Journal of Computing and Information Science in Engineering

Technical Notes

This section contains shorter technical papers. These shorter papers will be subjected to the same review process as that for full papers.

Interoperability Standards in the Semantic Web

Steven R. Ray
e-mail: ray@nist.gov
National Institute of Standards and Technology,
100 Bureau Drive, STOP 8260,
Gaithersburg, MD 20899-8260

The growth in the use of the Internet brings with it an increase in the number of interconnections among information systems supporting the manufacturing supply chain as well as other businesses. Each of these interconnections must be carefully prescribed to ensure interoperability. However, the sheer number of interconnections and the resulting complexity threaten to overwhelm the ability of the standards community or industry to provide the necessary specifications—a way out of this impasse must be found. This paper outlines the elements of an approach and the technology to move toward self-integrating systems, wherein the systems negotiate meaningful interfaces as needed in a dynamic environment. [DOI: 10.1115/1.1480024]

Introduction

Industrial manufacturing today is as competitive as it ever was, but in new ways. In the past, having a better design or manufacturing process was often the key to a company's prosperity. Today, quality practices and quality products are assumed; the determining factor is shifting toward other aspects of manufacturing—agility, lean manufacturing practices, information management, and effective use of the supply chain. The book ''A Stitch in Time'' [1] chronicles how the American apparel industry managed to exploit information technology in order to survive against low-cost competition.

To equip tomorrow's manufacturers with the tools and capabilities they will need, therefore, we must speculate on what the dominant and determining trends will likely be. If indeed the ability of a company to manage its information and its partners turns out to be the critical determinant of success, then neutral, manufacturing-information standards for the vast amount of data shared among partners will become of paramount importance.

But will these information standards look like today's standards? We think not. For one reason, it is becoming impractical for large communities to meet physically each time the need to share information becomes important. Secondly, the demand for more interconnected applications is increasing rapidly, and the timeframe for the required solutions is decreasing. Finally, as industry's appetite for increasingly specific information grows, so does the complexity of the standards.

As will be described in this paper, one other trend that will transform the manufacturing landscape is the emergence of computer programs that ''understand'' the information they are manipulating, rather than simply crunching data and displaying results. There are some prerequisites for this transformation to occur, but once they are in place, we can expect to see a sudden growth in enterprise activity and productivity equivalent to the effect the Internet and the World Wide Web has had on providing people with information.

World Wide Web 2—The Semantic Web

The Internet and the World Wide Web have brought about tremendous changes in the way we interact, conduct commerce, and retrieve information. However, during all of this rapid growth, computers have been shipping data to each other mindlessly. They have no understanding of what the data means. The Web has exploded as **people** were suddenly given access to vast amounts of information by computers that have been little more than improved versions of telephones. Soon however, computers **will** be able to ''understand'' (and reason about) the meaning of the data they are manipulating. Then, computer applications will demonstrate the same fundamental ground-shift that people have with the advent of the World Wide Web. Why will this be so?

Most information on the Web is simply text for people to read. That's great for people, and even for B2C (business-to-consumer) electronic commerce activities such as browsing catalogs of goods. But the real revolution that's underway is in B2B (business-to-business) commerce—that's where the real money is.[1] Furthermore, the true impact will take place when computers transact business with other computers.[2] They will be communicating about carefully defined things, and they will not tolerate ambiguity as to what the terms of a contract are. However, before this can take place, there are a few pieces of technology missing.

In ''Weaving the Web,'' [2] Tim Berners-Lee speaks of a dream for the Web:

''In the first part, the Web becomes a much more powerful means for collaboration between people . . . In the second part of the dream, collaborations extend to computers . . . A 'Semantic Web' which should make this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy,

Contributed by the Engineering Informatics (EIX) Committee for publication in the JOURNAL OF COMPUTING AND INFORMATION SCIENCE IN ENGINEERING. Manuscript received Jan. 2002; Revised Mar. 2002, Associate Editor: R. Rangan.

[1] The U.S. Census Bureau, in its ''first official snapshot of e-commerce activity'' states ''Census Bureau data show that business to business (B-to-B) e-commerce dominated 1999 e-commerce activity . . . Manufacturing led all industry sectors with 1999 e-commerce shipments that accounted for 12.0 percent ($485 billion) of the total value of manufacturing shipments.'' (see http://www.census.gov/econ/estats/papers/estatstext.pdf)

[2] Intel, for example, describes a multi-phase introduction of information technology for B2B, consisting of a ''Web Order Management'' system giving customers real-time access to the company's Enterprise Resource Planning systems (thus by-passing a static catalog and giving human customers direct access to real-time data), and a ''Supply Line Management'' system that automates inventory replenishment (thus going a step further and connecting customer's automated systems to Intel's automated systems). See http://www.intel.com/eBusiness/pdf/busstrat/plan/hi011003.pdf

and our daily lives will be handled by machines talking to machines, leaving humans to provide the inspiration and intuition . . . The first step is putting data on the Web in a form that machines can naturally understand, or converting it to that form.''

