

INDUSOFT'S USE OF XML AND SOAP

Introduction

EXtensible Markup Language (XML) offers interoperability using a flexible, open standard giving us new ways to access and deliver data. In industrial automation, XML will be a driving force in data exchange, manipulation, and transmission. XML in collaboration with HTML and SOAP will provide the foundation for Web-based Human Machine Interfaces.

Indusoft Web Studio imports and exports recipes, reports, and real-time data into XML format. Indusoft Web Studio brings XML support to the plant floor, allowing for dynamic, autonomous, and automated data exchange between applications. XML makes E-automation a reality by allowing plant-floor dynamic data in XML format to be accessed easily and incorporated in other software products, such as Microsoft's BizTalk and Web applications.

Why XML?

The success and popularity of XML stems from its ability to be "self-describing." For example, every piece of XML data contains information that describes exactly what it does. In turn, XML simplifies communications between nonhomogenous platforms and allows the vision of the virtual supply chain to become a reality. XML is not new, but started in 1996 and has been a World Wide Web Consortium (W3C) standard since February 1998. Before there was XML, there was SGML, developed in the early 80s, an ISO standard since 1986, and widely used for large documentation projects. HTML, which has evolved since 1990, is a widely popular for display data on the Web. The designers of XML simply took the best parts of SGML, and guided by HTML, produced XML.

XML came about because of the need to represent and manage data on the Web. HTML, GIF, and JavaScript™ have become the standard for visual display and user interface on the Web. These standards allow a page to be created once and be displayed at different times by many receivers. Now, the need exists to represent and manage the data between different platforms. For example, HTML does not provide a common way of representing data so software can search, move, display and manipulate data. XML and XML extensions have become the standard for Web-based data transactions such as managing e-commerce business processes that connect the enterprise with its suppliers and customers. In fact, companies have begun developing XML-based voice files to deliver information to wireless phones. Detroit-based General Motors Corporation has claimed that they will be providing OnStar XML-based wireless voice files to deliver weather forecasts, news, sports scores and stock updates to wireless phones built into 30 models of their premier cars. According to Daryl Plummer, an analyst at Gartner Group Inc. in Stamford, Conn., if a company strategy is to deliver Web content to devices, then they need to be using XML and XSL (Extensible Stylesheet Language) because 85% to 90 % of companies are committed to doing it that way.

XML is a subset of SGML optimized for delivery over the Web defined by W3C. XML improves upon some of the deficiencies found in SGML such as legacy requirements and features that make generating SGML-based documents easier, but complicates its use on the Web

XML ensures that structured data will be uniform and independent of applications or vendors. The goal of XML is to provide:

- A method for putting structured data in a text document

XML is used to represent highly structured hierarchical information. Often times, we have data that is generated by an application such as Excel, Word, which can be viewed in either binary format or text. In text format, we can view the information without requiring the application that produced it, but often times it is inadequate because it loses the structure of the information. XML is a set of conventions for designing text formats for such data, in a way that produces files that are easy to generate and read by a computer. It avoids common pitfalls, such as lack of extensibility, lack of support for internationalization/localization, and platform-dependency.

- A format that is as easy to understand as HTML

Similar to HTML, XML uses tags and attributes to describe data. Tags indicate the characters that delineate a particular markup element. For example, a FONT element is represented by a pair of tags, the *start tag* and the *end tag* . The text being marked up appears between the tags and is called the content. An attribute is the name of something that qualifies an element. Most attributes have either a discrete or infinite set of values (or no value at all). Attributes always appear within the *start tag*. For example, content goes here. . In other words, FONT element's SIZE attribute is set to the value of 3. HTML markup describes what each tag and attribute means while XML uses the tags only to delimit pieces of data and leaves the interpretation of the data completely up to the application that uses it.

An XML document also consists of two parts:

- Prolog – declares element names, attributes, and rules for valid markup of data.
- Document instance – data with markup has one root element with all other elements as children of the root; tree representation.

