Using XML to Integrate Aircraft Mechanical 3D CAD Data into a Visual Repository of Tessellated, Articulated 3D Geometry

George W. Earle
VP Enterprise Services
Enterprise Services & Solutions Group
Right Hemisphere Incorporated
San Ramon, CA, USA
George.earle@righthemisphere.com

Sourajit Guha
Solution Architect
Enterprise Services & Solutions Group
Right Hemisphere Incorporated
San Ramon, CA, USA
Sourajit.Guha@righthemisphere.com

Abstract – *Airframe manufacturing has been described as more art than science. This is due, among many reasons, to extensive use of manual labor instead of robots and assembly procedures based on text and paper blueprints. The elimination of paper blueprints and instructions can be achieved with the use of tessellated mechanical CAD data projected onto computers in manufacturing. A system that extracts, transforms and loads 3D CAD data and XML part positioning and hierarchy information into a repository of tessellated 3D geometry can eliminate the use of paper instructions and blueprints. The XML architecture automatically transfers CAD data, performs 3D matrix transposition and scalar data transpositions to populate a visual information repository for use in assembly manufacturing. This visual repository is populated by articulated 3D instructional procedures, used in place of paper blueprints and instructions. Airframe manufacturing using design-based information therefore will move from an art based process to a science based process.*

Keywords: 3D, CAD, MCAD, tessellation, visual information, 3D XML, visual enterprise, airframe manufacturing, aircraft manufacturing

1 Introduction

Manufacturers' 3D computer aided design (CAD) product data should be sourced, accessed, and shared in a secure, collaborative, concurrent engineering and multi-disciplinary way. The system supports this goal through the employment of 3D visualization technology to distribute airframe product information across all departments: engineering, technical publications, simulation, training, manufacturing and so on. This enterprise integration solution uses uniform Service Oriented Architecture (SOA) interfaces and federated search to export 3D product data (i.e., CAD) from multiple engineering product data management repositories (PDMs) into a visual product information system.

The solution eliminates manual methods and processes of visual product information searching, data exporting, and change management across multiple departments in an organization. The solution automates 3D product data integration and uses XML to hide data structure and the complexities of data accessibility from the organization's non-engineering users.

2 Current Practice

In recent decades, mechanical design has become completely visual and digital, residing in large (terabyte) repositories of CAD data structured as 2D and 3D information. This information is sequestered in the design departments of manufacturers. Mechanical CAD data typically requires expensive workstations and CAD software to access, view and manipulate 3D product designs. Product CAD data is therefore not reused easily, inexpensively, efficiently or quickly.

Manual methods and processes of visual product information searching, data exporting, and change management across the multiple parts of a manufacturing organization are incredibly inefficient, wasteful and labor intensive. Users often replicate product visual information multiple times, disassociating the product or assembly's graphics and visualization from current and ongoing development work, therefore ensuring the visual representations are often out of date immediately after reproduction. This leads to a possibly hazardous situation where follow-on engineering, manufacturing and product work is using out of date information from these visual reproductions.

Efficiency in maintaining the relationship to the CAD data is important because of the thousands of CAD entities and product changes that occur during a product's lifetime. Exporting CAD data in a manually oriented manner can be burdensome and error prone that can result in exporting the wrong data geometry, bill of material structures, or release of CAD data. Choosing an operationally efficient mechanism for this association is imperative to lowering the overall time and cost of downstream graphics reuse.

3 Best Practice

Our best practice is to replace the current norm with the employment of a 3D CAD integration, transformation, and visualization system deployed between engineering repositories and a visual information enterprise system. An intelligent, visual product information system supports all of a company's products and presents 3D interactive product models for employees while supporting data security policies, contains a straightforward user interface, and provides scalability for terabytes of data. This allows all users, non CAD and non-engineers alike, access, federated search and delivery of accurate, visual (lightweight 3D, not CAD) product information.

