Appendix B

THE EXTENSIBLE MARKUP LANGUAGE

Commerce is buying and selling. E-commerce is commerce with human interaction replaced by digital information insofar as practical. To work, it needs a commonly understood modality of exchange; a reliable description of the product to be purchased; and, in some cases (see Appendix E), a succinct statement of the buyer's expectations. People do all this by talking to each other, using brainpower, social cues, and a shared cultural context to figure out what each is trying to say. Machines, to repeat a familiar refrain, have only symbolic notation to go by. Thus, if E-commerce is to get past its current incarnation as mail-order but with keyboards, rather than phones, such notations must be explicit, mutually understood, and well-formatted—hence, standardized.

In a field that insists on shedding its skin as often as the Web does, it may be premature to say that the search for such a standard has ended. But today's bettors seem increasingly inclined to place their money on a metalanguage, XML. The use of *metalanguage* is deliberate. It has been remarked that XML solves how we are going to talk to each other, but we still need to agree on what we are going to talk about. XML is the grammar, not the words—necessary, but by no means sufficient. And therein lies both the hope and the hype of what may be the keystone of tomorrow's E-commerce.

In analyzing XML, this case study attempts to do several things: explain the broader advantages of markup, trace the history of XML through its origins in earlier standards, limn its current status, and portray the hurdles it must overcome to fulfill its promise.

XML AS A MARKUP LANGUAGE

The many limitations of HTML have prompted the industry to conclude that it is time to move beyond it. While HTML is a useful way to present information, it does little to organize it.[1] As one result, every time a document format is changed, the markup has to be redone, and a developer cannot alter a document's presentation without creating a different version of the document with new markup. XML tags, such as <price>5.95</price>, by contrast, indicate variables and their values when attached to any data element in a document. This permits XML documents to be processed by computers as well as read by humans. Tags can be used to drive searches and comparisons of data elements within a company's site, or across data sets provided by many companies. They also allow data to be arranged in specific ways for specific users. It provides users only the data elements that are of interest to them. In contrast, HTML tags contain only format information, forcing search engines to do textual analysis of Web pages and leading to many useless "hits" that do not fit the context of the search.

The strength of XML is that the standard opens itself up to an infinity of tags representing the infinity of objects and qualities one might want to keep tabs on. Unlike HTML, which has a single standard set of tags, however, XML tags can be defined document by document, application by application, industry by industry, or globally. (Data Interchange Standards Association, 1998.) This extensibility is one of the most attractive features to many users.

But it is also a potential weakness. For tags to permit cross-company or cross-industry comparisons, they must represent common concepts in commonly denoted ways. When people talk about the vast new global E-commerce markets facilitated by XML, they are

---

[1] HTML has been able to accommodate random parenthetical material and presentation hints since version 2.0. The META construct permits variables and values to be inserted as markups in documents. Some META constructs send information about an HTTP header field to an HTTP server; e.g., <META HTTP-EQUIV="Expires" CONTENT="Tue, 04 Dec 1993 21:29:02 GMT">. Other constructs are user-supplied; e.g., <META NAME = "television character" VALUE = "Barney">. In the META tag one can glimpse a early version of open markup but one that did not become the basis of a stronger descriptive vocabulary. The META construct lacked a way of defining a tag in a document (either directly or by reference), any structure, or any way to mark up text using such tags.

implicitly assuming that companies will create their catalogs and other information using common tags. Ideally, this would happen, and companies would compete on the qualities of their product and service offerings. However, the world is far from ideal. "Browser wars" between Netscape and Microsoft were fought in part through the use of proprietary HTML tags, as the two companies tried to expand the capabilities of their products to create better-looking documents and attract developers and users. Still, despite the possibility that some companies will not use standard tags, there are many efforts under way to agree on common tags within and across domains.

XML documents specify their tags and the relationship among them by leading off with or at least referring to a DTD,[2] which specifies what elements may exist where, what attributes elements may have, what elements must be found inside other elements, how elements may combine, and in what order. DTDs allow a validating XML parser (i.e., a computer program that reads XML), to determine whether the document's tags are "legal" and properly arranged for a given type of document. Those that fail generate error messages. But with every new DTD, a new set of tags becomes possible—hence the "extensible" in XML, a capability that HTML lacks; if a tag is not in the HTML standard, an author cannot define it into existence.

