Standards Today

A Journal of News, Ideas and Analysis

A publication of CONSORTIUM INFO.ORG GesmerUpdegrove LLP

October - November 2009 Vol. VIII, No. 6

SETTING INFORMATION FREE: THE WORLD OF XML

The Information Age of the 20th Century was great, but the Digital Age of our new Millennium can be even better – that is, if you can find what you're looking for.

The Internet and the Web get most of the glory, but without an ever-growing cast of supporting standards, they would be a far poorer place to explore. One of the most fundamental of those unsung standards is XML.

Before there were electronic documents, information could only be gathered by hand from multiple sources, and then combined into new documents that in turn became static. The same would be true for electronic documents today, if it weren't for the Extensible Markup Language and the seemingly endless stream of derivative languages it has made possible.

Thirteen years ago, XML went from concept to first draft in just 20 weeks. Since then, it has brought order to untold millions of documents around the globe. In this interview, XML Co-Editor Tim Bray talks about how XML came to be, what it has become, and how we'll share information on the Web in the future.

Some things never seem to change. One is the desire of some vendors to conquer the world through the use of proprietary standards, rather than share even greater wealth by using open ones. Another one is that this strategy rarely works for long.

Standards Today is a free bi-monthly electronic Journal sponsored by the Boston law firm of Gesmer Updegrove LLP. The current issue of ***Standards Today*** and a subscription form may be found at www.consortiuminfo.org/bulletins. Questions or comments about the articles in this issue or about ConsortiumInfo.org may be directed to Andrew Updegrove at updegrove@consortiuminfo.org, or by telephone at 617/350-7800. To learn more about Gesmer Updegrove's standards and open source practice, visit http://www.gesmer.com/practice_areas/consortium.php, or contact Andrew Updegrove.

EDITOR'S NOTE:

A Standard for a Digital Age

> *Zettabyte: noun; (a) 1,000,000,000,000,000,000,000 bytes = 1000^{7}, or 10^{21}; (b) the amount of new data made available on the Web each year.*

It is now some decades since the phrase "Information Age" entered our language. What led to the coining of the phrase was the transition of the United States from a manufacturing-based, blue collar, industrial economy to one driven by services and office jobs – and hence the production of information rather than tangible goods. Computers played an important part in that transformation, but in the days of mainframes, that role was in the background.

With the arrival of the Internet and the Web, the phrase "Digital Age" replaced the Information Age in common parlance. That semantic distinction recognized, among other revolutionary transitions, that access to electronic information could become universal. And indeed, with mobile computing devices now replacing cell phones as affordable mass-market products, digitized information is becoming accessible to almost everyone, even in the Third World. This new ability to exchange information everywhere has the capacity to truly transform lives around the globe.

Accomplishing that enormously beneficial goal takes more than an Internet connection, though. Just as was the case with the Internet and the Web, it takes standards to permit the universal creation, storage, searching, and sharing of the richer and often more complex information to be found in lengthy documents, rather than simple Web pages. One standard in particular stands head and shoulders above all others in this respect, and that's what this issue is about.

That standard is the Extensible Markup Language, or XML. Just a decade after its creation, XML is the foundation upon which an ever-expanding family of markup languages and related schema (hundreds of them) has been built. Without XML, digitized text would still be almost as unmanageable as it was when found only in tangible media, warehoused in libraries and records rooms.

In this month's Editorial, I expand on the significance of XML, while noting that the task of realizing the promises of the Digital Age will not end with its development. My ***Feature Article,*** as usual, takes a deeper dive, examining the origins of XML, explaining how it operates, and surveying the many language and schema progeny that it has spawned. I follow with an ***Interview*** with Tim Bray, one of the co-editors of XML and someone who has helped to realize its promise in the years that followed. Tim gives his own first-hand take on how XML came about, and why it matters.

In my ***Standards Blog*** entry, I bring us back into the rough and tumble of the real world of commerce and standards (in this case, for eBook readers). The eBook example demonstrates that while you can lead an industry to XML, only market pressures can ultimately force a dominant player to implement it. Happily, I think that it will only be a matter of time before the implementation of open, XML-based standards becomes ubiquitous in this niche as well.

I close this issue with some ruminations on a different kind of format – the wonderful, multi-faceted, ever-evolving musical genre that is Jazz. In this month's ***Consider This*** essay, I riff on the easily traceable evolutionary thread that runs from jazz musicians to computer engineers, and from jazz standards to open source software. Not so? Read it and decide for yourself.

As always, I hope you enjoy this issue. But either way, it's always great to hear what you think. Let me know, why don't you?

Andrew Updegrove
Editor and Publisher
2005 ANSI President's
Award for Journalism

The complete series of Consortium Standards Bulletins can be accessed on-line at http://www.consortiuminfor.org/bulletins/ It can also be found in libraries around the world as part of the EBSCO Publishing bibliographic and research databases

Sign up for a free subscription to ***Standards Today*** at

http://www.consortiuminfo.org/subscribe/2.php?addentry=1

EDITORIAL:

Tagging the Noosphere

Andrew Updegrove

One of the many intriguing concepts mooted by Pierre Tielhard de Chardin, a French philosopher and Jesuit priest with polymathic insights (his academic explorations range from paleontology to the meaning of the Cosmos) is the "noosphere." In de Chardin's vision, the reality of the world encompassed not just the geosphere (inanimate matter) and biosphere (all forms of life), but an ever expanding nimbus of knowledge representing the fusion of the minds and knowledge of all humans.