Computers are extremely pedantic, and have great difficulty using loosely defined terms. For this reason, they are limited in their ability to navigate Web material intended for our flexible human minds. What Berners-Lee is talking about is the need for a rigorous means of representing and conveying information over the Web that will be well-suited to an audience of computers.

The Standards World

The world of voluntary national and international normative[3] standards has also been evolving in the presence of the growth of the Internet and the widespread use of computers in personal and business life. Early on, these standards were in the form of protocols for moving information around, such as the ASCII[4] standard for encoding letters and symbols used by teletype machines, or even the pervasive TCP/IP[5] networking protocol standard that drives the Internet itself. These standards typically focus on the way in which information is to be encoded (the syntax) and only peripherally describe the nature of the information being standardized (the content). Traditionally, such standards are specified in terms of English prose. Specifications of ISO (the International Organization for Standardization) standards are typically highly structured texts intended for a human reader.[6]

Increasingly, the kinds of information structures being standardized today are much more complex than they were even a decade ago. Syntactic specifications have begun to be defined in a computer-readable form such as the Express language[7] or XML.[8] ISO 10303, (informally known as the Standard for the Exchange of Product Model Data), is a specification that describes how computer-based design information is to be shipped from one computer aided design (CAD) system to another, and at last count included over 30,000 definitions. This kind of standard distinguishes between the information model that describes the information content, and the encoding mechanism that specifies the syntax. In doing this, the need for rigorous definitions of terms has become even more apparent, since the intended producers and consumers of the information are computer programs. Thus, it is becoming clear that information standards in the future will need two additional components: unambiguous definitions of terms, and a rigorous, computer-readable means of stating these definitions.

Teams of researchers are tackling these new requirements in the form of definition languages such as the Resource Description Framework (RDF) proposed by the World Wide Web Consortium (W3C) and the DARPA Agent Markup Language (DAML)[9] + the Object Interchange Layer (OIL),[10] sometimes referred to as DAML+OIL. At the same time, others are working on standardizing the definitions themselves, such as the Standard Upper Ontology[11] work.

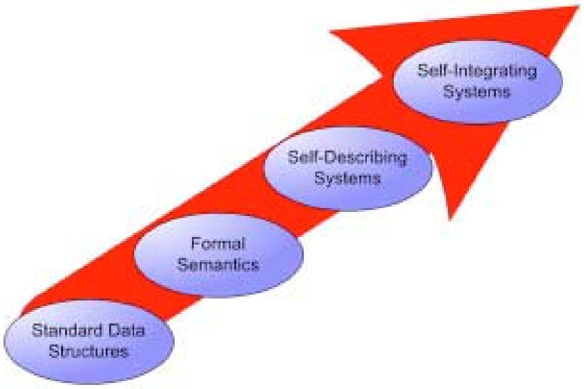

Fig. 1 Evolution of systems integration approach

But the pursuit of well-defined terms is more than just an attempt at engineering elegance. Yes, it is true that this greater degree of rigor will improve the interoperability of systems, but there is another factor that will eventually overwhelm the standards world—time. The number of communication standards is growing geometrically,[12] driven partly by the growth of communication technologies, but also by the increasing drive to integrate formerly separate functions within the business community. Practically speaking, the standards community (whether formal or informal) will have to change its way of doing business because new requirements are outpacing the ability to keep up. It will not be possible to convene a committee of all relevant players each time a new data interchange need is recognized.

Examples of this changing operating model for standards groups already exist. The Object Management Group (OMG), an industry consortium that ''was formed to create a component-based software marketplace by hastening the introduction of standardized object software'' calls for members to come forward with solutions to the consortium's requirements, rather than trying to create solutions themselves. This saves time, since OMG only has to spend time picking, or possibly slightly modifying, the best solution. But even with a standard specified, there is still much work to be done before it can be used successfully. Computer programs that conform to the standard must be written, and their conformance to the standard must be tested. The complexity of testing conformance to a standard grows with the complexity of the standard itself. And finally, if the standard is about connecting one computer system with another (either at the network level or at the information content level) the interoperability of systems that conform to the standard must be tested to see if ambiguity within the standard itself leaves room for multiple incompatible interpretations. All of this takes time, more time than the current pace of progress allows. As a result, software vendors and users are forced to rush into implementations of systems where large uncertainties still exist, and must patch in solutions as problems arise.

So, even if we have completely unambiguous definitions of terms, the standards bodies will eventually fail to keep up with the world's appetite for information exchange protocols. A fundamental change in how systems come to communicate with each other is needed. We are calling this new approach ''self-integrating systems.''

Figure 1 shows a predicted evolution from traditional data-

[3] Normative standards, in contrast to regulatory standards (such as environmental limits) or metrology standards (such as realization of the fundamental units of length, mass, etc.), have to do with specifying how one represents something, or how one carries out a procedure. Network protocols, Web-based formats such as HTML and XML, are all examples of normative standards.

[4] The American Standard Code for Information Interchange. See also ISO-14962-1997 and ANSI-X3.4-1986 (R1997).