Other differences between HTML and XML are as follows:

HTML	XML
For display	For data structure
No knowledge of data	Presentation independent
Closed language; Standard	Open Language
Case insensitive	Case sensitive (because of Unicode)
Empty tag like requires nothing special	Tags without content and those which are empty elements must use “/>” notation
White space is ignored	White space, including line breaks, is significant within content

There is a basic set of XML commands that were defined in the XML 1.0, the specification that defines what “tags” and “attributes” are, but there has been a growing set of modules that provide sets of tags and attributes for specific tasks. For example, Xlink defines a standard way to add hyperlinks to an XML file, CSS, the style sheet language and XSL is the advanced language for expressing document transformations. In addition, OPC has defined a draft of extensions specific to industrial automation. For example, the World Wide Web Consortium has developed the template for DTD (Document Type Definition -- files to facilitate the description and definition of XML data structures and tag names and define how tagged elements are nested), but the message tag definitions must be driven by collaboration between vendors and customers in specific industries. Outside of agreed standards, both ends of a transaction must still agree upon the content and definition of the XML message tag. However, one of the nice parts of XML (as compared to SGML) is that it can parse and partially interpret random XML-formatted data without reference to the DTD it conforms to. It is important that when something is put into XML format that it also defines the schema for the data through a DTD.

Using SOAP?

Simple Object Access Protocol (SOAP) is a lightweight protocol for exchange of information in a decentralized, distributed environment. SOAP is a way to use the current Internet infrastructure to enable applications to remotely interact with each other. It is an XML based protocol that consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined datatypes, and a convention for representing remote calls and responses.

For example, imagine a component that lives somewhere on the Internet that implements a notification and generation of a problem report interface. This method would be invoked when an alarm acknowledgement is received. The following HTTP request represents how you would invoke such a method using the SOAP protocol:

```
POST /cgi-bin/problem-report.cgi HTTP/1.1
Methodname: Problem-report
InterfaceName: soap:cdl:com.develop.demos.problem_report
MessageType: Call
Content-Type: text/xml-SOAP

<ProblemReport>
  <Number>0201379368</Number>
</ProblemReport>
```

This HTTP request points to a uniform resource identifier (URI) of /cgi-bin/problem-report.cgi. Since the SOAP specification does not state how a component is activated, it is up to the code behind this URI to decide how to activate the component and invoke the specified method.

The Content-Type header uses a value of text/xml-SOAP. SOAP uses this identifier to define an HTTP entity-body containing an XML-encoded method invocation or response. All SOAP requests and responses must use this Content-Type value to be recognized.

SOAP defines three extended HTTP headers: MessageType, MethodName and InterfaceName. MessageType and MethodName, must be used with every SOAP request. MessageType header defines whether the given HTTP body contains a method call or a method response and must contain the values Call or CallResponse. MethodName header defines the name of the method to invoke. InterfaceName is optional and defines the name of the interface to which MethodName belongs.

How InduSoft uses these Technologies for E-Automation

InduSoft separates the content from the presentation layer to prepare for delivering information to a range of devices. This is the power and beauty of XML; it maintains the separation of the user interface from the structured data. It also allows applications avoid keeping multiple copies of data.

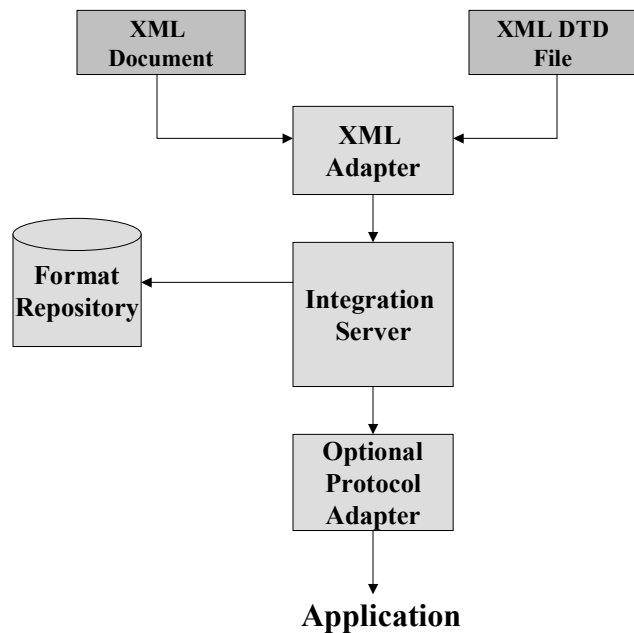
Indusoft Web Studio imports and exports recipes, reports, and real-time data into XML format. For example, InduSoft allows developers to store recipe data in XML format. Recipes, in this case, mean a group of tags that are going to have their values saved in XML and retrieved through the Internet Explorer V5.0 Web browser. The XML parser in Internet Explorer 5 can read a string of XML data, process it, generate a structured tree, and expose all data elements as objects.

