**Proceedings of DETC 2004:**
**ASME 2004 Design Engineering Technical Conferences and Computers and Information in Engineering Conference**
**September 28–October 2, 2004, Salt Lake City, Utah USA**

# DETC2004-57743

# STEP, XML, AND UML: COMPLEMENTARY TECHNOLOGIES[1]

**Joshua Lubell**
National Institute of Standards and Technology
100 Bureau Drive, Stop 8263
Gaithersburg MD 20899-8263 USA
Email: lubell@nist.gov

**Russell S. Peak**
Georgia Institute of Technology
Manufacturing Research Center
813 Ferst Drive, MARC 373
Atlanta, GA 30332-0560 USA
Email: Russell.Peak@marc.gatech.edu

**Vijay Srinivasan**
IBM Corporation
1133 Westchester Avenue, Mail Drop 160
White Plains, NY 10604 USA
Email: vasan@us.ibm.com

**Stephen C. Waterbury**
NASA Goddard Space Flight Center
Code 562
Greenbelt, MD 20771 USA
Email: stephen.c.waterbury@nasa.gov

## ABSTRACT

One important aspect of product lifecycle management (PLM) is the computer-sensible representation of product information. Over the past fifteen years or so, several languages and technologies have emerged that vary in their emphasis and applicability for such usage. ISO 10303, informally known as the Standard for the Exchange of Product Model Data (STEP), contains the high-quality product information models needed for electronic business solutions based on the Extensible Markup Language (XML). However, traditional STEP-based model information is represented using languages that are unfamiliar to most application developers. This paper discusses efforts underway to make STEP information models available in universal formats familiar to most business application developers: specifically XML and the Unified Modeling Language™ (UML®). We also present a vision and roadmap for future STEP integration with XML and UML to enable enhanced PLM interoperability.

## 1. INTRODUCTION

Many businesses are turning to business-to-consumer and business-to-business solutions based on the Extensible Markup Language (XML) [1] to reduce transaction costs, open new markets and better serve their customers. These solutions, which tend to emphasize messaging and business processes, require basic information about products. Many of these XML business vocabularies are ad-hoc and/or conflict with other XML applications. Missing from these solutions is a rigorous definition of the business information concerning the design, manufacture and support of these goods. To ensure this business data's longevity, it should be represented using a language defined by an open standard and not dependent on any particular software application. In this paper we discuss three open standards we believe are useful for representing product information: the Standard for the Exchange of Product Model Data (STEP) [2] [3], XML, and the Unified Modeling Language™ (UML®) [4].

### 1.1. Types of Standards

What do we mean by an "open standard" in the context of technical information management? Open standards are one of at least three types of standards used in industry, as described below.

1. **Open Standards** relate to the general idea of interoperability and integration — an agreement that people make so that products and systems made by different parties can work together. Open standards are not software applications; they are only specifications

[1] Commercial equipment and materials are identified in order to describe certain procedures. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose. Unified Modeling Language, UML, Object Management Group, OMG, and XMI are trademarks or registered trademarks of the Object Management Group, Inc.; in the U.S. and other countries. Java is a trademark or registered trademark of Sun Microsystems, Inc.

explaining how information should look. Open standards are developed by consensus in an industry group. There is a tremendous variation in the membership rules of processes for these organizations, and they range from official organizations like the International Organization for Standardization, or ISO (http://www.iso.ch), to small vertical industry groups. STEP is an example of an open standard. It is developed by ISO, with the help of industrial consortia such as PDES, Inc. (http://pdesinc.aticorp.org) and ProSTEP (http://www.prostep.de). XML and UML are also open standards, even though they are not ISO standards. XML is developed by the World Wide Web Consortium (W3C, http://www.w3.org), and UML is developed by the Object Management Group (OMG, http://www.omg.org).

2. **Industry Standards** are technologies that are commonly used, but are not open or democratically managed by a group of users. The Java™ technology is a well-known example of an industry standard. There are a number of companies involved in the Java Community Process (http://jcp.org), but because one company wields a tremendous amount of control over the process, Java is classified as an industry standard, not an open standard.

3. **De facto Standards** are in wide use because of their value or association with other technologies, and not necessarily because they were produced by a standards organization. A commercial software product may be a *de facto* standard because of its wide adoption. The Microsoft Windows operating system is a *de facto* standard for personal computers. The Simple Object Access Protocol (SOAP) [5] was initially a *de facto* standard, because of its broad use in Web services, though it has now been formalized as an open standard in the W3C. *De facto* standard status does not mean that there are no alternatives to a particular technology; such alternatives are just rarely used.

