Using XML Web Services for Information Exchange in America's Shipyards

Dr. Thomas Daggett
Dr. Margaretha Price

Electric Boat Corporation, General Dynamics
Agenda

- ISE Project Overview
  - Business Case
  - Approach
- Electric Boat’s activities
  - Product Data Web Service
    - Architecture
    - Approach
    - Products utilized
  - Information Modeling
    - Schema architecture and encoding
    - Persistent identifiers (keys)
    - Type checking
    - Associations
Electric Boat is leading the ISE Project as part of the National Shipbuilding Research Program (NSRP)

- Being executed by a consortium of approximately 20 Shipyards, CAD Vendors and Universities
- Objective
  - develop and deploy an interoperability infrastructure and tool set that will result in the integration of shipbuilding systems through the effective use of readily available Web technologies
Integrated Shipbuilding Environment
Project Approach

- Develop and demonstrate tools that are low cost
  - can be selectively used by shipyards to support interoperability
  - capitalize on XML and related Internet technologies

- Flexibility is critical
  - allow shipyards to transform their data to/from common information model

- Drive development of manufacturing standards

- Construct a single Shipbuilding Information Model

- Demonstrate and educate U.S. ship building community
Co-design
* Web-based system diagrams
* Web-based interop with simulation & analysis

Co-production
* Web-based product model sharing
* Browser access to product-related documents

Logistics
* Web access to as-planned & design product structure

Information Support

ISE
Integrated Shipbuilding Environment

Maritime Supply Chain
* Web-based supplier catalogs
* Shared parts library

Product Development Environment
Activities

- **Information Modeling**
  - construct Information Models (XML Schema) to describe shipyard product data
    - capitalize on existing ISO 10303 manufacturing standards (STEP APs)
    - drive manufacturing standards to support Shipyard data
  - define a common XML approach
    - encoding styles and use of Persistent Global identifiers

- **Product Data Management (PDM) Web Service**
  - construct a web service that allows querying of a Product Data Virtual Document
    - SOAP Web Service utilizes Xqueryx to specify query
  - demonstrate PDM Web Service through Enhanced Product Model (EPM) thin-client application
ISE EPM Project utilizes following products:

» Apache SOAP Server
  – SOAP Messaging used between presentation layer and PDM data Web Service

» Apache Jakarta (Web/JSP Server) Server
  – JSPs and XSLT scripts comprise presentation layer

» Apache Xalan/Xerces XSLT Engine/XML Parser
  – XSLT scripts utilized for information model transformation from EB specific form into ISE Information Model form

» Oracle 9i XSQL/XSLT Server
  – used to extract data as XML content

» Oracle Transparent Gateway
  – used to host legacy DB2 database
ISE Information Interoperability

**Parts**
- ISO 13584 (PLIB)

**Product Data**
- ISO 10303:
  - 227 Piping
  - PDM
  - 216 Hull Forms
  - 218 Structures

  • 209 Analysis (STEP)

**CAM-NC**
- ISO 14659 (STEP-NC)

**ERP**
- ISO 15531 (ebXML, OAGIS)

**Customer**

**Analysis**

**Production Planning**

**Manufacturing / Outfit**

**Collaboration**

**Life-Cycle Support**

**ISE Virtual Document**
- Electronic Deliverables
- Interoperability (system-to-system)
- Data Exchange
- Dependencies are from the more specific schemas to the more general schemas
- Not all schemas are needed for all contexts
Object Serialization Early Binding (OSEB)

- streamlines the exchange of XML instance files to achieve a flat & size-optimized structure
- uses XML Schema attributes for data and references to other entities
- mainly used for application to application transfers

STEP-SOAP

- almost always consistent with the SOAP-encoding
- XML Schema elements are used for data and references to other entities
- elements can be contained or referenced
- mainly used for user readability
An EDO is a persistent, globally accessible entity that has
  » a type and an identity

Each EDO has an EDO Key (of the same type + "_Key")
  » instance files can contain just keys
    (keys have enough information for retrieving the objects)
  » keys have server information
  » keys contain primary key attributes

Sample OSEB-encoded EDO key and its object

```xml
<pdm:Item_Key x-id="i1" x-ob="i2" x-href="www.company.com" Id="12345"/>
<pdm:Item x-id="i2" Name="Pipe" Description="Straight pipe" shp:Representations-r="i3"/>
```
In the STEP-SOAP format:

- the attributes are contained elements
- key attributes are also in the object

(instance files can contain just keys, just objects, or both)

```xml
<pdm:Item_Key id="i1">
    <key x-href="www.company.com" Id="12345"/>
</pdm:Item_Key>

<pdm:Item id="i2">
    <key x-href="www.company.com" Id="12345"/>
    <Name>Pipe</Name>
    <Description>Straight pipe</Description>
    <shp:Representations>
        <hsh:Shape_representation href="i3" xsi:nil="true"/>
    </shp:Representations>
</pdm:Item>
```
Type Checking in STEP-SOAP

Is the referenced entity of the right type (or its subtypes)?

```xml
<shp:Representations>
  <hsh:Shape_representation id="i3">
    <key x-href="www.company.com" Element_id="111"/>
  </hsh:Shape_representation>
</shp:Representations>
```

Schema

```xml
<xs:complexType name="Shape_representation"> ... </xs:complexType>

<xs:element name="Shape_representation" type="hsh:Shape_representation" nillable="true"
block="extension restriction"/>

<xs:complexType name="Specific_shape_representation">
  <xs:complexContent>
    <xs:extension base="hsh:Shape_representation"> ... </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:element name = "Specific_shape_representation" type = "hsh:Specific_shape_representation"
substitutionGroup = "hsh:Shape_representation" nillable = "true" block = "extension restriction"/>
```
Is the referenced entity of the right type?

```xml
<pdm:Item x-id="i2" Name="Pipe" Description="Straight pipe" shp:Representations-r="i3"/>
<hsh:Shape_representation_Key x-id="i3" x-href="www.company.com" Element_id="111"/>
```

**Schema**

```xml
<xs:element name="uos" type="osb:uos">
    ...
    <xs:key name="Shape_representation_Key">
        <xs:selector xpath="hsh:Shape_representation_Key|
                          hsh:Specific_shape_representation_Key|...(all subtypes)..."/>
        <xs:field xpath="@x-id"/>
    </xs:key>
    <xs:keyref name="Item.Representations" refer="Shape_representation_Key">
        <xs:selector xpath="pdm:Item"/>
        <xs:field xpath="@shp:Representations-r"/>
    </xs:keyref>
    ...
</xs:element>
```
Associations among EDO

- Schemas contain flexible & reusable types
- Associations among EDO are separated from the types
- Inverse associations are always available

Diagram:

- A
- B
- C
  (associations)
- A2
- B2
- Context

Arrows indicate:
- Import
- Redefine

Other schemas
<pdm:Item x-id="i2" Name="Pipe" Description="Straight pipe" shp:Representations-r="i3"/>

**Schema**

```xml
<xs:complexType name="Item">
  <xs:complexContent>
    <xs:extension base="edo:Edo">
      <xs:attribute name="Name" type="xs:string" use="optional"/>
      <xs:attribute name="Description" type="xs:string" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

**Redefined Schema**

```xml
<xs:redefine schemaLocation="/.../osb/Edo/Product/Pdm.xsd">
  <xs:complexType name="Item">
    <xs:complexContent>
      <xs:extension base="Item">
        <xs:attribute ref="shp:Representations-r" use="optional"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:redefine>
```
Any questions or comments?