First, you must enter the task tab under InduSoft Web Studio. Right click on Recipe, select insert, and the recipe worksheet will be displayed. This worksheet has a checkbox that allows developers to save the recipe file and tag values in XML. Whenever this check box is checked, it is possible to write and read tags values to and from files in XML format, by using the Recipe () function.

The Alarms and Trends functions in InduSoft will be SOAP V1.1 compliant in the next major release. InduSoft plans to have the Alarms and Trends function able to exchange information in a distributed environment using SOAP to do XML-based application-to-application communication.

Using XML to communicate to other packages?

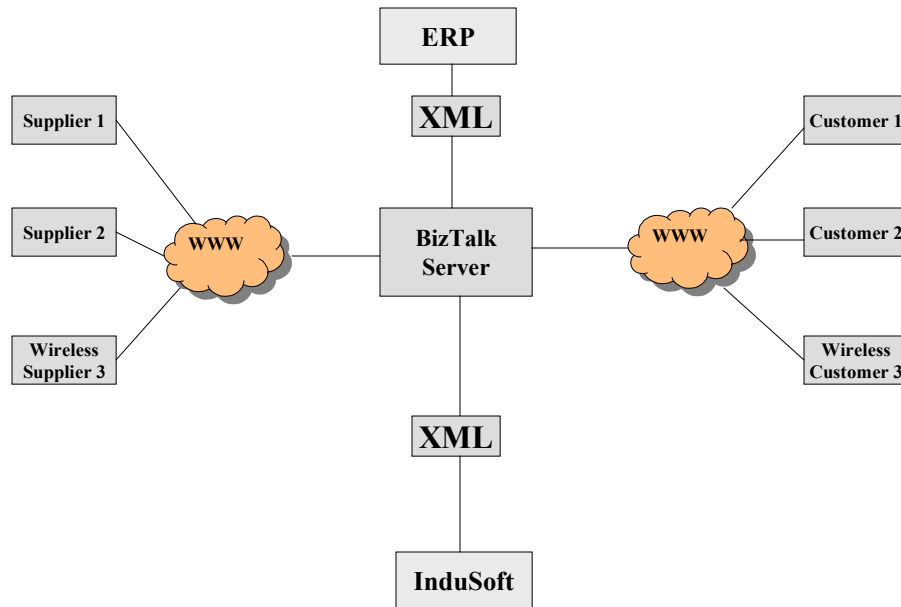
XML is used as a data definition and manipulation language. Because data is being exchanged in the XML format, it can be easily merged from different sources. For example, Microsoft's BizTalk specification consists of a framework description, a repository for cross-industry schemas, and a process for posting and validating schemas (diagram 1).



Biztalk framework is a set of guidelines for how to publish schemas in XML and how to use XML messages to easily integrate software programs together. In addition to XML, the BizTalk Framework takes advantage of standard such as MIME and SOAP. The support for SOAP will allow developers to create applications and services that can be more easily integrated, independent of operating system, programming model or programming language. The BizTalk Framework 2.0 allows organizations to use standards such as Hypertext Transfer Protocol (HTTP) and Simple Mail Transfer Protocol (SMTP) to transmit information over the Internet

The BizTalk server provides business-to-business document interchange and provides a server for routing information between different people based on the specific values. Many leading technology companies like SAP, BAAN, CommerceOne and InduSoft are developing application integration components (AICs). These AICs provides the capabilities for software packages to integrate with Microsoft's BizTalk. For example in InduSoft, an alarm occurs that generates an XML-based Biztalk's representation of a problem report. This is sent to a BizTalk Server and the BizTalk Server then shares the information with other

application or the web (diagram 2). The BizTalk Server is a scalable server product that will facilitate the interchange of BizTalk Framework-encoded information



Conclusion

XML is growing in importance in the industrial automation world from embedded systems to human machine interfaces, to distributed systems and database management. It has become the de facto standard for data communication between different applications, systems and throughout the Internet. The real benefit of XML will be when the industrial automation world standardizes on the vocabularies that allows problem reports, supplies request, part number, bill material numbers, customer numbers, document types, shipping characteristics, and all the terminology that allows us to work together and communicate from the factory floor to the top floor using the same language. And when you think about the inefficiencies that exist in the industrial world with disparate devices and systems that do not communicate or cannot share information. XML is a huge opportunity for cost savings, interoperability and new opportunities.