The 3D CAD integration system consists of an orchestration server and collaboration adapters to perform CAD data extraction, format normalization, and transformation. The adapters normalize multiple PDM systems' application programming interfaces (API) and their exported bill of material (BOM) data while the orchestration server performs BOM and CAD data extraction and processing. Below are some additional technical characteristics of the integration system:

- Java Platform, Enterprise Edition (Java EE) architecture is used by the integration system (DS Connect) to orchestrate real-time CAD export and processing out of multiple PDMs (running on either Unix or Windows operating systems),

- XML interchange, multi dimensional product data transformation [1-3].

The integration server provides an operationally efficient mechanism for CAD association by maintaining a data relationship between the two data sets and executing a processing workflow that is tied to the PDM release process for its CAD data. It is also responsible for utilizing any security infrastructure tied to the PDM and its CAD data.

4 CAD data and XML transformation

Our integration technology automatically exports and transforms a CAD system's proprietary engineering BOM (EBOM) that structures the CAD data into our Deep Server BOM (DS BOM). The system not only transforms the CAD system's EBOM but also normalizes the EBOM across multiple PDM systems to that the integration system only has to perform the transform into the DS BOM once.

XML mapping software can be used to map the fields in the source XML (CAD EBOM) to the target XML (DS BOM). Our process adds a normalization step. This means we transform the CAD EBOM to another intermediate XML, PDNet XML, and then transform that into the DS XML. Mapping between XML schemas is similar to what is shown in Figure 1.

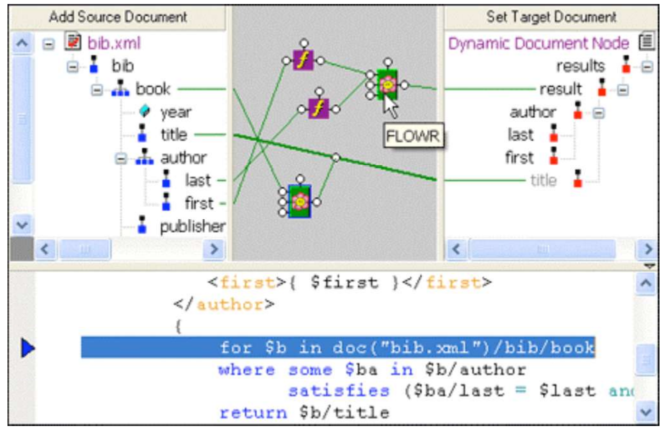

Figure 1. Example of mapping between XML schemas.

Each system maintains a strict formal definition of their EBOM, PDNet BOM and the DS BOM. Based on those, we developed an algorithm for automatically transforming EBOM into the PDNet BOM and then into the DS BOM.

The Integration System:

- Performs transformation (XForm) services that allow the source and target to use different communication protocols. It uses mapped Product data and relationships (BOM) between the PDM structure and the RH structure.

- Encodes business rules controlling product data transfer. These business rules are triggers that execute when a part or assembly is created, updated, deleted, renamed, repositioned or changed in any other way defined by a product design team.

- Provides routing services, relieving the PDM application from being aware of the target application, in this case the Visual Repository system.

- Navigates the hierarchical BOM product structure in a PDM based on Query criteria, i.e., combinations of Part Number, Part Name and Part Version

- Creates an intermediate eBom/CAD BOM structure based on the open standard, the OMG PLM Services [4]

- Performs rule based exports, e.g., part file names matching certain patterns, etc.

- Transforms the product structure XML into a targeted XML structure (i.e., the DS BOM) to further integrate with other systems, e.g., Content Management, Digital Asset management etc.

- Iteratively navigates the BOM product structure identifying CAD files to extract and performs the CAD extraction onto disk

- Creates an upload job to push all CAD data and BOM structures into the target system.

The PDTNet BOM to Deep Server BOM transformation executes following similar logic through an XSLT Style Sheet implementation [5].