Forcing a tag to be a standard had its uses in HTML. If the browser recognizes the tag, it knows how to present the tagged information (e.g., whether to highlight, italicize, or offset the text). But Web designers found that they needed to use tricks to overcome the limitations imposed by the limited number of HTML tags. Some proprietary tags have been invented, reducing cross-platform useability (e.g., "This document best viewed with Netscape Navigator."). HTML has been used in ways that were never intended: single-point GIFs and too many tables. After a while, documents become hard to manage.

---

[2]Although most XML documents published contain DTDs, documents without DTDs may still be valid (by contrast, SGML requires a DTD for every document). An author can, in effect, create a DTD by implication—arranging tags in a way that an XML parser finds acceptable. Without a DTD, though, there is no automatic way to check whether all the tags that should be present are present and tags that should be absent are absent.

XML tags for their part have few or no inherent clues for presentation. This forces the use of style sheets to determine how a document would be presented. The separation between content markup and style definition allows the same document to be processed or published without additional work. The ability to attach many style sheets to the same XML document allows much finer control of the way a document looks in various presentations without affecting content in any way. A designer may, for instance, use one style sheet for display on regular computer screens; a different one for small screens, such as Palm Pilots; a third one for display on browsers with graphics turned off; a fourth one for printing; etc. Each style can be defined to the satisfaction of the designer without requiring that a document's markup be redone. By contrast, such presentation control can only be achieved with HTML by creating the same data with different markups for each presentation, storing all these documents in a database, querying the requesting device as to its type, and returning from the database the version that matches the specific type of browser. Dynamic HTML uses scripting to display and redisplay pages, based on user actions. XML's ability to let designers use standard style sheets or to create new style sheets from standard components offers a tremendous economy of effort in separating content from presentation. In addition to economizing effort, style sheets are more varied and much more flexible than HTML tags.

Style sheet standardization is dealt with via the Extensible Stylesheet Language (XSL), a descendant of the Document Style Semantics and Specification Language (DSSSL—ISO 10199) with roughly the same relationship as that of XML to SGML.[3] The W3C published the first draft of the XSL specification on August 18, 1998. (W3C, 1998b.) Later versions have already been published, with the latest version published in conjunction with other specifications touching on XSL and XML. (W3C, 1998c.)

In addition to tag structures, XML also provides facilities for link structures (Cover, 2000) via the XML Linking Language (XLL), with its two major components: XLink and XPointer. XLink

[3]The W3C's proposed recommendation for "Associating Stylesheets with XML Documents" was released in late April 1999. To kick-start the creation of XSL style sheets, Sun Microsystems and Adobe sponsored a contest with prizes valued at $90,000 for those who could develop layout engines for Mozilla, Netscape's open-source browser software. See also Johnson (1999).

> specifies constructs that may be inserted into XML resources to describe links between objects. A link, as the term is used here, is an explicit relationship between two or more data objects or portions of data objects. XLink uses XML syntax to create structures that go beyond the simple unidirectional hardwired hyperlinks of today's HTML to include sophisticated multi-ended and typed links[4]

They include:

- multidirectional links (so that users can return to the original location via a corresponding link at the first link's destination)
- multiple-destination links (giving users a choice)
- links to fragments[5]
- link databases to store links (thereby making it easier to adjust to changing link addresses).

The XPointer language defines "constructs that support addressing into the internal structures of XML documents. In particular, it provides for specific reference to elements, character strings, and other parts of XML documents."[6]

GETTING TO XML

What was the point of trimming SGML to get to XML? SGML is a heavyweight language meant to tackle "large, long-term document publishing" (see Jellifee, 1998), such as DoD's entire corpus of technical documents. Yet, its very size and complexity made it "just too hairy for real people to get into; you could crack great big problems, but sometimes not do the simple things simply. Then the Web came

---

[4]See the W3C working draft on the XML Linking Language (XLink)(W3C, 1998b). Note also the comparison with the complex links of HyTime.

[5]With XLL designers can (1) place content directly into the document being viewed without user intervention (so that a document on, for instance, chemical compounds could be viewed and a section on fructose automatically inserted from an entirely different Web site), and (2) replace content in line with updated content from another document. Yet, if the original text has original markup that conflicts with the markup of the document being viewed, strange-looking documents may result. Also, direct insertion prevents the quoting author from adding his own markup for emphasis.