In addition to the three types of standards mentioned above, there is open source software. Open source software is not necessarily an open standard. Open source refers to software source code that is available to the general public and does not have licensing restrictions that limit use, modification, or redistribution under the same terms as the license of the original software[2]. The GNU/Linux operating system (http://www.linux.org) and Eclipse software development environment (http://www.eclipse.org) are examples of open source technologies. Some companies frequently release software as open source when they want to lower the barrier of entry for certain technologies. The XML4J (XML for Java) XML parser (http://www.alphaworks.ibm.com/tech/xml4j) and the Apache project's SOAP implementation (http://ws.apache.org/soap) are two such examples.

### 1.2. STEP: Powerful Content Models

ISO 10303, also informally known as the Standard for the Exchange of Product model data (STEP), is a family of standards defining a robust and time-tested methodology for describing product data throughout the lifecycle of the product. STEP is widely used in Computer Aided Design (CAD) and Product Data Management (PDM) systems. Major aerospace, automotive, and ship building companies have proven the value of STEP through production implementations resulting in savings of $150M per year in the US [6] [7]. STEP contains the high-quality and high-fidelity information models many XML business applications require.

But the STEP information modeling language is not based on an XML vocabulary. The objects to be represented and exchanged using STEP, as well as the associations between these objects, are defined in schemas written in EXPRESS (ISO 10303-11) [8], a modeling language combining ideas from the entity-attribute-relationship family of modeling languages with object modeling concepts. The following EXPRESS schema models two-dimensional drawings consisting of points and lines. Although this example is very simple, it illustrates several EXPRESS language features. These include:

- Built-in simple types such as STRING and REAL.
- Constructs representing collections such as SET.
- Inheritance of types (*point* and *line* both inherit properties from *shape*).

```
SCHEMA example;

ENTITY drawing;
   name : STRING;
   elements : SET [1:?] OF shape;
END_ENTITY;

ENTITY shape;
   label : STRING;
END_ENTITY;

ENTITY point SUBTYPE OF (shape);
   x : REAL;
   y : REAL;
END_ENTITY;

ENTITY line SUBTYPE OF (shape);
   end1 : point;
   end2 : point;
END_ENTITY;

END_SCHEMA;
```

EXPRESS actually has more capabilities than this very simple example suggests. EXPRESS can represent complex inheritance relationships and functions, and includes a rich set of constructs for specifying constraints on populations of instances. In the interest of brevity we do not focus on advanced EXPRESS language features in this paper. However, as an example of a relatively simple EXPRESS "WHERE" constraint, consider the following definition for a point on a parabola represented by the equation $y = x^2$:

```
ENTITY point_on_parabola SUBTYPE OF (point);
WHERE
   parabola : y = x**2;
END_ENTITY;
```

"WHERE" expressions can be far more complicated than the parabolic equation above. For example, the following is a "WHERE" constraint on ordinal dates from an actual STEP standard [9]. The constraint enforces the rule that the date's *day_component* must be an integer from 1 to 365 or, if the year

[2] See http://www.opensource.org/docs/definition_plain.php for a more detailed definition of "open source".

is a leap year, from 1 to 366. *leap_year* is an EXPRESS function (not shown in the example) defined to compute whether the date's *year_component* is a valid leap year.

```
WHERE
   wr1: (((NOT leap_year(SELF.year_component))
      AND (1 <= day_component)
      AND (day_component <= 365))
      OR (leap_year(SELF.year_component)
      AND (1 <= day_component)
      AND (day_component <= 366)));
```

Although EXPRESS is a powerful language, it is relatively unknown to most programmers. Moreover, STEP data (i.e., an instance population of an EXPRESS schema) are typically exchanged using an ASCII character-based syntax defined in ISO 10303-21 (also known as "Part 21" of STEP) **[10]**. The Part 21 syntax, although adequate for the task at hand, lacks extensibility, is hard for humans to read, and - perhaps most limiting - is computer-interpretable only by software supporting STEP.

For example, consider the drawing shown in Figure 1. A Part 21 instance population that represents this information is as follows:

```
#10 = point ('P01', 2.0, 2.0);
#20 = point ('P02', 5.0, 2.0);
#30 = point ('P03', 5.0, 4.0);
#110 = line ('L01', #10, #20);
#150 = line ('L02', #10, #30);
#200 = drawing ('Lines and Points',
       (#10, #20, #30, #110, #150));
```

First the type of instance is indicated (e.g., point) followed by a list of attribute values (where the list is ordered based on the attributes sequence in the EXPRESS definition of the entity).

