Teamcenter Engineering Data Exchange

Bronek Grzegorek
bronislaw.grzegorek@eds.com
EDS Corporate Overview

- 139,000 employees
- 20,000 support Manufacturing clients
- 2007 revenues of $22.1 billion
- Ranked 115 on the 2008 Fortune 500
- 2 in the IT Services category
- 2000 clients globally, in 65 countries

EDS Campus, Plano, Texas

- EDS provides a broad portfolio of business and technology solutions to help its clients worldwide improve their business performance
- EDS serves 75 of the top 100 Manufacturing clients globally
- 21 of the top 25 Global 1000 Automotive companies
- Siemens PLM Software is an EDS Alliance Partner
- EDS has a dedicated PLM services practice
EDS has a global footprint

Approximately 139,000 employees in over 60 countries

- North America: 53,000
- Asia: 30,000
- EMEA: 35,000
- South America: 14,000
- Australia/New Zealand: 7,000
EDS Understands the Importance of Commitments

• **We manage globally:**
  - CRM services from 149 locations in 47 languages
  - 300,000 servers, over 100 data centers and just over 3 million client desktops
  - 2,500 firewalls and 3,000 systems to detect intrusions, threats and vulnerabilities

• **We process annually:**
  - Over 5 billion credit card transactions
  - 500 million travel reservation transactions
  - $95 billion in U.S. Medicaid benefits for more than 20 million recipients

• **We prevent monthly:**
  - 550 million junk e-mails from reaching our employees and clients

• **We serve clients in every major industry and geography, including:**
  - 75 of the top 100 manufacturing companies
  - 109 domestic and international air carriers - more than any other IT services provider
  - 200+ government clients in 24 countries
  - more than 200 clients in 21 countries worldwide through our BPO Administrative and Transaction business units.
Agenda

– Needs for Data Exchange
– Existing Standards
– XLM technology overview
– PLMXXML data schema overview
– MSXML Parser with Internet Explorer
– How to read and manage PLMXXML files
– Demo of practical, working examples
Product Lifecycle Management

Is an integrated, information-driven approach integrating people, processes, business systems and technology to all aspects of a product’s life including design, validation, manufacture, maintenance and disposal.
Needs for Data Interoperability

Design Engineering

Manufacturing Engineering

Purchasing

Accounting

Legacy PDM Migration

ERP

Sales and Distribution

Maintenance

TeamCenter Engineering
Collaborative Product Development

It strategy, work process and collection of software applications that facilitates different organizations to work together.
CPD Needs

- Perform version control of files.
- Release status of products and assemblies.
- Support change management.
- Work with dispersed organizations.
- Process different geometry and metadata formats of used by partners’ systems.
- Provide secure exchange mechanism.
Asynchronous vs. synchronous CPD

An **asynchronously** a chief designer outlines the assembly configuration and the detailed component design tasks are assigned to individual designers to carry on separately.

A **synchronous** collaboration activity is conducted in a way such that a group of designers are dedicated to the same task actively.
Export for Change

OEM transfers files to a supplier that is allowed to those files, and/or may be depending of the authorized way of working.

OEM must protect these files against any modification, and is not allowed to create new revisions.
Export for Revise

OEM transfers files to a supplier that is not allowed to modify those files, but the supplier may only create new revisions.

OEM must protect these files against change and revising.
Export for Reference

OEM transfers files to a supplier who is not allowed to modify the files, or to create new revisions;

OEM may make modifications or revisions, and has to inform the supplier, when these revisions have impact on the work assigned to that party.
XML (Extensible Markup Language)

- Technology for WWW applications
- Sponsored by World Wide Web Consortium
- Design to simplifies business-to-business transactions
XML Benefits

• Applications can easy access and extract information.

• Enables robust database contents exchange

• Data can be used in HTML pages.

• Documents must follow certain rules.

• Data is structure is self-describing and extendable.
Simple XML Example
XML Schema

Example: Purchase Order Schema

The following example shows a schema, `po.xsd`, that defines a purchase order. This example shows the use of `element` and `attribute` declarations, `complexType` definitions.

```xml
  <xs:element name="PurchaseOrderType"/>
  <xs:element name="comment" type="xs:string"/>
  <xs:complexType name="PurchaseOrderType">
    <xs:sequence>
      <xs:element name="shipTo" type="URIMAddress"/>
      <xs:element name="BillTo" type="URIMAddress"/>
      <xs:element name="itemCode" type="ItemCode"/>
    </xs:sequence>
    <xs:attribute name="orderDate" type="xs:dateTime"/>
  </xs:complexType>
  <xs:complexType name="URIMAddress">
    <xs:sequence>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="street1" type="xs:string"/>
      <xs:element name="state" type="xs:string"/>
      <xs:element name="zip" type="xs:decimal"/>
    </xs:sequence>
    <xs:attribute name="country" type="xs:IDNString" fixed=""/>
  </xs:complexType>
</xs:schema>
```
XML Parsers

Validating Parsers

- The Apache XML Project is maintaining XML parsers in Java, C++, and Perl. These products from Apache.org: all Java, C++, and Perl platoform.