[6]See the W3C working draft on the XML Pointer Language (XPointer)(W3C, 1998a).

along and showed the power of doing simple things simply."[7] XML is designed for doing "efficient, small, short-term documents." (See Graphic Communications Association, 1999.)

Die-hard SGML advocates could have justifiably argued that the SGML's bells and whistles were no real barrier to designers—who could have simply ignored those features they did not feel like exploiting. But those who wrote programs (such as browsers) to read marked up text would have had to accommodate any feature that the text's authors felt like putting in. Within a delimited universe (e.g., defense contractors, automakers) this problem could be avoided by developing master DTDs that avoided the more obscure features of SGML. But once the challenge became interpreting random text produced by someone outside the institutional aegis, ignoring obscure features could easily have led to disappointment or disaster were such features to be used. This is an example of how the role and thus the content of standards designed to unify a heterogeneous corporate infrastructure under a single authority (e.g., CORBA) failed to fit the model of a Web that may encompass literally anyone.

Although the itch to lighten SGML was long-standing, only in mid-1996 did Jon Bosak of Sun Microsystems convince the W3C to create a working group for SGML on the Web. The SGML Editorial Review Board included chief information officers, Internet IPO architects, and standards editors. The original idea was to "put in everything that's proven to work . . . and throw the rest out." Within a year it had become the XML Working Group. Although the SGML community leapt on board instantly, the "Webheads" held off.[8] As Jean Paoli of Microsoft observed, HTML was a more-or-less standard, widely used tool that worked. By contrast, XML's early fans were those least happy with HTML's limited power. (Seybold Publications and O'Reilly Associates, 1997.) Once Microsoft decided to use XML in its Channel Definition Format (its "push" technology) and announced the decision in March 1997, XML began to generate significant interest among programmers and Internet professionals. (Seybold Publications and O'Reilly Associates, 1997.) XML has been in ascendance ever since.

---

[7] Tim Bray, interviewed in Veen (1997), p. 1.

[8] Tim Bray, interviewed in Veen (1997), p. 2..

XML, by restricting choices present in SGML, grew simpler (see Johnson, 1999):

- A specific choice of syntax characters was made so that everyone using XML will use the same concrete syntax. For example all tags must begin with "<" and end with ">". Attribute values must be enclosed in quotes.
- A new empty-element tag was invented to indicate that an end tag is not expected. It looks like this: <some text/>.
- Tag omission is forbidden; each nonempty element must have both a start tag and an end tag. All tags must be properly nested (e.g., this is wrong).
- DTDs may be omitted.

Nevertheless, XML is not *that* much lighter. After all, every legal XML document is also, by definition, a legal SGML document. SGML has had a hard slog in the marketplace, accepted only in some communities. So why the optimism for XML? HTML helped; it taught both professionals and amateurs the value of working with markup. With HTML accepted, XML is seen as a way to overcome the limitations facing HTML. Users have moved past presenting pages and are looking for capabilities to search, collate, and move information and to allow computer systems to communicate without human intervention. Proclamations and product announcements by mainstream Web firms, such as Sun, IBM, Lotus, Oracle, Adobe, and Microsoft, have raised the odds that XML could become central to the Web's future. (Alshuler, 1999.)

If the purpose of XML was "to enable generic SGML to be served, received, and processed on the Web in the way that is now possible with HTML," the recasting of HTML into XML format has to be key. On May 5, 1999, the W3C HTML Working Group released a revised version of "XHTML 1.0: The Extensible HyperText Markup Language. A Reformulation of HTML 4.0 in XML 1.0" (W3C, 1999a) which provided a new set of modularized XML DTDs for HTML. By breaking up XHTML into a series of smaller element sets, it permitted the combining of elements to suit the needs of different communities. How easy or smooth will the transition from HTML to XHTML be? XML has much stricter rules than HTML, and XHTML is expected to comply with the rules of the XML specification. The key

is whether programmers who are used to playing fast and loose with current HTML are willing to trade that for the greater expressive power of XML.