Hash numbers like #10 and #20 denote object instance identifiers so that instances can be easily referenced and used elsewhere in the population. For example, the #10 and #30 used in the line L02 instance indicate that its *end1* and *end2* properties correspond to P01 and P03 respectively. Hash numbers are valid within the scope of a single STEP instance population (usually a text file), and thus their actual values are arbitrary as long as they are consistently used within the population.

Aggregate members are enclosed in parentheses as exemplified in the #200 drawing instance, where the hash numbers of all five drawing elements in Figure 1 are contained.

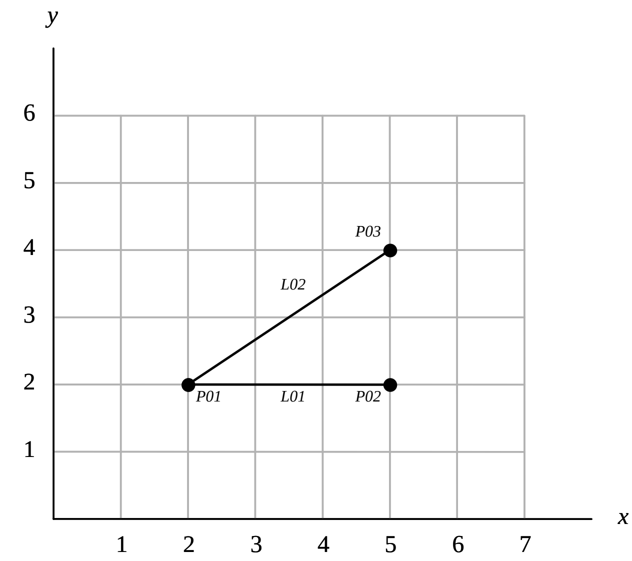


**Figure 1 - Drawing consisting of two lines connecting three points.**

### 1.3. XML and UML: Ubiquitous Tools

Unlike the STEP Part 21 syntax, XML is easily extensible and is supported by numerous inexpensive and widely used software tools. Thus, from the perspective of a typical programmer, it is easier to render XML data into forms that are suitable for human perusal. Many applications developed today that import or export data use or support some form of XML format. Finally, XML is used in virtually all new work done on developing standard data formats for many domains, including some domains that are within the scope of STEP. So it is obvious that STEP needs to accommodate itself to XML.

UML is a widely accepted and supported standard software modeling language. UML tools abound. By contrast, the tools that are used to develop and manage STEP schemas and instance populations have a relatively small user community. STEP modeling tools are adept at the EXPRESS language and can develop and validate very complex information. However, while EXPRESS is a powerful information modeling language, it has traditionally been relatively unknown in the world of general software modeling methods. For software developers who need to deal with STEP instances, it would be a tremendous boon to be able to capture, model, and visualize the relationships between STEP constructs and the other information types that they use in their development process. UML, though its class diagrams and through its profile extensibility mechanism, is already being used to visually represent XML schemas [11]. As we shall see, the capability to visualize STEP constructs in UML will be available soon, as methods for integrating EXPRESS schemas with UML models are now emerging.

## 2. CAN STEP WORK WITH XML AND UML?

In order to capitalize on XML's popularity and flexibility, and to accelerate STEP's adoption and deployment, ISO is developing a standard for representing EXPRESS schemas and instance populations in XML. The expectation is that this emerging standard, ISO 10303-28, *Implementation methods: XML Schema governed representation of EXPRESS schema governed data* [11] [13], will not only enable developers to use low-cost, ubiquitous XML software tools to implement file-based exchange and visualization of STEP instances, but can also potentially facilitate the use of STEP information in emerging areas such as XML-based Web Services.

Our "lines and points" drawing example from Section 1.2 can be represented as an XML document conforming to a Part 28-generated XML schema. The resulting lexical representation might look like this:

```
<p28data>
  <point>
    <label>P01</label>
    <x>2.0</x>
    <y>2.0</y>
  </point>
  <point>
    <label>P02</label>
    <x>5.0</x>
    <y>2.0</y>
  </point>
  <point>
    <label>P03</label>
    <x>5.0</x>
    <y>4.0</y>
  </point>
  <line>
    <label>L01</label>
    <end1 ref="P01"/>
    <end2 ref="P02"/>
  </line>
  <line>
    <label>L02</label>
    <end1 ref="P01"/>
    <end2 ref="P03"/>
  </line>
  <drawing>
    <name>Lines and Points</name>
    <elements>
      <point ref="P01"/>
      <point ref="P02"/>
      <point ref="P03"/>
      <line ref="L01"/>
      <line ref="L02"/>
    </elements>
  </drawing>
</p28data>
```

Even in this simple example one can see relative benefits and weaknesses in different approaches for representing the same or similar information. For the Part 21 instance omits the names of properties such as *x*, *y*, *end1*, and *end2*. The Part 28 instance, on the other hand, includes this information. Hence, Part 21 instances are more concise than Part 28 instances (resulting in smaller file sizes), whereas Part 28 instances are more human readable.