1. Look for <Assembly_component_relationship> tag in the PDTNet BOM like below

```
<Assembly_component_relationship
id="ACR_PARTS-
42489796F2B551E8_4248979BF0133467"
xsi:type="Next_higher_assembly">
```

2. Look for the Key to the <Item_Id> tag

3. Follow the <Item_Id> Key to find the <Item_Id> tag in PDTNet XML

```
<Item id="I_PARTS-_57x01-011-409-
107">
```

4. Find <Item_version> tag to find the actual Part version

```
   <Item_version id="IV_PARTS-
42489790040F3542_3030303030303030">
```

5. Find <Item_Instance> tag to get the Instancing information

```
        <Item_instance id="II_PARTS-
42489796F2B551E8_4248979BF0133467"
xsi:type="Single_instance">

          <Id>II_PARTS-
42489796F2B551E8_4248979BF0133467</Id>

        </Item_instance>
```

6. Look up <Assigned_document> key to look up <Document> tag

```
<Assigned_document>DOCV_PARTS-
42489790040F3542_42489790040F3543</A
ssigned_document>
```

7. Find Document tag

```
<Document id="DOC_PARTS-
42489790040F3542_42489790040F3543">
    <Document_id>901-011-409-107/1
/E /6 /EDD/3D/U/RTRS/STRUT, ANTI-
DRIVE</Document_id>
```

8. Look up <File> key to lookup <Digital_File> tag

```
        <File>DF_PARTS-
42489790040F3542_42489790040F3543</F
ile>
```

9. Find <Digital_File> tag

```
<Digital_file id="DF_PARTS-
42489790040F3542_42489790040F3543">
    <File_id>901-011-409-107/1 /E /6
/EDD/3D/U/RTRS/STRUT, ANTI-
DRIVE.model</File_id>
     </Digital_file>
```

10. Recursively traverse for all the instances of the Assembly, i.e. Steps 2-9

11. Construct Deep Server BOM with XML elements like the following

```
<instance uid="I_PARTS-_57x01-011-
409-107" name="901-011-409-107">
      <position p1="1.0" p2="0.0"
p3="0.0" p4="0.0" p5="1.0" p6="0.0"
p7="0.0" p8="0.0" p9="1.0" p10="0"
p11="0" p12="0" />
      <metadata>
        <property name="filename"
value="901-011-409-107/1 /E /6
/EDD/3D/U/RTRS/STRUT, ANTI-
DRIVE.model901-011-409-107/1 /E /6
/EMU/3D/U/RTRS/STRUT, ANTI-
DRIVE.model" group="CADMetadata" />
      </metadata>
       </instance>
```

12. We have now converted from one XML format to another, an algorithm for automatically transforming EBOM into the PDNet BOM and then into the DS BOM.

Now having created a mapping system from the intermediate and normalized PDNet BOM we never have to create another mapping from a CAD system to a visualization system.

The result of the architecture of normalizing different CAD and PDM BOM structures is that we can add new PDM repositories to the integration system much quicker since we are not building a whole new integration system. We simply create an adapter to that new PDM system that outputs PDNet XML. The technical effort has therefore been encapsulated into the adapter logic and does not affect the rest of the integration system. This greatly shortens system development and testing time.

5 Using a Visual Repository

By merging BOM XML data with visuals in the context of manufacturing work instructions, a visual manufacturing solution is enabled that reuses existing CAD data, textual manufacturing instructions and part bill of material structures to eliminate waste and accelerate manufacturing processes with higher quality (example in Figure 2).