In general, SGML's advocates have been helpful to XML. Until XML came along, the largest single source of support for SGML was DoD's logistics community, whose CALS (né Computer-Assisted Logistics Support) program imposed requirements on defense contractors to document their technical support material in a standard way. Text was to be rendered in SGML, images in a series of ever-more sophisticated standards culminating in STEP. Groups developing STEP (largely in the aerospace and automotive sectors) realized that they can use SGML, and now XML, to integrate product documentation fully into product data management, to view structured information repositories of complex documentation and legacy data warehouses via Web browsers, and to manage technical and administrative flows of information within supply chains and consortia. (Wrightson, 1999.) As a result, efforts for full harmonization between STEP and SGML/XML are under way. The U.S. government's CALS standard has officially shifted SGML to XML. Meanwhile, NIST is transferring resources from three-dimensional representation (Virtual Reality Modeling Language [VRML]) into XML. The Text Encoding Initiative, a project funded since 1988 by the National Endowment for the Humanities to tag all of the world's literature, was another heavy user of SGML. In the last five years, the initiative has developed a compact tag set to foster more use. (Burnard and Sperberg-McQueen, 1995.) C. M. Sperberg-McQueen, a primary force behind the initiative, has become a pillar of the XML community.

XML AND E-COMMERCE

XML was built for applications that

- require the Web client to mediate between two or more heterogeneous databases
- attempt to distribute a significant proportion of the processing load from Web server to Web client
- require the Web client to present different views of the same data to different users

- use intelligent Web agents to tailor information discovery to the needs of individual users.

Each is relevant to E-commerce. As buyers compare products and prices from virtual catalogs (i.e., databases) maintained by a variety of sellers, process this information to determine the best match (usually in their own machines rather than distant servers), negotiate transactions, and track delivery and payments, they are using all the application types described above.

But to understand the potential effect of XML on E-commerce, it helps to look at consumer-to-business and business-to-business transactions separately.

These days, most consumer-to-business commerce requires the full-time attention of the consumer and the electronic attention of the business. Such trade is often little more than an advanced version of catalog shopping—only with a much-larger catalog and some ability to engage in long-dormant pricing behavior (e.g., auctioning off standard manufactured items). XML may permit software to scour the Web looking for purchasing opportunities that are specifically coded as offerings. XML pages with standardized tags, such as <Price> or <ModelNumber> could allow the search for and perhaps even the negotiation of best matches between buyers and sellers, presenting the buyer with a set of options for final selection and approval. Clothes, for instance, might be described in terms of a data set so complete (e.g., fabric, piece sizes, color) that a customer could simulate its appearance on a range of body types. Travel arrangements could be automatically calculated by mixing and matching the arrival and departure times of various segments. Much of the processing would move from the seller's server to the buyer's client machine, while the server would contain product information in a format most convenient for the seller. A third party could rate and otherwise compare varying offerings by their parameters (e.g., what colleges offer) and their performance (e.g., medical outcomes). Indeed, all that is required to justify XML is the need to describe in standard terms something that may inform or lead to a purchase.

The case for XML in business-to-business transactions is a good deal more straightforward, inasmuch as they are already becoming (1) completely automated processes and (2) are backed by standards (ANSI X12 in the United States and the United Nations' EDI for

Administration, Commerce, and Transport [EDIFACT] internationally, i.e., in Europe). X12 and EDIFACT specify digital formats used to encode key business documents, such as invoices, bills of lading, and payment transfers.

Business-to-business E-commerce is more complicated than business-to-consumer transactions. It is iterative and requires the participation of different people within sellers' and buyers' organizations, with each person contributing part of the transaction. There are two different types of business-to-business transactions: repeat purchases within a long-term relationship and one-time purchases. In the former, buyer and seller negotiate product attributes, prices, and terms of purchase, after which authorized buyer representatives (and in some cases sellers themselves) can trigger purchases of individual items. In the latter, the buyer usually specifies and sends requirements to several potential suppliers. After several of these submit bids by a specified date, the purchaser decides which bid to accept. One or more rounds of negotiations with one or more potential suppliers precede selection. In both transactions, once a supplier is chosen, goods or services are ordered, and sometimes partial payments are made before or during production. There are specific documents that must be exchanged between buyer and seller before goods and services are accepted and final payment is made.

