Transactions of the VŠB – Technical University of Ostrava, Mechanical Series
No. 2, 2008, vol. LIV,
article No. 1612

Marek BABIUCH[*], Lenka LANDRYOVÁ[**]

DATA MODEL IN INDUSTRIAL AUTOMATION USING NEW TECHNOLOGIES

NOVÉ TECHNOLOGIE DATOVÉHO MODELU V PRŮMYSLOVÉ AUTOMATIZACI

Abstract

This work presents some issues of the development and design of an industrial application based on new technologies and explores the technological dimension of data acquisition, storing, access, and use of data structures necessary for the control of production processes.

Abstrakt

Tato práce prezentuje na příkladech některá hlediska vývoje a návrhu průmyslové aplikace s pomocí nových technologií a otevírá technické rozměry získávání a ukládání dat, přístupu k nim a využívání jejich datových struktur důležitých pro řízení výrobních procesů.

1 INTRODUCTION

Referring to sources the semantic Web initiative led by W3C, the forum for information, commerce, communication, and collective understanding, has changed the semantics landscape. Through this initiative, researchers and developers joined their efforts to provide standard semantics markup languages based on XML, and to develop a framework for loading ontologies from files and via the Internet, for locally creating, modifying, querying, and storing ontologies and designing ontology management systems and other useful tools. In addition, the value of semantics in many applications was rediscovered, such as in the area of information and process integration.

In this work we would like to describe how this initiative has changed the software architecture used for production control and information management, the Service Oriented Architecture (SOA) as a software architecture that uses software services independent of the underlying platform and programming language. Since the interface definition hides the implementation of the language-specific service, the SOA-based systems can therefore be independent of development technologies and platforms (Java, .NET etc.). For example, services written in Java running on Java EE platforms and services written in C# running on .Net platforms can both be used by a common combined application.

Web services as a new class of applications were developed to serve multiple computer architectures, languages, and operating systems, enabling them to be exposed in a standard way. Therefore, the application developers can create new MES solutions using services for communicating via messages from different sources on the Web. The format of standard XML messages that are used to communicate among collaborative production management (CPM) system modules is further defined by Simple Object Access Protocol (SOAP), the standard defined by W3C.

[*] Ing., Ph.D., Department ATŘ-352, Faculty of Mechanical Engineering VŠB – Technical University of Ostrava, 17. listopadu 15, Ostrava - Poruba, 708 33, Czech Republic, (+420) 59 732 4173, e-mail marek.babiuch@vsb.cz

[**] doc. Ing., CSc., Department ATŘ-352, Faculty of Mechanical Engineering VŠB – Technical University of Ostrava, 17. listopadu 15, Ostrava - Poruba, 708 33, Czech Republic, (+420) 59 732 4112, e-mail lenka.landryova@vsb.cz

2 XML TECHNOLOGY

XML is the technology available here for us for sharing information easily based on a format of documents designed for reading over the Internet. XML is made for everybody and to be used by everybody and for almost anything by being easy to understand, easy to use, and easy to implement (Ortiz, 2002). This is one of the many reasons why it has become the universal standard and has faced and met the challenges of convincing us - the development and user community.

When the creators of XML were working out their design, their goals set for an XML document, among other factors, were defined by directions as to how XML is to be used:

- XML documents shall be easy to create, legible to read and reasonably clear, it should be easy to write programs that process XML documents,
- XML design should be prepared quickly, it should also be formal and concise,
- XML shall be straightforwardly usable over the Internet,
- XML shall support a variety of applications.

As referred to sources (Ortiz, 2002), XML offers a simple solution to a complex problem, a standard format for structuring data or information in a self-defined document format. This way, the data can be independent of the processes that will consume the data. But this concept behind XML is not new. It is a subset of a huge amount of specifications and conditions declared and developed by the World Wide Web Consortium (W3C) in 1986. The W3C began to develop the standard for XML in 1996. Since then, many software vendors have implemented various features of XML technologies.

An XML document contains a variety of constructs, also referred to as *elements*. Some of the frequently used ones include:

- Declaration: each XML document can have the optional entry <?xml version="1.0"?>. This standard entry is used to identify the document as an XML document conforming to the W3C recommendation for version 1.0.
- Comment: an XML document can contain HTML-style comments such as <!--Equipment data -->.
- Schema or Document Type Definition (DTD): in certain situations, a schema or DTD might precede the XML document. A schema or DTD contains the rules about the elements of the document. For example, we can specify a rule like "An equipment element must have EquipmentName, but AliasName is optional." .NET uses these schemas exclusively.
- Elements: an XML document is mostly comprised of elements. An element has a start-tag and an end-tag, for example Equipment Requirement. In between the start-tag and end-tag, we include the content of the element. An element might contain a piece of data, or it might contain other elements, such as EQUIPMENT. For example:

```
<Equipment Requirement>
  <Property List/>
  <Segment Requirement>
    <EQUIPMENT>
      <Property List/>
    </EQUIPMENT>
  </Segment Requirement>
  </Equipment Requirement>
```

- Root Element: in an XML document, one single main element must contain all other elements inside it. This specific element is often called the root element.