[5] TCP/IP stands for Transmission Control Protocol/Internet Protocol. For a clear and concise description of TCP/IP, see http://pclt.cis.yale.edu/pclt/COMM/TCPIP.HTM.

[6] See, for example, ISO Directive, Part 3: 1997 3rd edition, ''Rules for the structure and drafting of international standards'' at http://www.iso.ch/iso/en/ISOOnline.openerpage.

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

[8] See http://www.w3.org/XML/.

[9] See http://www.daml.org

[10] See http://www.ontoknowledge.org/oil

[11] See http://suo.ieee.org/

[12] See, for example, http://www.iso.ch/iso/en/commcentre/pressreleases/1999/Ref762.html as an illustration of the growth of the ISO 9000 and ISO 14000 ''certification industry.''

exchange standards to a more powerful and agile systems-integration technology. At the lower left we see the current state of the art, with XML- and Express-based standards as examples. The next bubble shows a generation of standards with definitions of terms (or tags) that are formal enough (or pedantic enough) for a computer to be able to use that definition. This will allow what Tim Berners-Lee calls ''self-describing'' systems, shown in the third bubble. It implies a formal, semantic-definition language that is rigorous enough to support logical inference. Finally, in the fourth bubble, is the prize that will unleash the Web for computers—self-integrating systems. To accomplish that, you need several more ingredients.

Components of Self-Integrating Systems

In addition to a rigorous, semantic-representation language, the following elements are needed before self-integrating systems can become a reality.

- A negotiation protocol, analogous to diplomatic protocols between representatives of different nations (with different cultures), that will specify how two systems will greet one another.
- A semantic equivalence metric—a means of determining whether terms in one's own world (or ontology) have equivalents in one's partner's world, i.e., a quantitative measure of how alike two concepts are.
- A reasoning or inferencing capability within the communicating systems, to enable the systems to make judgements and draw conclusions about the meanings of terms.

Each of these elements is discussed below.

Semantic Representation Language. A discussion of the language properties needed to capture the semantics of a given context is beyond the scope of this paper. However, there is general agreement that a language such as the Knowledge Interchange Format (KIF)[13] that is based on first order logic would probably suffice for most circumstances. For a more thorough discussion of the merits of, and requirements for a formal logic foundation to support semantic communication, the reader is referred to Broekstra et al. [3] and Heflin [4].

Negotiation Protocol. Today's fax machines invisibly carry out a process of negotiation that makes it seem trivial to send messages between machines that are of different ages and capability. This negotiation process was defined and agreed to[14] by the fax manufacturers so their machines would remain interoperable in the presence of several, incompatible, facsimile-communication standards.[15] In its simplest form, a modern fax machine initiates a connection with an older vintage machine by sending a test signal using its most capable protocol. Upon failing to receive a response by the older machine that is not equipped with that protocol, the sending machine then tries a series of less capable protocols until a response is received. Finally, following some handshaking transmissions, the actual facsimile images are transmitted.

The fax example illustrates the concept behind the self-integration negotiation needed for the more complex transmission of semantic meaning between computers. In self-integration, at least one of the communication partners has to identify the context, or ontological model, in which it operates. This might be by means of identifying a previously registered ontology (should such a registry exist), or by supplying all the necessary semantic definitions directly. In a simple transaction, it might be sufficient to simply send the few definitions needed. The initiating partner also sends an indication of the nature of the desired communication, be it a query for information, or an instruction for execution, or an alert.[16] The receiving partner then evaluates the definitions offered and determines whether there is sufficient similarity with its own concepts. If not, then meaningful communication is by definition impossible; but if sufficiently equivalent meanings are discovered, then the receiver can proceed with a response to the communication.[17]

Semantic Equivalence Metric. In his roadmap[18] for the semantic web, Tim Berners-Lee implies that semantic equivalence mappings could come about by authors or third parties defining equivalence links between concepts in disparate semantic models. This is a very practical approximation for the near term, but a more quantitative metric will be needed eventually. In other words, two terms are never completely ''not the same'' or ''the same.'' They are functionally equivalent to some degree, depending on the context being considered. One could not hope to declare all such equivalence values *a priori*, since there is an infinite set of contexts one could use. Instead, one would have to evaluate the equivalence on each occasion, possibly looking for relevant similarity patterns in definitions and associations.

Just how to measure semantic equivalence is probably the greatest research challenge inherent in the vision of self-integrating systems. It is possible that fuzzy mathematics is applicable, or perhaps Baysian statistics, but the most suitable theoretical foundation still remains very much an open question. The reader is referred to Jones et al. [5] for more discussion of this issue.

Inferencing Support. The process of determining semantic equivalence will probably involve inferencing, since a given set of definitions offered by one communication partner will be unlikely to align with the other's definitions. It will be necessary to derive new constraints or assertions from those supplied to be able to compare terms. Again, as in the case of first order logic languages, the artificial intelligence community offers many viable inferencing engines and theories that have been well tested, any one of which could be pressed into service.