EDI, in its current incarnation, has been pushed by large organizations, which want to decrease their purchasing costs and have the clout to make the smaller trading partners use EDI. But such EDI has severe limitations. First, its use of specific message formats imposes a strict structure on the transaction. Second, complex person-to-person arrangements must often occur before two business units can reliably use EDI. Third, it is expensive because it usually involves proprietary software and proprietary Value Added Networks (VANs) to translate messages among various EDI software packages and provide electronic mailbox hosting services for trading partners. Although Web-based X12 applications are being developed, these applications do not remove EDI's most important limitations.

XML would do away with today's EDI's limitation on the content of communications between buyers and sellers, as well as with the expense of VANs—and thereby boost E-commerce. It could allow any two buyers and sellers anywhere to communicate directly, using

their own formats for documents and a common set of content tags, all supported by commercial software and without the need for intermediaries. To preserve existing investments in X12 data, its message formats and tags could be included in XML-based EDI applications. But XML would also allow buyers and sellers to do things now impossible with today's EDI, e.g., to include human interaction within the E-commerce transaction stream, as different people are presented with Web-based forms for inputs and approvals within their organizational units or functions.

WHAT THE WORDS MEAN

But first XML must cope with the well-understood fact that the specification cannot alone ensure interoperability. XML's "body of knowledge" must include detailed syntax and vocabularies for communities of users—and the definition of communities must partition the universe of users cleanly enough so that there is little ambiguity among users over which language to use in conducting which business.

Thus, standard DTDs and vocabularies must be available to users via some sort of repository. High-level and general repositories could be managed by standards organizations; industry-specific repositories could be managed by industry groups, and more specialized repositories could be maintained by groups of partners or within individual companies. Several standards, addressed to the needs of individual communities, have already been published through the W3C, including the Mathematical Markup Language, the Chemical Markup Language, and the Astronomical Markup Language.

But many more groups are developing DTDs, suggesting that XML may be a victim of its own early popularity. The old saw that the wonderful thing about standards is how much choice one has in them is, at this juncture, less than completely amusing. Take the following examples:

- The Open Trading Protocol is a consortium of banking, payment, and technology companies specifying information requirements for payment, receipts, delivery, and customer support.
- The Open Buying on the Internet initiative, launched by American Express, Ford Motor, Office Depot, and others is

automating large-scale corporate procurement of office and maintenance supplies.

- RosettaNet is a PC industry initiative, managed by a board of 34 chief executive officers and chief information officers of major information technology users and vendors, which defines how to exchange PC product catalogs and transactions among manufacturers, distributors, and resellers. RosettaNet participated in a pilot project with CommerceNet (a consortium of several hundred information technology companies) on catalog interoperability because the project included laptop computers.
- Under the rubric of the Information and Content Exchange, CNET (part of the News Corp), Vignette, and other information content providers are developing ways to create and manage networked relationships, such as syndicated publishing networks, Web superstores, and on-line reseller channels.
- The Open Financial Exchange, proposed by CheckFree, Intuit, and Microsoft, supports banking, bill payment, investment, and financial planning activities by consumers.
- A consortium of 40 companies, spearheaded by software vendor Ariba Technologies, has developed Commerce XML (cXML) to standardize catalog content and purchasing data exchange.
- Microsoft has its BizTalk initiative.
- In June 1999, J. P. Morgan and PricewaterhouseCoopers LLP announced the Financial Products ML, designed to address the needs of the financial derivatives community.

XML may be standardized for commerce if combined with X12. There, too, several groups compete with the others in that they take different approaches, yet all claim to cooperate with each other. CommerceNet's framework for open Internet commerce, eCo System, was originally (1996) based on CORBA and later (1997) recast on an XML foundation (thanks in large part to the support of the big software companies). This framework promulgates a set of Business Interface Definitions (BIDs), which, when posted on the Web, tells potential trading partners what on-line services a company offers and what documents to use when invoking them. Its Common Business Library, an extensible public collection of generic BIDs and document templates, includes XML message templates for the basic

business forms used in X12 transactions. The Defense Information Systems Agency is funding more work into interfaces between XML and X12.

The U.S. XML/EDI working group was established in July 1997 (see XML/EDI Group, no date) with W3C's infrastructure support but with no explicit endorsement (this requires a formal working group recommendation to be submitted to a vote of the membership and then approved by the director). An international XML/EDI Group, housed by the Graphic Communications Association Research Institute (Alexandria, Virginia), is looking to create "a new powerful paradigm, different from XML or EDI" by "first implementing EDI dictionaries and extending our vocabulary via on-line repositories to include our business language, rules and objects." (Graphic Communications Association, 1999.)