- Attributes: since an element can contain other elements or data or both, an element can also contain zero or more so-called attributes. An attribute is just an additional way to attach a piece of data to an element. An attribute is always placed inside the start-tag of an element, and we specify its value using the "name=value" pair protocol. For example: <RollMill_Diameter ID="3501" EquipmentName="RM_Diameter" Alias="RM_Diameter" Description="Roll Diameter" />

There is a more complete list of XML's constructs at www.w3c.org/xml.

In an XML document, the data are stored in a hierarchical structure. This hierarchy is also referred to as a data structures tree. That suits very well to the purpose of using it by system clients in our system architectures (Babiuch, 2007).

3 CREATING XML DOCUMENTS

When XML has emerged as the web standard for representing and sending data over the Internet, the W3C worked out and established a series of standards for XML and related technologies, including XPath, XSL, and XML schemas. VS.NET provides a number of tools to work on XML documents (Viscom .NET Team, 2007):

- XPath is a query language for XML documents. XPath queries are executed on data items. Search results are returned as a list of items. Each XPath expression may specify both the location and a pattern to match. Boolean operators, string functions, and/or arithmetic operators can be applied to XPath expressions in order to build quite complex queries against an XML document. Furthermore, XPath provides functions to evaluate numeric expressions such as summations and rounding. The full W3C XPath specification can be found at www.w3.org/TR/xpath (Ortiz, 2002).
- XSL translates XML documents from one format to another. The Extensible Stylesheet Language Transformation (XSLT) is the transformation component of the XSL specification by the W3C (www.w3.org/Style/XSL). It is basically a template-based declarative language that can be used to transform an XML document to another XML document or to other types of documents, for example HTML and/or text. Various XSLT templates can be developed and applied to select, filter, and process various parts of an XML document.
- XML schemas define the structure and data types of the nodes in an XML document, such as the 95 Equipment XML Schema, and other schemas developed and available from the World Batch Forum.

These technologies are industry standards backed up by the W3C. All of these standards were taken and packaged into the .NET architecture which we are working with now.

4 DATA MODELS DESCRIPTIONS

ISA-95 is the international standard for integration of control systems into management systems. It was written for common production environments and can be applied to any industrial area and for continuous, discrete and/or batch type of processes. The main objective was to create a framework for a project integration, to help to separate business processes from production, and to focus on production functions instead of functions of organizations or individuals (Landryova & Osadnik, 2008).

ISA-95 standard implementation provides models and information in multiple levels of detail and abstraction. Each model increases the level of detail defined in the previous model, however, they all come from the ISA-95 major object model, which is structured into:

- The basic resources models – such as equipment and material used in a process;

- A model for a product definition and product production rules, such as the business view of production on how to make a product and what is needed to make a product.

Assuming ISA-95 standard implementation and the use of its definitions for the Level 3 – MES, we will focus on object modeling for the component-based development of an application for a rolling mill supervisory control and visualization, as described in Figure 1.