An entire W3C XML Schema [14] generated from the EXPRESS using Part 28 would be a bit too verbose to reproduce for this paper, but the element definitions for *p28data* and *drawing* might look like this:

```
<xs:element name="p28data">
  <xs:complexType>
    <xs:choice minOccurs="1" maxOccurs="unbounded">
      <xs:element name="drawing" type="Drawing"/>
      <xs:element name="point" type="Point"/>
      <xs:element name="line" type="Line"/>
    </xs:choice>
  </xs:complexType>
</xs:element>

<xs:complexType name="Drawing">
  <xs:sequence>
    <xs:element name="name" type="xs:string"/>
    <xs:element name="elements">
      <xs:complexType>
        <xs:choice maxOccurs="unbounded">
          <xs:element name="line" type="Line-ref"/>
          <xs:element name="point" type="Point-ref"/>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

The ISO STEP committee (TC184/SC4) is also developing another standard, *ISO 10303-25*, *EXPRESS to OMG XMI binding* [15] (also known as Part 25), for transforming EXPRESS schemas into UML models. This will enable developers to use their familiar UML tools to see the contents of STEP (EXPRESS) schemas and eventually to specify relationships between STEP information models and the other UML models that they use. A Part 25 mapping from our EXPRESS schema to the XML Metadata Interchange (XMI®) format [16] would produce a UML class diagram as shown in Figure 2.

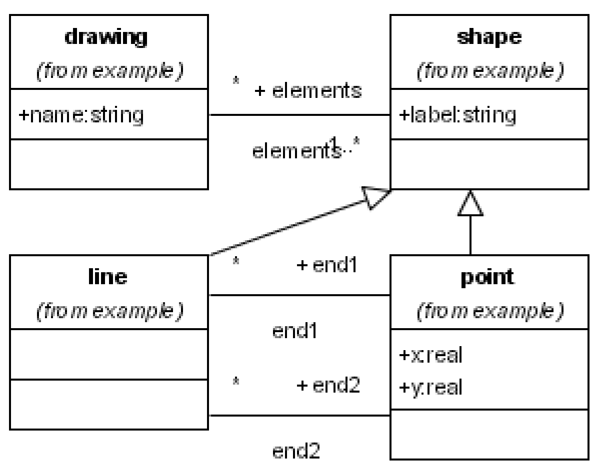


**Figure 2 - UML class diagram obtained from "lines and points" EXPRESS via Part 25.**

So the answer to the question "Can STEP work with XML and UML?" is *yes*. Capabilities for STEP to interoperate with and be integrated with XML and UML are now emerging.

 

## 3. SOFTWARE TOOLS FOR STEP AND XML

Until recently, there were few software tools for using STEP schemas and instance populations in the XML and UML worlds. However, there are now several promising development efforts underway to create software for STEP that capitalizes on the popularity of XML and UML.

- The STEP Module Repository (http://stepmod.sourceforge.net) is a collection of resources tagged in XML to serve as the core of a modular environment for developers of STEP and related standards. The Extensible Style Language Transformation (XSLT) standard [17] is used to produce both ISO-compliant as well as developer-friendly documentation.
- The "EXPRESS For Free" (*exff*) project (http://exff.sourceforge.net) is developing tools to convert between EXPRESS and UML. The initial goal is to be able to use UML-based code generation tools to help implement STEP. Future plans include supporting the use of UML modeling tools to build EXPRESS schemas. The current *exff* release provides translators between XMI and EXPRESS marked up in XML using the STEP Module Repository Document Type Definition.
- The National Shipbuilding Research Program (http://www.nsrp.org) has implemented a translator for its Integrated Shipbuilding Environment from STEP Application Protocol AP218 (Ship structures) to AP209 (Composite and metal structural analysis and related design). The translator uses Part 28 to represent both APs and XSLT to convert from one to the other.
- Various STEP software vendors are developing Part 28 EXPRESS-to-XML translators and/or have products that import or export product model data in an XML format. For example, one vendor not only is supporting Part 28 but also uses XML in several ways, including for converter configuration settings and as user interface specifications for an EXPRESS-based application for AP210 (Electronic assembly interconnect and packaging design).

Meanwhile, some proprietary tools and formats that compete with STEP are emerging:

- JT, a proprietary format based on the Jupiter technology [18], is intended to be an efficient format for the visualization of large 3D models. The JT format is optimized for visualization and interaction of large 3D models. The scene graph can also contain different levels of detail to minimize the memory footprint while viewing large assemblies. A JT file may also include assembly structure, text and symbolic annotations (e.g., tolerances). A recently formed JT Consortium (http://www.jtopen.com/) is attempting to make JT an industry standard.
- PLM XML (http://www.eds.com/products/plm/xml/) is a proprietary XML format for enabling product lifecycle interoperability. It is based on W3C XML Schemas. PLM XML contains product information and geometric representation data.

## 4. SUITES OF STANDARDS: HOW THE PIECES FIT

While it is useful to compare the expressiveness and richness of information modeling languages, it is perhaps even more important to understand:

- What the roles of each standard technology are.
- The quantity, quality, scope, and interoperability of standard schemas that exist in a given technology, and the degree to which such schemas have been implemented and deployed.
- How different technologies can be used to complement each other.

| Schema Language | Mapping Language | Serializations (lexical) and Interface Methods (APIs, ...) | Standardized Content Schemas |
|---|---|---|---|
| Express | Express-X | Part 21, Part 28 ed.1 and 2 (XML), Part 25 (XMI), etc. | ISO 10303 series (STEP) et al. O(1000) man-years of effort & O(10,000) standardized engineering & product concepts |
| UML (XMI, …) | QVT[3] | Web (XSP[4]), SOAP, WSDL, CORBA, PB[5] | UML Profiles emerging (e.g., SysML[6] for systems engineering) |