A Common Ontology? All of this communication would of course be much simpler if we all just agreed to a common set of definitions in the first place. There have been numerous attempts to build a single, unifying model of concepts and definitions in the hopes of tying together different automated systems.[19] It is this author's opinion that such a model will never be practical for three reasons: different communication partners operate in different contexts with different definitions, as mentioned earlier; any particular definition is subject to change over time, making the maintenance of a single large model a nightmare; and, finally, the world will never agree on what that model would be. We believe a more practical assumption is that any two systems that need to communicate will be based upon distinct ontologies.

Having said that, it is perfectly reasonable and practical for smaller communities, such as industry groups, to agree to some common semantic models to avoid having to negotiate meaning from the ground up each time a communication is needed. For example, the OAGIS specification developed by the Open Appli-

[13] KIF (see http://logic.stanford.edu/kif/dpans.html) for the ANSI draft standard.

[14] For example, the T.30 portion of the ITU-T Group 3 facsimile standard addresses the exchange of information about supported characteristics and management of the fax session, but does not talk about how the image is represented.

[15] Fax standards have evolved over the years, such as the CCITT Group 1, 2, 3 & 4 standards. The coexistence of these multiple standards (with increasing functionality) led to the need for a negotiation capability in modern fax machines. The same is true with modems.

[16] The need to characterize the nature of the interaction is illustrated in the domain of knowledge bases through the work supporting the Knowledge Query Manipulation Language (KQML). This language partitions components of conversations into types, such as query, response, alert, etc. KQML-type issues are outside the scope of this paper, however. See http://www.cs.umbc.edu/kqml/ for more information.

[17] See also D. McDermott, M. Burstein, and D. Smith, ''Overcoming Ontology Mismatches in Transactions with Self-Describing Service Agents'' First International Semantic Web Working Symposium (SWWS'01), 2001, (http://www.semanticweb.org/SWWS/program/full/paper39.pdf), for an excellent discussion of the use of service descriptions as a way to accomplish meaningful communication.

[18] See http://www.w3.org/DesignIssues/Semantic.html

[19] See, for example, http://www.cyc.com/cyc-2-1/intro-public.html

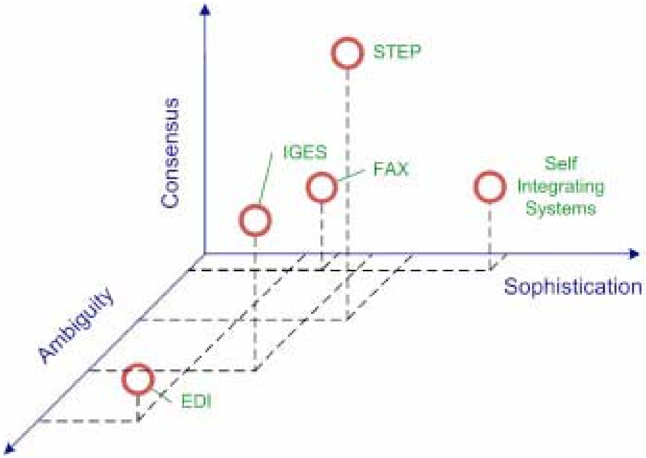

Fig. 2 Three-space view of the tradeoffs among standards solutions

cations Group[20] is gaining in popularity in several industrial sectors. It specifies ''Business Object Documents'' (or BODs) that capture the content of many business interactions, particularly as needed by enterprise resource planning systems. Alternatively, it may be practical for communities working in similar contexts to combine terms in different ways by means of reusable dictionaries. An effort at NIST has been underway since the late 1990's to define a practical, yet formal ontology supporting process information, for use in applications such as manufacturing planning and scheduling, called the Process Specification Language project.[21]

Discussion

So, the picture we have is one of a system of formal, definition standards for terms, based upon rigorous semantic theory that enables efficient, consistent testing and implementation of information exchange. Such a system will unleash the next Web revolution of computer-based integration and understanding of information.

Deployment Scenarios. Just how this picture will come into focus is of course uncertain. Here I describe three, possible, deployment scenarios for this technology. These scenarios may in fact represent three milestones in a roadmap for the evolution of information standards.

1. Industry sector consensus

This scenario represents the most pragmatic possibility, where there is an evolution of industry-sector consensus beyond current syntactic standards toward semantic standards. Industry sectors would still rely upon up-front agreement on definitions of terms, but these definitions are formalized beyond textual or even XML syntax in an attempt to reduce ambiguity and occasions for misinterpretation. Conformance- and interoperability-testing practices would improve based upon the increased rigor of definitions and consequent consistency of the standards themselves.

2 Agreement on some contextual dimensions

This scenario offers the promise of some degree of flexibility and ad hoc integration by characterizing terms by means of a classification code, similar to the group technology (GT) coding schemes used to classify product families in many manufacturing facilities. This would enable a practical determination of common context for communication without resorting to fundamental reasoning methods. Once a common context is established, a successful comparison of definitions becomes more likely, even in the presence of differing naming conventions. A trade-off of this approach is the possibility of false determinations of equivalence of terms, since a full inferencing analysis is not carried out.