IBM's XML Parser for Java

Also known as SAX, Version 1 of IBM's XML Parser for Java was the highest rated Java XML parser in Java Report's February 1999 review of XML parsers. Version 2 adds those exciting new features: Customizable, Modular Architecture, High Performance, Real-Time, and XML/ODBC Support. Support for XML 1.0, DOM 1.0 and SAX 1.0 is also included. IBM's XML APIs are based on the Apache project. IBM's XML Parser Version 1.0.5. New features include experimental version of DOM Level 2, SAX2 (Java 2) and parts of XSD Schema. (See also the enormous XMLAPIWorks home page "Specialized XML Software" which is free software from IBM's all Java platform.)

JavaSoft's XML Parser

See Java Report's XAPI section for articles on Java parsers in the world of XML parsers. [Free product from JavaSoft; all Java platform.]

Oracle XML Parser

Oracle released its XML Parser forJava, a stand-alone XML component that enables parsing of XML documents through either SAX or DOM interfaces using validating or nonvalidating modes. See also the Oracle XML site [Free product from Oracle; all Java platform.]

XMLSAX

XMLSAX generates XML parsers for C, C++, Java, etc. According to the company, XMLSAX is able to achieve performance comparable with message-specific hand-coded parsers by skipping the intermediate step where the message is turned into a generic DOM tree using a generic parser which must take the entire generality of XML into account and support every feature, no matter how obscure. The parser generated by XMLSAX only requires the XML feature required to parse the message at hand, and produces directly a parser that initializes application-level data structures without going through any time-consuming intermediate representation. Tool features: (1) Generates parsers, which are between 5 and 10 times faster than generic parsers in C, C++, Perl and Java. (2) Produces working data structures in the host language, rather than a dynamic and possibly typed generic tree. (3) The XML message to parse comes from a file, a message, a socket, a data stream, etc. (4) Produces relatively validating parsers, far beyond the validation possibilities of DOM. (5) (Commercial product for C, C++, Perl, Java.)

SAX, the SimpleXML Parser

The SimpleXML Parser (Sax) is a parser and a complete XML API for Java. It is part of XML4J, a desktop-based environment. XML4J includes SAX. XML4J is an interface language, among other things. The SAX DOM is organized using modules, including (1) the encoding of the user information, (2) the encoding of the user information, (3) the encoding of the user information, and (4) the encoding of the user information. (5) Produces DOM-based parsers, all Java platform.

XMLAPIWorks XML Parser

Microsoft XML parser in Java is included in IE5. The version presently available provides the first XML 1.0 spec by month. "The parser checks for well-formed documents and optionally checks of the document's validity. Once parsed, the XML document is exposed as a tree through a simple set of Java methods, which Microsoft is working with the World Wide Web Consortium (W3C) to standardize. These methods support pushing and pulling XML structures. "See sample parsing of an XML file using SAX. (Microsoft also includes an XML parser in C++ and IE5, which is high-performance, nonvalidating parsers, free support of the W3C XML specification.) Free product from Microsoft; all Java platform."

DOM

XML is now being developed with a development effort with Microsoft and IBM. It is included in Internet Explorer 5.0. The DOMchileXMLParser (DOM) is a validating parser written in Java. DOM is specifically aimed at providing a base for server-side applications that need to integrate XML-based software. DOM provides the highly substituted error-checking mechanisms required for XML-based data interchange. DOM has not been tested on IBM's JavaXML (XML API) parsers, one of the earliest XML parsers. See also DOMDocument and the "Specialized XML Software" and XMLAPIWorks; all Java platform.

Larval

Larval is Tim Bray's validating XML processor built on the same code base as Lark (below). Larval is a full validating XML processor; it reports...
XML Schema

**XPath** (XML Path Language) is a language for selecting nodes from an **XML** document.

*Examples:*

- `PLMXML/ProductRevision`
- `PLMXML/ProductRevision/AssociatedDataSet`
- `PLMXML/Product[@id='id1245']`
- `PLMXML/Form[@role='IMAN_MASTER_FORM']/UserData/UserValue`
- `PLMXML/DataSet[@type='DirectMode']`
- `PLMXML/ExternalFile[@id='id40']/locationRef`
An XSL stylesheet templates that define how the elements in an XML document should be transformed.
XML Resources

Amazon.com over 980 books about XML!
DOM and XML

The Document Object Model (DOM) as implemented in MSXML provides a programmatic representation of XML documents.
PLMXML

- Similar to STEP product data definition (schema)
- It is becoming industry “de facto” standard
- Implements XML format for data exchange interoperability
PLM XML is defined by a set of XML schemas that are compliant with the W3C XML schema recommendation.
Exporting PLMXML Interactively

