrial applications and are quite poor in terms of efficiency, compactness, and ease of reuse.

XML MORPHS INTO THREE PIECES

The XML strategy to overcome HTML limits is to split large amounts of information into simple elements and collect them in an XML document. A document is nothing more than a data container, where data functions in any application (not only to display it).

Elements label by using a starting and ending tag and can organize into complex structures with father-son relationships. For instance `<name>Paolo</name>` is an element called `name` that has the value `Paolo` and can be father of the element `son`.

In the industrial world, `<transmitter>` can be father of `<pressure>`, `<temperature>`, `<level>`, and the like.

An element can also have attributes: `<transmitter tag=PT01>` assigns the value `PT01` to the attribute `transmitter` tag of the element. There is no fixed rule to decide whether a piece of information is an element or an attribute.

Normally, it is wise to define information as an element if it is germane to several applications or if it has further specifications that can become sons of the element.

An XML document is well formed when the data structure respects XML grammatical rules. An XML parser can read a well-formed XML document even if the data structure is unknown. A further advantage of XML documents is that they contain only ASCII characters, which guarantees full portability to any platform.

Because XML documents contain only data, the information about the data structure and the display commands is contained into two further files:

The document type definition (DTD) defines the scheme of the data. The DTD can

package inside the XML document, or one can request it via an external link.

Extensible style language (XSL) is a style sheet for the document. The XSL can extract data from an XML document (a query) or customize the data for a specific peripheral. XML and XSL together give the same content as an HTML file.

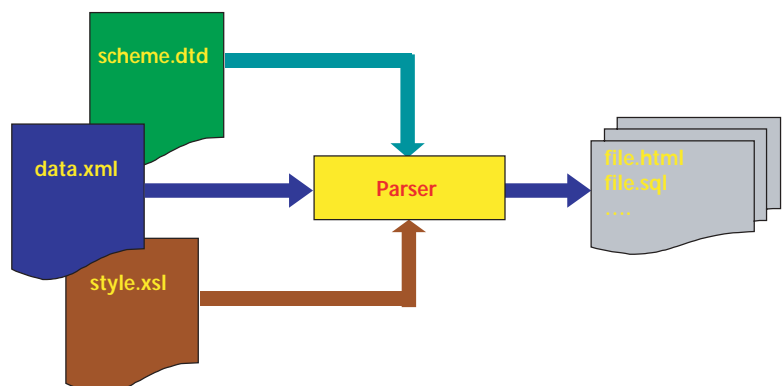
An XML file is split into three pieces that can transmit all together or separately. The XML parser mixes the three files to obtain the final format.

When a parser receives an XML document, it checks the coherence between the DTD and the data to determine whether the document is valid. The DTD is necessary to query a database, as it contains the pointers to the requested data.

XML FOR INDUSTRIAL AUTOMATION

The forces that are pushing the use of XML are outside the industrial automation world. XML is becoming a force on the Internet and in office

XML files and their interactions



Further reading on XML

"Exchanging data with XML: Part 1"

www.isa.org/journals/ic/feature/1,1162,382,00.html?index_userid=&first=101&last=125

"XML reaches factory floor's automation islands"

www.isa.org/journals/ic/feature/1,1162,439,00.html?index_userid=&first=76&last=100

"Exchanging data with XML: Part 2"

www.isa.org/journals/ic/feature/1,1162,407,00.html?index_userid=&first=101&last=125

automation. History suggests the technology will migrate from CIS and encompass industrial automation in three to five years.

We can easily foresee that in the next years XML will move from the Internet into industrial automation and that it is necessary to prepare now. With regard to communication, the goals of the CIS and IA are identical. Both of them wish to make data transmission faster; make data search more efficient and rapid; and transport software applications on different platforms.

XML can help users to reach these goals.

XML'S ADVANTAGE

The advantages of XML with respect to HTML are clear: The data structure and style (DTD and XSL files) are transmitted only once during system initialization, and only raw data is transferred during operation (XML file).

Moreover, as data is clearly an element of the XML file, data retrieval is much easier. The emerging architecture for industrial automation derives from the CIS architectures and it is based on a network protocol using Ethernet plus TCP/IP, a switched-Ethernet configuration, a client/server structure for data sorting and service request, and XML/XSL for data exchange.

Below the surface, XML hides a problem that may make it inefficient in industrial automation if not solved. A programmer in the DTD can freely define the data structure of a XML document, as long as he adheres to XML rules.

To avoid the conflicts that might arise inside a document with multiple DTD (homonyms, nesting structures, and the

like), a unique uniform resource identifier identifies the elements.

The standardization proposals are checked and validated by the W3C. Several companies and organizations have defined standard DTDs for different environments, including medical data, insurance, mathematics, and financial markets.

The definition of a dictionary of DTDs for industrial automation will open again the discussion that was born with fieldbus regarding *device profiles*. At the very beginning of fieldbus history, one of the first ideas to guarantee interoperability among the devices connected to a digital fieldbus was to define standard device profiles that address all data that can be transmitted from/to a given type of field device.

The profile definition took several years, and today these profiles exist for some fieldbuses. Most programmers, however, do not like to use them and prefer to operate directly on the devices' internal variables.

Surely profiles (and DTDs as well) constrain creativity, but they make interactions between different devices easier and safer. Moreover, the new trends in industrial automation suggest that one download the control sequences or loops into the field devices using the IEC 61131 standard languages.

The network protocols and languages must support this kind of transmission, and tools should be available to support this task. IEC 61499 will use XML to define the new block model.

XML is gaining popularity day by day among programmers in the world of com-

Terminology

IEC 61499 standardizes aspects of the application of software modules called *function blocks* in distributed industrial process measurement and control systems, particularly those utilizing the *fieldbus* standard, IEC 61158.

IEC 61131 standardizes the programming interface for industrial control programming. It harmonizes the way people design and operate industrial controls and allows people with different backgrounds and skills to create different elements of a program.

IEC 61158 standardizes a LAN such that it can interconnect the control, sensor, and actuator components in the field. The LAN is an all-digital, serial, two-way communications system at the base level in the hierarchy of plant networks. It ensures interoperability.

munication and information systems and is likely to replace HTML in Internet-based applications.

And because industrial automation follows information technology trends with only a few years' delay, XML will soon be the programming language for data exchange (at least at higher levels).

The main advantage of XML for industrial automation is that data transmits separately from style info and data structure. This makes database sorting and data transmission easier and faster.

On the other hand, we will need to define standard document type definitions for the most common devices and functions needed in industrial automation.

The nearly abandoned work on *device profiles* has to start again under a new and wider perspective. The sooner, the better

IT



Behind the byline

Paolo Pinceti, Ph.D., teaches industrial measurements and control at the University of Genoa's (Italy) Electrical Engineering Department. His research focuses on digital control systems using fieldbus technology. He has been the adviser of the ISA Student Section of Genoa University since 1998.