3 Pure first contact, no presumptions

This more ambitious scenario makes almost no assumptions about a priori agreements between partners. It does require something like the Standard Upper Ontology[22] and a supporting language, to be agreed upon ahead of time, to enable any meaningful communication to take place. Under those assumptions, the self-integrating scenario described earlier can support a semantic discovery process for each and every communication. While this is considerably more laborious, it offers tremendous flexibility for automated, ad hoc integration of communication partners.

Tradeoffs between Consensus, Sophistication, and Ambiguity of Information. Figure 2 illustrates the inherent tradeoffs between the degrees of consensus, sophistication and ambiguity that industry must accept in any communication context. By ''consensus'' I mean the number of agreements required ahead of time (rather than the pervasiveness of the agreement), such as standard models for information, architectures, protocols, and so on. In this sense, a large amount of required consensus thus reduces one's ability to respond to changes in the marketplace. By ''sophistication'' I mean the complexity and technology needed to carry out a communication. ''Ambiguity'' is the inverse of the degree to which conformance to a given approach guarantees interoperability, or successful communication. Thus, one would like to move as close to the origin of Fig. 2 as possible, ideally through a lightweight, unambiguous, communications mechanism that requires only simple technology.

The EDI (Electronic Data Interchange) standard requires few agreements on definitions (consensus), and is relatively simple in terms of sophistication. Nevertheless, it suffers from great ambiguity as to the nature of information being conveyed. The IGES (Initial Graphics Exchange Specification[23]) contains moderate sophistication, but again suffers from too much ambiguity for many high-precision applications. The STEP standard (ISO 10303) was an attempt to reduce this ambiguity by adding more consensus (as defined above) in terms of layered standards and application protocols, plus increased sophistication in terms of the EXPRESS definition language. Finally, in our original, fax-machine example, newer model machines evolved by adding sophistication (protocol negotiation capabilities) in order to reduce the required level of consensus (pre-agreement on a single rigid protocol).

Scenario 1 implies consensus in terms of pre-agreed, semantic definitions, much like today's syntactic standards, but because of the increased rigor of the definitions (increased sophistication), the ambiguity is reduced. Scenario 3 implies significantly more sophistication in terms of communication mechanisms, semantic representation, and inferencing languages, to name a few, but a reduced need for consensus, thus offering more agility in the marketplace. Scenario 2 is a compromise between the two, requiring agreement on the classification scheme for term definitions (consensus), but without requiring general reasoning capabilities (sophistication).

Summary

As we move from highly structured, rigid agreements on how information is to be exchanged toward more flexible, sophisticated techniques, maintaining and testing integrated systems will become ever more difficult. However, in the face of the increasing pace of business and the increasing reliance on distributed manufacturing supply chains, we have little choice but to move forward

[20] See http://www.openapplications.org/

[21] See http://www.nist.gov/psl

[22] The Standard Upper Ontology effort (http://suo.ieee.org) is an attempt to gain universal acceptance of a small, core set of definitions in order to allow minimal discourse. There remain questions on the size of this minimal set of definitions, and on the definitions themselves.

[23] See https://www.uspro.org/new_catalog/approved_iges.html

with new technology solutions that can accommodate this shifting landscape. Interoperability between systems is becoming one of the principal barriers to achieving the time-to-market demanded by today's competitive environment, and it is only through the adoption of new approaches such as those described in this paper that the standards community will be able to respond.

Acknowledgments

I would like to acknowledge many colleagues who offered insights, ideas, comments and suggestions that made this paper much more complete, specifically Michael Gruninger, Al Jones, Sharon Kemmerer, Evan Wallace, Russell Kirsch and others I'm sure I have neglected to mention.

Disclaimer

No approval or endorsement of any commercial products by the National Institute of Standards and Technology is intended or implied. Certain commercial software systems are identified in this report in order to facilitate understanding. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the software systems identified are necessarily the best available for the purpose.

References

[1] Abernathy, Frederick H., Dunlop, John T., Hammond, Janice H., and Weil, David, 1999, *A Stitch in Time*, Oxford University Press.

[2] Berners-Lee, Tim, Fischetti, Mark (Contributor), and Dertouzos, Michael L.,2000, ''Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web,'' Harperbusiness.

[3] Broekstra, J., Klein, M., Decker, S., Fensel, D., and Horrocks, I., 2000, ''Adding Formal Semantics to the Web: Building on Top of RDF Schema,'' Proceedings of the ECDL 2000 Workshop on the Semantic Web, see http://www.ics.forth.gr/proj/isst/SemWeb/proceedings/session2-2/paper.pdf

[4] Heflin, J., 2001, ''Towards the Semantic Web: Knowledge Representation in a Dynamic, Distributed Environment,'' Ph.D. Thesis, University of Maryland.