Europeans have their own XML/EDI Pilot Project, under the European Center for Standardization/Information Society Standardization System (CEN/ISSS). They seek to "explore how XML can be used to provide an interface between existing EDI applications and the next generation of XML-aware applications" and study how XSL could help present EDI messages to people in ways that account for variations in their linguistic and cultural background."[9] It also comments on how the W3C's work on XML and EDIFACT can be used with "the multilingual and mixed trading practices found in Europe." (CEN/ISSS, 1998b.) Europe's work builds on other XML-EDI work, such as EuroStat and the Norwegian government projects on the interchange of statistical data, CEN TC2251 for health care informatics, TIEKE in Finland on transport-related messaging, EDIFRANCE on E forms, and UK/CEDIS on Simple EDI. The project's success factors include the quality of the XML DTDs it created, the acceptability of the software tools to end users, and the acceptability of XML as an alternative to today's EDI. (CEN/ISSS, 1998a.) The project published its preliminary findings in October 1998. Europeans worry that American efforts fail to refer to the relationship between X12 and EDIFACT—a poor way to promote globalization of commerce, which is a stated goal of many XML-related E-commerce efforts. (CEN/ISSS, 1998e.) The European Electronic Messaging

[9] Preceding quotes from CEN/ISSS (1998d).

Association EDI Working Group has proposed that the UN create and manage a repository of XML tags based on EDIFACT. (Raman, 1998.)

MANAGING PROLIFERATION

One approach to the problem of standards proliferation is the creation of ontologies (a concept from the study of the nature of knowledge), each of which codifies the concepts meaningful to a community. Thus, everyone would have a common understanding on which to build vocabularies. Ontology.Org and CommerceNet (Glushko et al., 1999) are working to create a set of business-related ontologies, such as various aspects of payments and business processes.

Another reaction has been the formation of consortia to develop and maintain a registry of vocabularies. OASIS is composed of vendors and consumers assembled to work on interoperability shortfalls between products or among software suites. The focus is on horizontal application products, such as XML table models or conformance suites. They are moving into registries, in what may be some competition with Microsoft's Biztalk initiative. As of mid-1999, the two efforts had become at least somewhat harmonized.[10] OASIS is tied into CommerceNet in that its Registry and Repository Technical Committee is (as of mid-1999) chaired by one of its employees.

XML AS A STANDARDS ABSORBENT

One sign of the hopes being invested in XML has been its ability to encompass other standards (e.g., SGML). Supporters of many other standards have hopped on the XML bandwagon by converting their vocabulary into tag sets, quietly chucking earlier vehicles. Many such standards, however, had yet to achieve much lift.

The W3C's Platform for Internet Content Selection (PICS), for instance, predates XML. It is a structured set of Web references and metadata tags through which Web sites could attach ratings (e.g., for movies) provided either by the site's owner or through an external

[10]Microsoft is a member of OASIS, but membership in a consortium has never been a bar to advocating an alternative standard. Although Microsoft is a member of the Object Management Group, it continues to tout its Common Object Model (COM) in competition with the latter's CORBA.

rating service. When developed, the standard was expressed as a parentheses-denoted Multipurpose Internet Mail Extension (MIME), type (an IETF standard designed to reformat 8-bit content into 6-bit legal characters used for Internet E-mail) and as META tags in HTML. Once XML was developed, however, PICS could be denoted as markup tag, and so parentheses were replaced by angle brackets and XML tags. But the words were the same. In time, the RDF (resource description framework) grammar will replace the PICS grammar, but, again, the words will remain. The world of digital libraries, as noted in Appendix C, provides further examples.[11]

Of note is HL7,[12] a standard way to specify and format messages to exchange, manage, and integrate data for clinical patient care (notably via admissions, discharge, and transfer systems). Although the standard has ways to describe the medical care given (i.e., what all the billing is about), it was not meant, at least originally, for doctor-to-doctor communications but for medical E-commerce. The standard appears to be well-established (the parent body, also called HL7, had 1,700 members in 1998), but the standard is not meant for casual use: Two parties who agree to implement the standard must write an auxiliary specification that specifies event triggers, messages, and optional fields used and omitted (so as to trim the broad list of data elements otherwise required). As with many heavyweight standards, HL7 is more suited for interoperability within an enterprise than among enterprises. (Lincoln et al., 1999.) Since starting in early 1987, HL7 has shifted from OSI to the now-ubiquitous TCP/IP. Moving it further to XML may represent a larger change because HL7, although transport-independent from its inception, was developed to encode messages according to strict rules.[13] Developing a DTD for HL7 and then extracting HL7's semantics apart from its syntax would be major changes that would have to be carefully engineered to ensure that the structural information in the current speci-

---

[11]For instance, ten years ago NIH adopted ASN.1 for PubMed classification. Having mooted CORBA, NIH is shifting to XML.