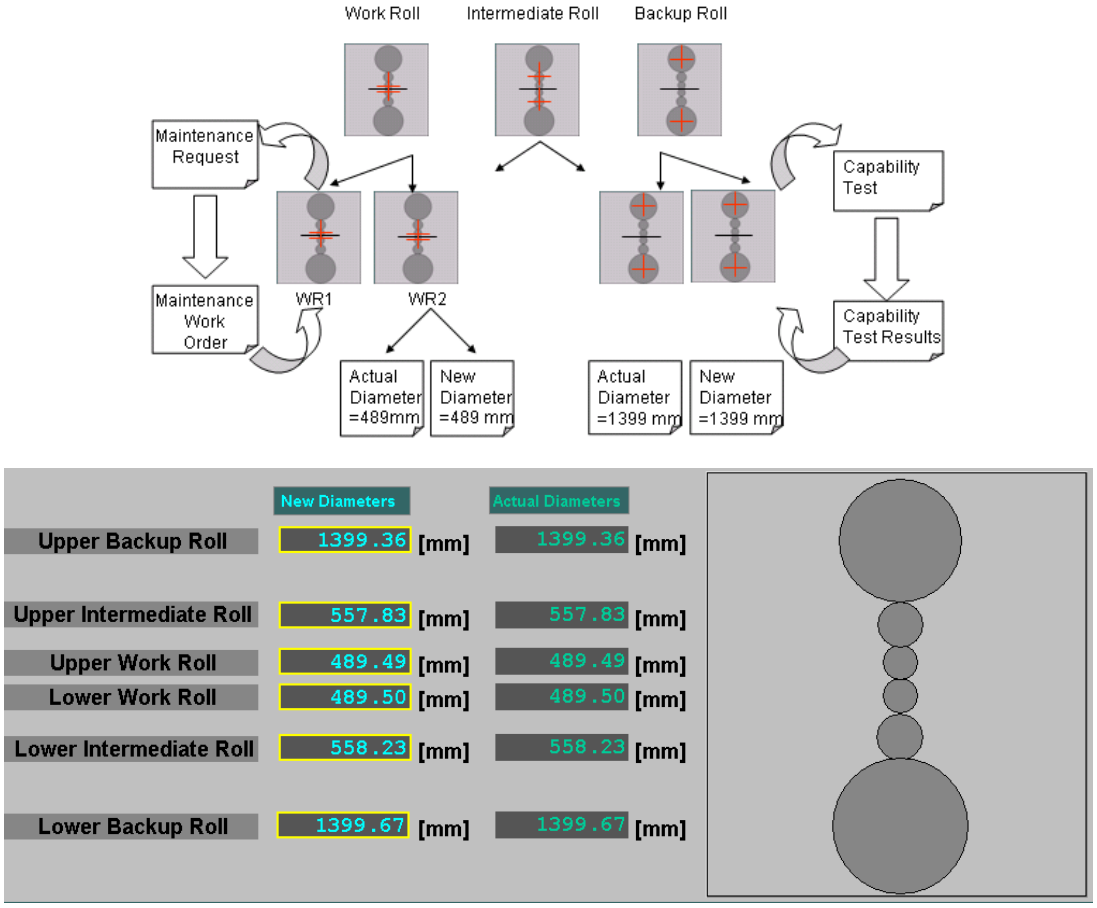

Fig. 1 Modelling equipment and equipment classes according to ISA-95 and the roll dimension millstand runtime screen

5 PRESENTING RUNTIME INFORMATION

Industries require the capability to handle events and display them to the end-users, supervisors, with the help of control system's *thin-clients*. Messages are defined for data being updated in the background of a manufacturing process. But the HTML protocol is *request-response based*. The *thin-client* browser must trigger "a new page" request. There is no default mechanism for event notifications without a trigger. So, the behavior of *browser-based* user interfaces of *thin-client* applications is that they can not automatically react to external events.

Services provided via web service interface allow industrial applications to access available data from the process. Web services here represent a layer between an industrial application and a *thin client*. They are the interface to the industrial application (API) developed according to a concept based on standard internet technologies (Babiuch, M.2007).

Simple Object Access Protocol (SOAP) used for this communication refers to the XML-based, extensible message envelope format enabling message transfer. SOAP consists of an *envelope*, a set of rule codes, and convention for *remote procedure calls* (RPC).

Web Service Description Language (WSDL) refers to the XML format allowing service interfaces to be described. WSDL file with service interface definition is an XML document using interface methods and their parameters.

Event notification is a type of service, which provides the mechanism to dispatch events to one or several supervisory stations. Events can be generated within the industrial application through:

- Business logic (BL) executed in an application server (AS), which provides services such as data consolidation, user authorization, client data access and scheduled report generation. The services are implemented as a set of objects and server processes.
- Custom scripting written in computer language, such as C#. The scripting environment provides an application programming interface (API) (Turtschi, Werry, Hack, Albahari, Nandu, Lee, 2003).
- By communication with external applications.

Working in environments and with software, which encodes meanings separately from data and content files, and separately from the application code enables end-users to understand the runtime screens, share knowledge and reason experience at execution time. With this type of software, adding, changing and implementing new relationships or interconnecting programs in a different way can be done as simply as changing and deploying the external model that these programs share. This gives the end-users the required flexibility in data manipulation before display in the GUI (Landryova, Valas & Winkler, 2008).

Using XPath expressions for *XML documents* simple calculations can be made in a runtime software environment. Therefore, the read-only types of properties whose values come from an associated calculation and not from the database are allowed and the calculation is configurable for runtime screens. The property, whose value is calculated, is then associated to an object. The object may, for example, change the color of its displayed background or of a text linked to it depending on some condition of its properties defined from specifications of the end-user.