[5] Jones, A., Ivezic, N., and Gruninger, M., 2001, ''Towards Self-Integrating Software Applications for Supply Chain Management,'' Information System Frontiers, **3**(4), pp. 403–412.

XML Representation of STEP Schemas and Data

Joshua Lubell
e-mail: lubell@nist.gov

Simon Frechette
e-mail: sfrechette@nist.gov

National Institute of Standards and Technology,
100 Bureau Drive, Stop 8260,
Gaithersburg, MD 20899-8260

The first Technical Note in this series [1] introduced the international standard ISO 10303, informally known as STEP (Standard for the Exchange of Product Model Data). Subsequent Technical Notes discussed various issues faced by users of STEP and how the ISO TC184/SC4 committee is addressing these issues. This article discusses the representation of STEP data using the Extensible Markup Language (XML). We begin by presenting XML's advantages as an exchange medium for STEP. We then discuss some of the issues involved in designing an XML exchange mechanism for STEP and conclude with an update of SC4's STEP/XML activities. [DOI: 10.1115/1.1476682]

Background

The objects to be represented and exchanged using STEP, as well as the associations between these objects, are defined in schemas written in EXPRESS [2,3], an information modeling language combining ideas from the entity-attribute-relationship family of modeling languages with object modeling concepts. Exchange is usually done by means of an ASCII file using a character-based syntax defined in Part 21 of STEP [4]. 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.

XML [5], a standard from the World Wide Web Consortium (W3C), provides a universal syntax for structured data on the Web. XML uses *tags* (words inside angle brackets) to delimit pieces of data called *elements*. An element's starting tag may also contain name-value pairs called *attributes* (not to be confused with EXPRESS attributes). Unlike Hypertext Markup Language (HTML), XML allows developers to design their own application-specific tag sets. Independently developed vocabularies can be combined using XML's namespace mechanism [6], enabling data to use multiple tag sets.

Unlike the Part 21 syntax, XML is easily extensible and is supported by inexpensive and widely used software tools. Also, it is easier to render XML data into forms that are suitable for human perusal. In order to capitalize on XML's popularity and flexibility, and to accelerate STEP's adoption and deployment, SC4 has undertaken a project whose goal is to standardize a method for representing EXPRESS schemas and data in XML. The expectation is that XML will not only enable developers to use low-cost, ubiquitous XML software tools to implement file-based exchange and visualization of STEP data, but can also potentially facilitate the use of STEP models in emerging areas such as XML-based web services.

As a simple example of how STEP data might be encoded in XML, consider the following EXPRESS definition of a point on a plane with x and y-axes. This definition declares point to be an *entity type* and x and y to be real-valued *attributes* of that entity type.

```
ENTITY point;
 x, y : REAL;
END_ENTITY;
```

The point (3.1, 5.7) represented using Part 21 syntax would look like this:

```
#1=POINT(3.1,5.7);
```

Now consider an XML representation of the same point. The point entity type and the x and y attributes are represented as elements. An XML attribute named id uniquely identifies the point.

```
⟨point id=“p1”⟩
 ⟨x⟩3.1⟨/x⟩
 ⟨y⟩5.7⟨/y⟩
⟨/point⟩
```

This XML representation is easier to read and also conveys more information than the Part 21 representation. In particular, the XML representation explicitly labels the point's x and y coordinates. The Part 21 representation, however, omits these labels. Thus, any application reading the Part 21 data has to also read and parse the EXPRESS definition of a point (or have the structure of a point hard-wired into its code) in order to perform any processing.

The preceding XML representation is an *early binding* in that the tag names correspond directly to their counterparts in the EXPRESS. In a *late binding*, the named components of the XML vocabulary do not directly correspond to EXPRESS names. Instead of defining EXPRESS names as tags, a late binding specifies them in the data either as XML attribute values or as element

Contributed by the Engineering Informatics (EIX) Committee for publication in the JOURNAL OF COMPUTING AND INFORMATION SCIENCE IN ENGINEERING. Manuscript received Jan. 2002; Revised Mar. 2002. Associate Editor: R. Rangan.

content. For example, a late-bound XML representation of a point specifying EXPRESS names as XML attribute values might look like this:

```
⟨entity id=“el” name=“point”⟩
 ⟨attribute name=“x”⟩
  ⟨real_value⟩3.1⟨/real_value⟩
 ⟨/attribute⟩
 ⟨attribute name=“y”⟩
  ⟨real_value⟩5.7⟨/real_value⟩
 ⟨/attribute⟩
⟨/entity⟩
```

Although late bindings are more verbose than early bindings, a late-bound EXPRESS-to-XML mapping is better suited for XML applications involving multiple EXPRESS schemas. A late binding allows for a single tag set to be used for all EXPRESS models, since the XML vocabulary defined by the tag set corresponds to EXPRESS metadata objects rather than to objects defined in the model. If an early-bound strategy is used for such applications, there must be a distinct XML language for each EXPRESS schema. Early bindings are therefore most useful for XML applications implementing a single EXPRESS schema. Early bindings are less verbose, more human-readable, and simpler to process than late bindings, and they are also better equipped to make use of XML software tools. As a result, STEP implementers tend to prefer early bindings, and we focus exclusively on early bindings for the remainder of our discussion.