[12]The *7* in *HL7* refers to the seventh or application layer of the OSI model. Like OSI, HL7's developers wish to bracket the standard with reference models and usage profiles.

[13]As of 1998, developers looking toward HL7 version 3 (version 2.3.1 became an official ANSI standard in May 1999) were trying to put it over an object-oriented methodology. (See Hentenryck, 1998.)

fication is not lost in the new XML rendering (even as the overall HL7 message envelope persists).

HOW XML MAY FAIL

The most obvious way that XML may fail is that the promise of interoperability may be lost in the welter of competing semantic standards that use the XML syntax. But there are other ways to fail.

Other Standards for E-Commerce May Arise

Some proposed standards for Web commerce are incompatible with XML. A UN group is promoting Object-Oriented EDI (OO-edi).[14] OO-edi comprises two views: (1) a Business Operational View (BOV), which defines parties to the exchange, their roles, business processes, agreements, and data, and (2) a Functional Service View (FSV), concerned with implementation details, such as the syntax and method used, communication protocols, and application interfaces. The Universal Modeling Language was then selected for business process and information modeling. Although the group favors BOV, it avers that XML can be used with one of many types of FSV implementation. However, XML's use within an OO-edi environment would require a tricky data mapping to business objects, whereas pure OO-edi does not require it. (Harbinger Corp., 1999.) The UN group has not endorsed the XML/EDI Group promotion of XML as the FSV solution for OO-edi. (Webber and Naujok, 1998.) (The complexity of this paragraph provides a good hint about the standard's prospects.)

Business system interoperation (BSI) is an approach to EDI that uses BSI servers at each end of an E-commerce transaction for encoding and decoding. A perhaps fatal limitation of this method is that it requires exchanges of updates between trading partners every time one of them makes a change to its internal process or software. A project on BSI in the reinsurance industry is being supported under Europe's ESPRIT IV and undertaken by the Distributed European System Interoperability for Reinsurance (DESIRE) consortium.

---

[14] The Techniques and Methodologies Work Group (TMWG), charted by the United Nations Centre for the Facilitation of Procedures and Practices for Administration, Commerce and Transport (CEFACT).

Although CEN/ISSS initially supported BSI (see CEN/ISSS, 1998c) it formally withdrew from the project in mid-1998.

Electronic Data Markup Language (EDML) is a metadata coding system for use in defining the NAME component of the META construct in HTML. According to its creators, it is not intended as a competitor to XML but can be used as a stand-alone (Galbraith and Galbraith, 1998)—but if it works, XML may not be needed for E-commerce applications.

Such approaches are, at worst, distractions. The UN effort is clearly the work of structuralists who believe that a rigorous descriptive architecture of any realm must precede (or substitute for) its semantics.

Too Much Capital May Have Been Sunk into Today's X12- and EDIFACT-Based EDI

Companies that use EDI now have large investments in EDI software and may be reluctant to throw it all away. Major EDI service suppliers, like GEIS (General Electric Information Services), are developing Web-based EDI applications, which might prolong EDI's life. Since a transition from X12 or EDIFACT to XML requires some form of translation, at least for legacy systems, it is not clear that moving into XML-based commerce will make economic sense in many cases. To succeed, XML product suppliers will have to provide flexible and scalable interfaces with a variety of legacy business systems—an untested capability. Indeed, reducing the cost of EDI may not be in everyone's interest: Large firms may look at the cost as a way of testing the seriousness of a vendor's commitment, while vendors who have made the requisite investment can regard such costs as a barrier to new entrants. Finally, but by no means decisively, XML-based transactions will also require somewhat more bandwidth—one estimate is roughly 15 percent (EPIFOCAL, no date)—than traditional EDI transactions.