| XML Schema, DTD | XSLT | Part 28 ed.2 (XML Schema), ... | MathML[7], MatML[8], FemML[9], ChemML[10], SVG[11], PDTnet[12], etc. |

**Table 1 - Primary technologies for schema-based engineering frameworks.[13]**

The right-most column in Table 1 - identifies example schemas for the indicated information modeling technologies (each row), and it estimates the capabilities and investments in the STEP family of standards. It shows how STEP provides on the order of 10,000 standardized concepts for engineering (see "STEP-on-a-Page" at http://www.nist.gov/sc5/soap/ -- note that it is now actually two pages since the advent of STEP Modules, which define small, reusable information models that are employed in combinations to support business processes). On the order of 1000 person-years of effort has been required to create this coordinated set of standards and gain consensus over the past fifteen-plus years. It would be cost-prohibitive and unnecessary to start from scratch and try to re-invent that capability using just XML directly.

[3] MOF 2.0 Query/Views/Transformations RFP, OMG Document ad/02-04-10 (http://www.omg.org/cgi-bin/doc?ad/02-04-10).
[4] Web (XSP) = "X" Server Pages (where X can be A, J, P, etc.).
[5] PB = Perspective Broker (see: http://twistedmatrix.com ).
[6] http://www.sysml.org
[7] http://www.w3.org/Math
[8] http://www.matml.org
[9] http://www.istos.org/femML, see also [19]
[10] http://www.xml-cml.org
[11] http://www.w3.org/Graphics/SVG
[12] http://www.pdtnet.org
[13] Based on Engineering Framework Interest Group (http://eislab.gatech.edu/efwig) emails from Stephen Waterbury (dated July 13, 2002) and David Leal (dated Nov. 26, 2002).

Instead, there are several ways to interface to STEP-based schemas using XML-based schemas, thereby getting the best of both worlds. For example, a simple XML layer can get information into and out of more complete STEP-based rich product models. And now there are parts of STEP that are specifically designed to produce XML and UML models, as noted above in section 2.

Furthermore, online repositories of purely XML standards (e.g., the *Cover Pages* directory of XML applications at http://xml.coverpages.org/xmlApplications.html), list only a few dealing with engineering and technical topics also addressed by STEP (e.g., architecture engineering construction, math, chemistry), and many areas that STEP does not deal with (e.g., legal applications).

There appears to be an opportunity for the best of both worlds to work together and create more complete product models and system models. For example, Figure 3 envisions the complementary usage of STEP, UML, and XML for systems engineering. Tools that work with the UML-based SysML will be able to interoperate with AP233-based systems engineering models and a variety of domain-specific models facilitated by STEP (e.g., AP210 for electrical CAD/CAE and connections with mechanical CAD).

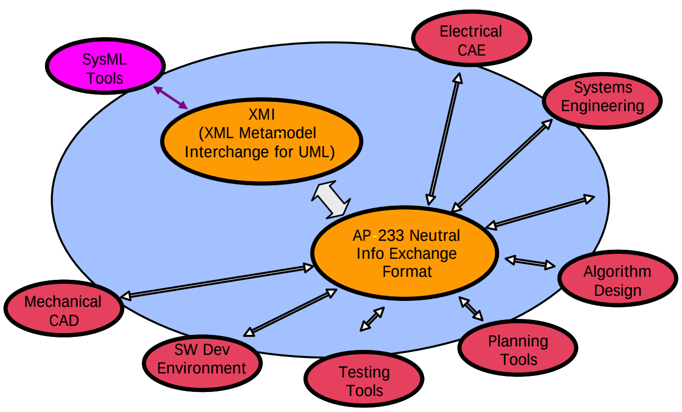


**Figure 3 - Complementary usage of STEP, UML, and XML for Systems Engineering: Envisioned AP233- SysML[14] relationship.**

## 5. STEP AND WEB SERVICES

Web Services are a technology for providing access to applications via the Internet, either for human interaction or for automated system-to-system interaction. Some likely candidates for commercial Web Services useful to enterprises could include:

- On-line catalogs
- Directories of commercial services
- Data management services
- Inter-enterprise transaction services
- Computing services

Within an enterprise or a virtual enterprise (such as an original equipment manufacturer and their supply chain), Web Services could provide the next generation of trading partner business communications, which might very well include some types of interactions that are currently conducted either as paper transactions or by relatively primitive, human-mediated file-transfer interactions.

What role should STEP play in the world of Web Services? While STEP has traditionally been used primarily in file transfer scenarios, STEP can also provide support for more agile and flexible Web Services applications.

Numerous inter- and intra-enterprise communications involve product data that has been modeled in STEP, such as:

- Requests for quotes
- Requests for proposals
- Technical data package management
- Work orders
- Engineering change requests
- Engineering change orders

These exchanges and others can be implemented as Web Services using STEP-defined standard data structures. STEP-based Web Services can enhance these processes by supporting whatever balance of machine automation/validation and human verification is optimal for such interactions.

One view of how STEP/EXPRESS-based product models and web services can be architected has been articulated by Martin Hardwick (STEP Tools, Inc. Newsletter Nov. 2003):

*The new [STEP] Part 28/XML standard will enable the definition of XML Schemas from the STEP Application Protocol mapping tables. This will make the definition XML data for STEP much more straightforward and easy to follow because these tables frequently restrict the large range of cases allowed in an EXPRESS model to one or two specific cases for a particular mapping.*

*A three-level product model stack is emerging for representing STEP data on the web with EXPRESS defining the lowest level, XML Schema the second level and RDF the top level. Soon this architecture will enable distributed STEP databases where engineers use search and integration engines to identify compatible products and processes on the World Wide Web.*