Issues

The following are some issues that arise when attempting to reformulate EXPRESS models as XML.

XML Schema Languages. To specify an XML binding of data modeled in EXPRESS, one must provide an algorithm for mapping EXPRESS constructs to their XML counterparts. The algorithm's input is an EXPRESS schema. Its output is a set of syntax rules for the XML binding's vocabulary. These rules may be specified using any one of the several schema languages available for XML.

The simplest and oldest XML schema language is the Document Type Definition (DTD), described in the XML 1.0 specification (see [5]). Using DTD syntax, we can say that the point element has an XML attribute id whose value is a unique identifier and whose content consists of element x followed by element y. Elements x and y each contain character data. The DTD markup declarations are as follows:

```
⟨!ELEMENT point (x, y)⟩
⟨!ATTLIST point
          id ID #REQUIRED⟩
⟨!ELEMENT x (#PCDATA)⟩
⟨!ELEMENT y (#PCDATA)⟩
```

These markup declarations show some of the shortcomings of DTD syntax. Because DTD elements cannot be assigned data types other than #PCDATA (parsed character data), a DTD cannot enforce the constraint that the point's x and y coordinates be real numbers. Also, DTDs do not permit multiple declarations for the same element. For example, if our EXPRESS schema declared, in addition to point, an entity type called x, we would be unable to add a second element declaration for x to our DTD.

A more sensible approach is to specify the binding using an XML schema language without these limitations of DTD syntax, such as the W3C's XML Schema definition language [7]. W3C XML Schemas can define context-sensitive element types and provide a much more extensive collection of built-in data types than DTDs [8]. An additional advantage of W3C XML Schema is that, because the definition language is itself an XML tag set, standard XML programming interfaces such as the Document Object Model (DOM) [9] can be used to automate schema conversion from EXPRESS to XML.

The following W3C XML Schema definition specifies the XML early binding syntax for a point. Elements x and y, when contained within point, are defined to have real numbers as their content. However, it is possible for x and y to be assigned different types in other contexts.

```
⟨element name=“point”⟩
 ⟨complexType⟩
  ⟨all⟩
   ⟨element name=“x” type=“decimal”/⟩
   ⟨element name=“y” type=“decimal”/⟩
  ⟨/all⟩
  ⟨attribute name=“id” type=“ID” use
  =“required”/⟩
 ⟨/complexType⟩
⟨/element⟩
```

It is worth noting that even W3C XML Schema, with its data typing abilities and support for context-sensitive elements, lacks the ability to describe all of the constraints that EXPRESS is capable of representing. For example, XML schemas cannot represent global constraints on a population, such as the requirement that all points be collinear. Although it is possible to develop an XML language that represents all EXPRESS constraints, an application using that language would have to supply its own logic for interpreting those constraints. Generic XML software would be of no help. Thus, for all practical purposes, any EXPRESS-to-XML mapping will result in a loss of information. However, the XML representation is able to retain at least *some* of the information present in the EXPRESS schema. For some use cases, the XML schema and data may be all that is needed. Other use cases may require that additional (unmapped) information from the EXPRESS schema be hard-coded into the application responsible for processing the XML-encoded STEP data.

Hierarchies. Although XML is hierarchical by nature, neither EXPRESS schemas nor Part 21 files have any intrinsic hierarchical structure. Nothing in an EXPRESS schema identifies an entity instance as part of, or belonging to, a higher-level entity. There are no clues in the EXPRESS schema to suggest which entity types can be mapped to hierarchical constructs in XML. An EXPRESS schema is a network of largely independent entity types connected by relationships somehow modeled using attributes whose values are entity instances.

Accordingly, an EXPRESS data set can be mapped into a sequence of XML elements representing EXPRESS entity instances. Each entity element has an XML ID attribute, providing a unique identifier for the entity instance (as in the point example in the Background section). Assume that EXPRESS attributes are mapped to XML elements rather than XML attributes. Then each element representing an EXPRESS entity type contains elements representing the EXPRESS entity's attributes. In addition, every entity type also maps to a *reference element*—an XML element type with an XML attribute of type IDREF and empty content. An occurrence of this element represents a reference to the entity instance with the given ID value. In short, the XML data consists of a collection of ‘‘flat’’ elements representing independent entity instances linked together by IDREFs, because no hierarchical structure can be deduced from the EXPRESS model.

For example, consider a line EXPRESS entity type modeled as follows:

```
ENTITY line;
 point_on_line : point;
 slope : REAL;
END_ENTITY;
```

A line represented by the point entity instance p1 and with a slope of 0.5 might look like this in XML (reference element in boldface):

```
⟨line id=“line1”⟩
 ⟨point_on_line⟩
```

```
   ⟨point-ref ref=“p1”/⟩
 ⟨/point_on_line⟩
 ⟨slope⟩0.5⟨/slope⟩
⟨/line⟩
```

To make better use of XML's inherent hierarchical structure, one might wish to use a representation method allowing entity instances to be contained within other instances. Such an XML representation method requires some formal annotation of EXPRESS schemas to cue the mapping to useful hierarchical data structures in XML. Some of these annotations might belong to the EXPRESS schema itself, while others might be specific to particular STEP implementations using the schema [10].