XML Is Still Too Complex

Because XML is not new, but a skinny version of SGML, it may not reduce the complexity of SGML enough. (Cover, 1998.) XML is itself complex, and many XML applications proposed include DTDs,

themselves quite complex. XML has to feel right to the average HTML coder before it attains the ubiquity to replace HTML.

It May Get Caught in the Browser Wars

If HTML's history repeats itself, XML may suffer from having different browser makers include various nonstandard features. According to the Web Standards Project, an international coalition of Web developers and Web experts, Internet Explorer 5.0 does not fully implement the XHTML 4.0 standards that Microsoft helped develop. While some standard features are missing, others are implemented in a way that would make them incompatible with other standard-complying authoring tools. (Olsen, 1999; Bray, 1999.) Since Netscape announced that Mozilla will be fully compatible with the XML standards, a repeat of the "browser wars" may be in the offing.

It May Get Caught in the Java Wars

Combining XML-marked-up data with cross-platform software, such as Java, allows the formation of movable objects. XML is platform-independent data, while Java is platform-independent software. Sun's Director of Java Software, Jonathan Schwartz, maintains that XML, together with Java, can support the requirements for reuse of information across arbitrary and idiosyncratic computer systems and display devices. (Alshuler, 1999.) The combination would also result in acceptable implementations of object-oriented EDI.

So why is Microsoft embracing XML so hard in its Biztalk effort—which combines an active registry program with vertical marketing of Microsoft products into the E-commerce sector and efforts to make future browsers XML-aware? Even though there is no reason that Java code cannot work with XML-formatted documents, an applet-centered world and a document-centered world pull people in different directions.

In an applet-centered world, the server provides the data and the applet to manipulate it; the data need not be formatted in any fashion that outsiders have to agree to. Why? The definition and treatment of the markup come from the same institution that produces the applet. It suffices only that the applet recognizes what the tags

mean; users do not have to. The wide use of the XML grammar can make applets easier to write because the tools to manipulate marked-up text will be widely available, but the words need not be standardized.

In a document-centered world, the tags would have standard meanings. That being so, off-the-shelf software can be built to recognize the denotations and connotations of the tags to manipulate the document. Applets are no longer as necessary because the manipulation capability can be built into the browser or an add-on. Thus, Microsoft's approach requires XML to push beyond grammar to words; Sun's approach exploits XML for the regularities in the grammar.

Sellers May Not Like Friction-Free Capitalism

Not every seller, after all, wants to be compared on the basis of a particular attribute to the exclusion of other attributes (e.g., revealing price but not customer support and thereby encouraging commoditization of the pricing structure). Nor do all sellers want to allow their sites to be searched by bots, thereby losing the ability to present their terms to human decisionmakers. With current technology, some sellers limit access to their sites for nonhuman visitors. When implementing their catalogs in XML, sellers might adopt nonstandard tags or might design their sites in a way that provides the information they choose to provide, regardless of the information requested, e.g., information on product or service bundles only. This is not necessarily a bad idea. Depending on the seller's brand and market power, it may be in a position to demand and get different trading terms than less successful competitors. XML provides a possibility of a level economic playing field in which consumers would benefit; it does not necessarily create conditions under which sellers will want to play.

Trust, Not Standards, May Be the Problem

Here too, XML alone may not suffice until and unless issues that relate to the social aspects of business are put to bed (see Appendix D's discussion of security and payments). One such issue is trust. Will every buyer that contracts for a purchase have the funds to pay

for it? Will sellers deliver the promised goods on schedule and at expected quality levels? It is always risky for new buyers and sellers to transact business until they build a record of fulfilled transactions and trust. Part of the "value added" that such intermediaries as General Electric Information Services provide is the screening of buyers and sellers, increasing comfort levels for both parties. While a global market is a theoretical nicety, relying on the kindness (or probity) of strangers is still a lot to ask.

CONCLUSIONS

XML, if it works, may very well be the heart of tomorrow's Web because documents structured in a standard can be understood and thereby manipulated by stupid but fast and cheap machines rather than intelligent but slow and expensive humans. But despite the enthusiasm with which XML is being offered to, and, accepted by the world, the hard work lies ahead. Whether the XML standards processes can result in commonly defined terms within (and, perhaps more importantly, across) the disparate communities of commerce is yet to be determined.