## 6. FURTHER WORK: BRIDGING THE STEP, XML, AND UML WORLDS

Further work is clearly needed in this area and several issues require more investigation. Some issues may be solvable near-term, while other aspects will require more time and resources.

Efforts are already underway to add Part 28 functionality to the STEP Module Repository to produce "equivalent" XML

[14] SysML Partners (Nov. 19, 2003) SysML Overview. Presentation to OMG Analysis and Design Task Force. http://www.sysml.org

schemas for portions of the Product Life Cycle Support (PLCS) family of STEP standards (http://www.oasis-open.org/committees/plcs), and use the module repository to produce XML-based specifications based on the XML schemas. If this project is successful, then XML developers without any knowledge of EXPRESS will be able to build STEP implementations. And we would gain some practical examples of how the EXPRESS world and the XML world can coexist and even benefit from each other's strengths.

Efforts are also ongoing to integrate/combine STEP with "Semantic Web" technologies. **[20]** Specifically, a mapping from STEP/EXPRESS to the Web Ontology Language (OWL) [21] has been proposed. The exff software distribution (see Section 3) includes an XSLT implementation of the mapping.

The following areas may merit further research.

### 6.1. Additional Comparison of STEP, XML, and UML

Good analogies could help people understand the technologies and the issues with various information and knowledge representation approaches. For example, older traditional programming languages can continue to thrive in the midst of newer languages due to significant libraries of code and the volume of related legacy applications (e.g., FORTRAN is alive and well due to its large embedded base of useful numerical routines, and its interfaces to other newer programming languages). Comparing modeling languages like XML, STEP/EXPRESS, and UML could be analogous to comparing traditional languages like C++ and Java. Their modeling features could be compared as well as their usage and technology factors such as related libraries, suitable application characteristics, developer base and popularity, and ease-of-use.

In other words, what are proper metrics and methods for comparing modeling languages like XML, STEP/EXPRESS, and UML (and comparing their associated collections of standards) and recommending where best to use what technology? And are there reports that compare XML, STEP/EXPRESS, and UML in terms of their expressive constructs, cost of modeling, available tools, developer base, etc. (i.e., various factors related to their total cost of ownership, capabilities, and adoption within industry and government)?

Additionally, it would be useful to describe the status of STEP/XML/UML in these timeframes:

- Their current status (specifications, tools, interrelations between specifications, etc.).
- Their likely near-term status (the likely state six months from now).
- Their target status over the next few years and recommended actions to achieve those targets.

### 6.2. Evaluation of Competing Technologies

Further investigation of PLM XML (mentioned in Section 3) would be useful - how similar is it to related STEP standards such as the STEP Product Data Management modules? Is it complete? Are the corresponding STEP standards more robust and/or more capable?

The STEP developer community could also describe and illustrate how it is often true (we believe) that EXPRESS-based standards like ISO 10303 are a superset of typically smaller scoped native XML-based standards. For example, we believe GenCAM (XML for electronics – http://www.gencam.org) is largely a subset of AP210 (http://ww.ap210.org) in terms of its representation capabilities (i.e., there are numerous design concepts and features that AP210 can represent which GenCAM cannot). It would be useful to make such comparisons in a structured manner, give guidelines how to do similar comparisons in other areas, and give recommended practices how to handle such situations (e.g., show how such standards can then work together).

### 6.3. Further Investigation and Usage of XML Core Technologies

XML is already being used extensively in the Module Repository (Section 3) to streamline the development and publication of modular STEP specifications. A description of how XML technology can help developers of standards like STEP (including how it is doing so now in the case of the Module Repository) would be useful.

Additionally, alternatives to the XML schema languages currently being used to represent EXPRESS should be investigated. RELAX NG **[22]**, a powerful yet easy-to-use schema language for XML, could substitute for the W3C XML Schema language in STEP Part 28 mappings. Another alternative worthy of consideration is Schematron **[23]**, a language for making assertions about patterns in XML documents. EXPRESS "WHERE" constraints, which are currently outside the scope of Part 28 – and are often impossible to specify as W3C XML Schema language definitions or RELAX NG patterns, could in some cases be represented as Schematron rules. For example, the following Schematron schema represents the "WHERE" constraint from the *point_on_parabola* EXPRESS definition from Section 1.2:

```
<schema xmlns="http://www.ascc.net/xml/schematron"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <pattern name="On parabola">
    <rule context="p28data/point">
      <assert
        test="number(y) = number(x) * number(x)">
Point not on parabola defined by the equation y=x**2.
      </assert>
    </rule>
  </pattern>
</schema>
```

None of the three points in Figure 1 satisfy the schema's rule. Therefore, when this Schematron schema is applied to the XML lexical representation of our "lines and points" example (see Section 2), a Schematron validator generates an error message for each of the three points. Figure 4 shows a screen shot of Schematron validator-created browser output.

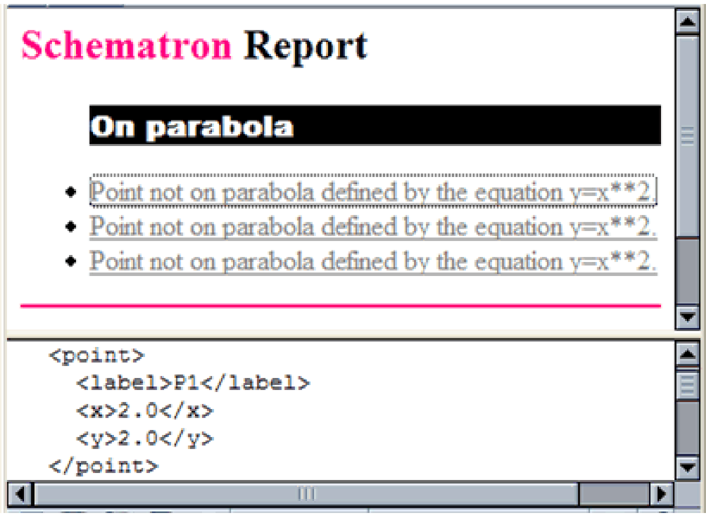


**Figure 4 - Schematron diagnostics for "lines and points" lexical data.**

## 7. SUMMARY

STEP provides a large body of standardized, rigorously-defined, high-fidelity technical concepts. Although STEP's quality compares favorably with that of other data exchange standards, traditional description and implementation methods for STEP (EXPRESS and Part 21) have failed to achieve the popularity of XML and UML. Thus, STEP,XML, and UML are complementary technologies. Together, they are a powerful force for lowering the barriers to widespread exchange and sharing of digital data. Emerging XML and UML-based STEP implementation technologies and current projects bridging the STEP and XML/UML worlds show great promise to enable greater PLM interoperability.

## ACKNOWLEDGMENTS

The authors wish to thank the PDES, Inc. team, David Price, Allison Barnard Feeney, Matthew Aronoff, as well as the *2004 ASME Computers and Information in Engineering Conference* reviewers for their helpful comments on earlier drafts of this paper.

## REFERENCES

[1] World Wide Web Consortium, *Extensible Markup Language (XML) 1.0*, W3C Recommendation 04 February 2004, http://www.w3.org/TR/REC-xml/.

[2] ISO 10303-1:1994, *Industrial automation systems and integration - Product data representation and exchange- Part 1: Description methods: Overview and fundamental principles*.

[3] Kemmerer, S. (ed.), *STEP: The Grand Experience*, National Institute of Standards and Technology, NIST SP 939, July 1999 http://www.mel.nist.gov/msidlibrary/doc/stepbook.pdf

[4] Object Management Group, *OMG Unified Modeling Language Specification*, Version 1.5, March 2003, http://www.omg.org/technology/documents/formal/uml.htm.

[5] World Wide Web Consortium, *SOAP Version 1.2 Part 0: Primer*, W3C Recommendation 24 June 2003, http://www.w3.org/TR/soap12-part0/.

[6] PDES, Inc., *STEP Success Stories*, http://pdesinc.aticorp.org/success_stories.html.