For example, in a runtime screen for monitoring a production process, if the value of the monitored property *low pressure* is larger or equal than a limit set by a manual entry, the background of one display field goes red, or if the value of the property *high pressure* is lower than a limit set, the entire display row's background goes orange. The monitored properties become a part of the XML document of the object and are available for other runtime screen displays, for other scripting, and for a web service and publishing or copying data to periodical history.

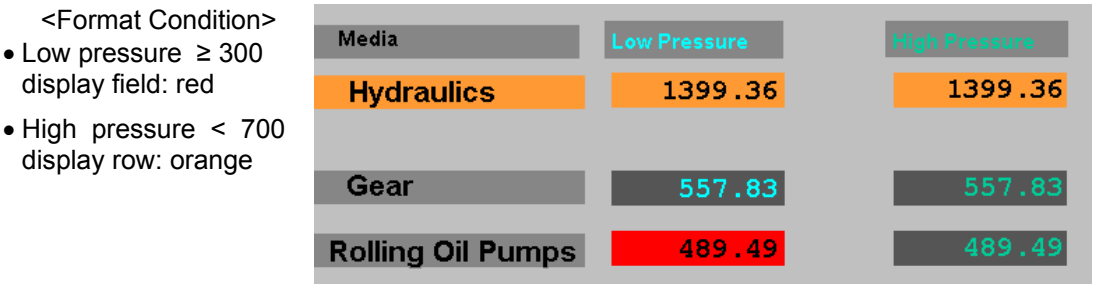

Fig. 2 Conditions for displays required by the end-user and the result from XPath implications on displayed values

6 CONCLUSIONS

Process automation serves to enhance product quality, improves functionalities of the whole range of products, improves process safety and plant availability, and efficiently utilizes resources and lowers emissions. The greatest requirement for process automation is the fastest growing demand for hardware, standard software and services of process automation. The traditional barrier between information and automation technology has disappeared. The latest technologies, including XPath, XSL, and XML schemas, help to improve the communication of process system modules.

The value of semantics was rediscovered in applications, which communicate among their system modules on Simple Object Access Protocol (SOAP), the standard defined by W3C, whose initiative has provided standard semantics markup languages based on XML as well. The data format and the format of standard XML messages that are used in industrial applications are defined by Business To Manufacturing Markup Language (B2MML) as W3C XML Schema for application implementation, while offering the framework for project integration.

7 ACKNOWLEDGEMENT

This work was concluded in our laboratories with the software and licenses provided by "Factory Suite Educational Demo Consignment" from Pantek (CS) s.r.o. Furthermore, the developed model and the created application are a part of our research conducted under the project Future Industrial Model for SMEs (Future SME) of FP7-NMP-2007-LARGE-1 supported by EU.

REFERENCES

[1] BABIUCH, M., (2007). Programming Applications for Internet II (in Czech). *Teaching text*, VSB-TU Ostrava, Czech Republic, pp. 182. ISBN 978-80-248-1504-6. Retrieved June 2, 2008 from http://www.352.vsb.cz/uc_texty/Welcome.htm.

[2] LANDRYOVA, L. & OSADNIK, P. (2008). Object-Oriented Approach to Development of MES Applications. In *XXXIII. Seminar ASR '2008 "Instruments and Control"*. Ostrava: VŠB-TU Ostrava, 27. 4. 2007, pp. 159-162. ISBN 978-80-248-1727-9.

[3] LANDRYOVA, L., VALAS, M. & WINKLER, O. (2008). Development of Tools for Online Access to Real Time Data from MES Software Solutions. In *XXXIII. Seminar ASR '2008 "Instruments and Control"*. Ostrava: VŠB-TU Ostrava, 27. 4. 2007, pp. 163 – 166. ISBN 978-80-248-1727-9.

[4] ORTIZ, J. (2002). *XML.NET Developer's Guide.* Syngress Publishing Inc. 593 pp. ISBN 1-928994-47-4.

[5] TURTSCHI, A., WERRY, J., HACK, G., ALBAHARI, J., NANDU, S., LEE, W.M. & DOT, Th.., (2003) *C# .NET Web Developer's Guide*. 817 pp. Retrieved Dec. 2., 2007 from http://www.podgoretsky.com/ftp/Docs/CSharp/CSharpWebDevGuide.pdf

[6] VISCOM .NET TEAM. OPC and .NET with COM Interoperability 2007. Retrieved January 15, 2007 from http://www.codeproject.com/dotnet/opcdotnet.asp.

Reviewers:

prof. Ing. Rudolf Palenčár, CSc., Slovak University of Technology in Bratislava

doc. Ing. Zdenka Prokopová, CSc., Tomas Bata University in Zlín