<table>
<thead>
<tr>
<th>Entry Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export Directory</td>
<td>Select the directory into which the PLM XML file along with associated files should be written to.</td>
</tr>
<tr>
<td>Export Filename</td>
<td>Specify the name of PLM XML file.</td>
</tr>
<tr>
<td>Transfer Mode Name</td>
<td>The name of Transfer Mode to be used for this translation. Based on the scope defined in the Transfer Mode, the contents of the xml file are appropriately generated.</td>
</tr>
<tr>
<td>Open PLM XML File</td>
<td>Allows you to view the XML file generated in an editor after completion of export, if the check box is on.</td>
</tr>
<tr>
<td>View Log file</td>
<td>Allows you to view the log file if this check box is on. It allows checking for any errors due to translation.</td>
</tr>
</tbody>
</table>
Command line export

```bash
plmxml_export -u=user -p=pw -g=grp -
xml_file=myAssembly1.xml -item=012345
    - rev=A -export_bom=yes -
transfermode=ConfiguredDataExportDefault
```
Creating Transfer Mode Objects

CLASS.Document:CLASS.Form:CONTENT.*:SKIP
CLASS.*:CLASS.ProductView:CONTENT.*:PROCESS
CLASS.ProductDef:CLASS.InstanceGraph:CONTENT.*:SKIP
CLASS.*:CLASS.*:CONTENT.*:DO
CLASS.*:CLASS.*:Attribute.*:DO
Loading PLMXML

```javascript
var root = new ActiveXObject("Msxml2.DOMDocument.5.0");

root.async=false;
root.load("myNEW.plmxml");
root.setProperty("SelectionLanguage", "XPath");
if (root.parseError.errorCode != 0)
{
    result = reportParseError(source.parseError);
    alert(result);
}
```
transforming with xslt

```javascript
stylesheet = new ActiveXObject("Msxml2.DOMDocument.5.0");
stylesheet.async = false;
stylesheet.load("hilite-xml.xsl");

if (root.parseError.errorCode != 0) {
    alert( reportParseError(source.parseError));
}

try {
    result = root.transformNode(stylesheet);
}
catch (e) {
    alert( reportRuntimeError(e));
}
```
Query with XPath

```javascript
function doQuery(theQuery) {
    var q = source.selectNodes(theQuery);
    alert(q.length + " Objects Found")
    for (var e = q.nextNode(); e != null; e = q.nextNode()) {
        try {
            alert( "Node " + e.xml )
        }
        catch (exception) {
            alert reportRuntimeError(exception);
        }
    }
}
```

```
theQuery="PLMXML/ProductRevision"
theQuery="PLMXML/ProductRevision/AssociatedDataSet"
theQuery="PLMXML/Product[@id='id1245']"
theQuery="PLMXML/Form[@role='IMAN_MASTER_FORM']/UserData/UserValue"
theQuery="PLMXML/DataSet[@type='DirectMode']"
theQuery="PLMXML/ExternalFile[@id='id40']/locationRef"
```
nodes=source.getElementsByTagName("ProductRevision");
//or nodes = root.selectNodes("PLMXML/ProductRevision");

for (var i=0; i<parts.length; i++) {
    var productRevId=nodes[i].getAttribute("id");
    var descriptionNode=nodes[i].getElementsByTagName("Description");
    var item_rev=nodes[i].getAttribute("revision");
    var item_name = nodes[i].getAttribute("name");
    var masterRef=nodes[i].getAttribute("masterRef").substr(1);
Evaluating Form Nodes

```javascript
var formNode = root.selectNodes("PLMXML/Form[@id="" + form_id + "]/UserData[@type='FormAttributes']/UserValue");

var productName = root.selectNodes("PLMXML/Form[@id="" + form_id + "]")[0].getAttribute("name");

for (var ii = 0; ii < formNode.length; ii++) {
    title=formNode[ii].getAttribute("title");
    value=formNode[ii].getAttribute("value");
}
```
Evaluating Form Nodes

```javascript
var productNode = root.selectNodes("PLMXML/Product[@id="" + masterRef + "]");
var productId=productNode[0].getAttribute("productId");
var master_formID=AssociatedDataSet[0].getAttribute("formRef").substr(1);
var AssociDataSet=nodes[i].selectNodes("AssociatedForm[@role='IMAN_specification']");
var pdm_formID=AssociatedDataSet[0].getAttribute("formRef").substr(1);
```
Setting new attribute value

```javascript
var formNodes = root.selectNodes("PLMXML/Form[@id="" + object.id + "]/UserData/UserValue[@title="" + object.name + "]");

formNodes[0].setAttribute("value", object.value);
```
function SaveXML (filename){
    if ( filename.length == 0 ) {
        alert ("Destination file name required !");
        return (0);
    }
    alert("Saving " + filename + ".xml");
    root.save( filename + ".xml");
}
Summary

- Growing needs for robust product data exchange.
- Why an existing standards are not adequate?
- PLMXML as vertical implantation open IT standard could become “de facto” standard of PLM.
- Overview of MSXML as one of many available XLM toolboxes.
- Example of code bites for managing PLMXML files.
- Demo of practical examples
Questions ?