An XML representation of a line instance containing a point entity instance could look something like this:

```
⟨line id=“line2”⟩
 ⟨point_on_line⟩
  ⟨point⟩
   ⟨x⟩3.1⟨/x⟩
   ⟨y⟩5.7⟨/y⟩
  ⟨/point⟩
 ⟨/point_on_line⟩
 ⟨slope⟩0.5⟨/slope⟩
⟨/line⟩
```

Other Issues. The preceding issues represent just a few of many challenges involved in mapping EXPRESS schemas to XML schemas. Two additional issues that are particularly thorny are the representation of EXPRESS attributes whose values are aggregates (lists, sets, arrays, or bags) and the handling in XML of multiple inheritance and other complex inheritance relationships in EXPRESS. In order to keep this Technical Note brief, we do not discuss the latter issue. The former issue, discussed at length in [11], makes it a problem to represent EXPRESS attributes as XML attributes. Thus, we map EXPRESS attributes to XML elements in our examples in this discussion.

Current Status

SC4 recently issued Part 28 of ISO 10303, entitled ‘‘XML representation of EXPRESS schemas and data,’’ as an ISO Technical Specification [12]. Because W3C XML Schema did not become a Recommendation in time for it to be normatively referenced by Part 28, Part 28 uses DTD syntax to specify mappings of EXPRESS to XML. This results in a sub-optimal solution for the reasons we discussed in the XML Schema Languages sub-section. Furthermore, Part 28 specifies three different XML mapping algorithms: a late binding method and two early binding methods. As a result, Part 28 is a lengthy and complex document of more than 300 pages.

Recognizing the limitations of the current version of Part 28, SC4 has begun to develop a second edition of the standard. This second edition will, in all likelihood, use W3C XML Schema to represent EXPRESS schemas. Because W3C XML Schemas are better suited for representing EXPRESS-modeled data than DTDs, the Part 28 team expects to have an easier time developing a single XML binding that satisfies most of the requirements of XML application developers wishing to use STEP. With a little luck and a lot of effort, Part 28 version 2 will be a concise, high-quality standard that will succeed in lowering the barriers preventing widespread industry adoption of STEP.

References

[1] Pratt, M. J., 2001, ‘‘Introduction to ISO 10303—the STEP Standard for Product Data Exchange,’’ ASME J. Comput. Inf. Sci. Eng., **1**(1), pp. 102–103.

[2] ISO 10303-11: 1994, Industrial Automation Systems and Integration—Product Data Representation and Exchange—Part 11: Description Methods: The EXPRESS language reference manual.

[3] Schenk, D. A., and Wilson, P. R., 1994, Information Modeling: The EXPRESS Way, Oxford University Press, New York, NY.

[4] ISO/FDIS 10303-21:2001(E), Industrial Automation Systems and Integration—Product Data Representation and Exchange—Part 21: Implementation Methods: Clear Text Encoding of the Exchange Structure.

[5] World Wide Web Consortium. Extensible Markup Language (XML) 1.0 (Second Edition). W3C Recommendation. 6 October 2000. Available on-line at http://www.w3.org/TR/REC-xml.

[6] World Wide Web Consortium. Namespaces in XML. W3C Recommendation. 14 January 1999. Available on-line at http://www.w3.org/TR/REC-xml-names.

[7] World Wide Web Consortium. XML Schema: Part 1: Structures. W3C Recommendation. 2 May 2001. Available on-line at http://www.w3.org/TR/xmlschema-1.

[8] World Wide Web Consortium. XML Schema: Part 2: Datatypes. W3C Recommendation. 2 May 2001. Available on-line at http://www.w3.org/TR/xmlschema-2.

[9] World Wide Web Consortium. Document Object Model (DOM) Level 2 Core Specification Version 1.0. W3C Recommendation. 13 November 2000. Available on-line at http://www.w3.org/TR/DOM-Level-2-Core/.

[10] ISO TC184/SC4/WG11. XML representation for data sharing (CEB Binding-Draft 3.0). Revision N1362000-09-29. Available on-line at http://www.nist.gov/sc4/wg_qc/wg11/n136/.

[11] Barkmeyer, E. A., and Lubell, J., 2000, ‘‘XML Representation of EXPRESS Models and Data,’’ Proceedings of XML Technologies and Software Engineering workshop, Toronto. Available on-line at http://www.mel.nist.gov/msidlibrary/doc/xse2001.pdf.

[12] ISO/TS 10303-28: 2002(E), Industrial Automation Systems and Integration—Product Data Representation and Exchange—Part 28: Implementation Methods: XML representation of EXPRESS schemas and data.