[7] M. P. Gallaher, A. C. O'Connor, T. Phelps (Dec. 2002), *Economic Impact Assessment of International Standard for the Exchange of Product Model Data (STEP) in Transportation Equipment Industries*, NIST Planning Report 02-5, http://www.mel.nist.gov/msid/sima/step_economic_impact.pdf

[8] ISO 10303-11:1994, *Industrial automation systems and integration—Product data representation and exchange—Part 11: Description methods: The EXPRESS language reference manual*.

[9] ISO 10303-41:2000, *Industrial automation systems and integration—Product data representation and exchange—Part 41: Integrated generic resource: Fundamentals of product description and support*.

[10] ISO 10303-21:2002, *Industrial automation systems and integration—Product data representation and exchange—Part 21: Implementation methods: Clear text encoding of the exchange structure*.

[11] D. Carlson, *Modeling XML Applications with UML: Practical e-Business Applications*, Addison-Wesley, 2001.

[12] ISO TC184/SC4/WG11 N223, ISO/WD 10303-28, *Product data representation and exchange: Implementation methods: XML Schema governed representation of EXPRESS schema governed data*, 2004-02-17.

[13] J. Lubell, *From Model to Markup: XML Representation of Product Data*, XML 2002 Conference, Baltimore, December 2002, http://www.mel.nist.gov/msidlibrary/doc/m2m.pdf.

[14] World Wide Web Consortium, *XML Schema Part 0: Primer*, W3C Recommendation 2 May 2001, http://www.w3.org/TR/xmlschema-0/.

[15] ISO TC184/SC4/WG11 N204, ISO/CD TS 10303-25, *Product data representation and exchange: Implementation methods: EXPRESS to XMI binding*, 2003-05-02.

[16] Object Management Group, *OMG XML Metadata Interchange Specification*, Version 1.2, January 2002, http://www.omg.org/technology/documents/formal/xmi.htm.

[17] World Wide Web Consortium, *XSL Transformations (XSLT) Version 1.0*, W3C Recommendation 16 November 1999, http://www.w3.org/TR/xslt/.

[18] Bartz, D., et al., *Jupiter: A Toolkit for Interactive Large Model Visualization*, in *Proceedings of Symposium on*

*Parallel and Large Data Visualization and Graphics* (2001), pp. 129-134.

[19] Michopoulos, J.G., *Development of the Finite Element Modeling Markup Language*, in *Proceedings of ASME 2002 DETC and CIE Conferences*, Montreal, Canada, 2002.

[20] Price, D., *A Brief Foray into Semantic Web Technology and STEP*, October 2003, http://www.exff.org .

[21] World Wide Web Consortium, *OWL Web Ontology Language Overview*, W3C Recommendation, 10 February 2004, http://www.w3.org/TR/owl-features.

[22] ISO/IEC FDIS 19757-2:2002, *Document Schema Definitions Languages (DSDL —Part 2: Regular-grammar-based validation—RELAX NG*, http://relaxng.org/.

[23] Jelliffe, R., The Schematron Assertion Language specification, version 1.5, 2002-10-01, http://www.schematron.com.