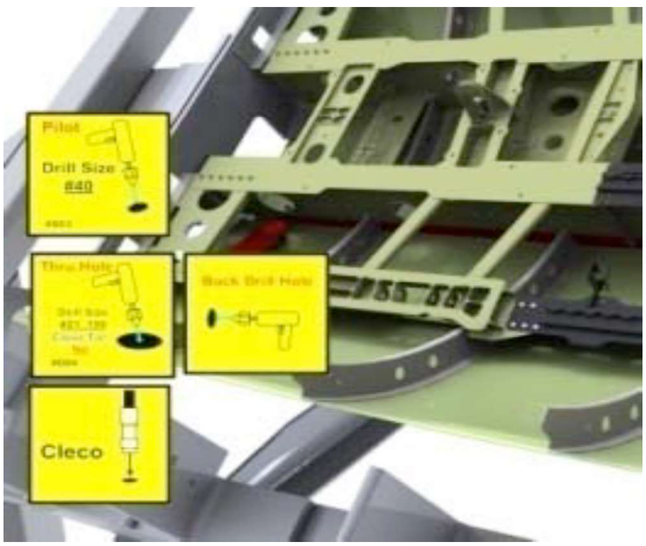

Figure 2. Example merge of BOM XML data with visuals.

The BOM XML data (CAD or DS versions) contains product configurations, engineering changes and effectivity. The integration system keeps Visual 3D work instructions up to date through the association of part numbers in the CAD BOM to the textual manufacturing instructions to the CAD geometry.

After the integration system exports, transforms, and processes the CAD geometry and XML structures, the complex CAD geometry is transformed into lightweight manufacturing models that are automatically structured and sequenced for each manufacturing process. An authoring process that uses the 3D model and XML structures can produce animated 3D work instructions. These animated 3D graphics depict the assembly procedure for each station on a manufacturing line.

6 Benefits of Articulated 3D in Manufacturing

Visual 3D work instructions reduce manufacturing errors by eliminating ambiguity and are parameterized by manufacturing execution information to coincide exactly with each product build. Production staffs learn new processes and achieve higher production rates significantly faster using 3D visual instructions. 3D visual instructions are articulated, animated, by authoring software that allows manufacturing engineers to create procedures that simulate the actual assembly of the product [6].

Articulated 3D manufacturing based on lightweight 3D provides a number of benefits:

- Auto-generated Manufacturing Instructions: 80% reduction in manpower effort
- Animated Work Instructions: Accelerated shop floor learning with superior retention
- Synchronized Engineering Changes: Accuracy on the shop floor eliminates rework
- MES/MRP Integration: Shop floor instructions are defined for each product configuration
- Language Independent Graphics: Consistent repeatable processes at any location in the world
- Lightest Weight 3D: Large models are fast and easy to view resulting in rapid user adoption
- Display of 3D CAD Features and Tolerances: Model Based Definition content is easily deployed.

7 Future Development

Future work will focus on increasing the sophistication of the transformation algorithms to enable just in time change management based on product development process notifications. This will enable the visual 3D models to maintain their currency with the product release process.

8 Conclusions

We have shown that CAD data can be extracted and repurposed into productive use outside of engineering through algorithms that transform the native data into new forms used in new applications. These algorithms are designed to normalize different CAD structures into a single structure, thereby avoiding the many to many mapping problem that prevents many projects from being successful. We have shown that this transformation leads to a new class of visual applications based on a company's valuable product CAD data.

We have shown that the elimination of paper blueprints and instructions can be achieved with the reuse of transformed CAD data projected onto computers in manufacturing operations.

References

[1] W3C® (MIT, ERCIM, Keio), "XML Schema," http://www.w3.org/XML/Schema.

[2] W3C® (MIT, ERCIM, Keio), "XSL Transformations (XSLT)," http://www.w3.org/TR/xslt.

[3] The Apache Software Foundation, "The Apache Xerces Project," http://xerces.apache.org/.

[4] M. Feltes, *Daimler Chrysler Research OMG PLM Services*, Object Management Group, 2005, http://www.omg.org/docs/omg/05-04-12.pdf.

[5] The Apache Software Foundation, "The Apache Xalan Project," http://xalan.apache.org/index.html.

[6] D. Prawel, "The Advent of Visual Manufacturing: White paper," Longview Advisors, Inc., 2008, http://www.ien.com/article/advent-visual-manufacturing/144284.