The postulation of a noosphere was appealing in its simplicity, but in those pre-networked days (de Chardin died in 1955) it was without much practical application. Even as knowledge continued to expand, information remained sequestered in hundreds of disparate languages, and archived in millions of globally distributed libraries. De Chardin's concept could therefore at best be considered a philosopher's abstraction – an interesting paradigm to be bandied about in conceptual discussions.

Must not the ability to synthesize and share all of the world's information in real time lead to another great leap forward?

With the advent of the Internet and the Web, though, de Chardin's noosphere seemed to have become real rather than abstract. With so much accessible so easily to so many, the philosopher's vision of the noosphere as the foundation for the next evolutionary step of the cosmos seemed plausible, or at least a jumping off point for the next major advancement in humanity's own developmental path. Indeed, the emergence of language allowed the first humans to share individual discoveries, and the development of writing permitted knowledge to be more widely shared and more reliably passed on to future generations. Each of these major advances had unquestionably provided the basis for new and dramatic advances in the development of human society. Must not the ability to synthesize and share all of the world's information in real time lead to another great leap forward?

One didn't need to buy into de Chardin's more abstract views of evolution to agree that indeed it might. But before such a transcendental (or even a less cosmic) next step could be taken, a great and invisible void remained to be filled, less obvious than the need for more powerful telecommunications lines and ubiquitous computer access, but equally essential and challenging to address. That abyss was the lack of the tools needed to manage and make sense of the constantly burgeoning flood of data that lay tantalizingly just beyond our grasp. Only if new automated tools could be developed to create, store, rediscover, and synthesize that data could the riches of the noosphere be realized.

When we think of that challenge, we are likely to think of Google, and assume that the creation of ever more sophisticated search algorithms will be sufficient to allow us to delve into the remotest corners of the Web. But essential as search technology may be, it would at best be an imperfect, inefficient and very user-unfriendly tool had not a series of mostly unknown and unsung software engineers, supported largely by corporate sponsors, devised the vital handful of standards needed to allow the noosphere to become practically accessible to us all.

One of the most important of those standards is the Extensible Markup Language, or XML, which allows any piece of information to be "tagged" once at the time of creation to make it not just now, but forever identifiable for any of a number of different purposes, both mundane and vital (e.g., as the contents of a table that needs to be formatted as such, or as the genus name of a new species). Moreover, as indicated by the "Extensible" in the standard's name, XML is a flexible tool that allows anyone to create their own tag set to make the recorded information of their unique knowledge domain universally manageable and more self aware, whether it be Old Testament scriptural references, chemical compounds, baseball batting averages or financial reporting data.

How pervasive has the use of XML become? The following is taken from the W3C press release marking the tenth year anniversary (on February 12, 2008) of the formal adoption of XML:

> Indeed, one can hardly get through the day without using technology that is based on XML in some fashion. When you fill your auto tank with gas, XML often flows from pump to station. When you configure your digital camera, on some models you do so via XML-based graphical controls. When you plug it into a computer, the camera and the operating system communicate with each other in XML. When you download digital music, the software you use to organize it is likely to store information about songs as XML. And when you explore the planet Mars, XML goes with you.

Today, just 11 years after the first release of XML, there are hundreds of XML languages, schema and supporting standards. Because of standards like XML (and HTML and the Unicode), the noosphere has morphed from a philosopher's foil to a boundless resource to be mined by the great and the humble, the rich and the poor, wherever they may be.

XML will not be the last standard we will need to fully capture the promise of the noosphere. But it is one of the small set of foundational standards that have set us on our way into a future that could not have been imagined but a short time ago. Except by visionaries, like de Chardin, who were able to look past the horizon of time to imagine a world that it will be our privilege to experience first hand.

FEATURE ARTICLE:

XML and its Many Children: Bringing Order to a Digital World

Andrew Updegrove[1]

Abstract: *Prior to the advent of computers, information was necessarily stored in tangible media that was searchable and understandable only through visual examination. With the advent of the Internet, both the opportunity and the challenge of automated access to knowledge were magnified a billion times. In the 1990s, it became clear that the riches of digitized data could only be mined if elements of text could be identified in such a way that they could be readily exchanged between computer systems of any type without losing knowledge of their own format and structure. Moreover, by permanently "tagging" elements of text with semantic, as well as formatting information, the data in documents could become self-aware, allowing information to be more intelligently searched, manipulated, and compiled. The mechanism invented to achieve this end was a standard called the Extensible Markup Language (XML), a tool that was strict enough to achieve the interoperable exchange of information, but flexible enough to allow the creation of a derivative based language to order and make greater sense of any domain of knowledge. In this article, I describe the origins, development and impact of XML, and the standards development organizations that maintain and continue to develop these essential tools of the Digital Age.*

Introduction: Prior to the Internet era, the science of organizing, archiving and accessing information evolved at a leisurely pace. With all information fixed in tangible media, only the highest level of indexing made sense, lest the volume of index materials become too cumbersome to be useful. In due course, libraries adopted common conventions for indexing (via subject, author and title) and archiving (using the Dewey Decimal System) that provided useful, albeit slow and limited, ways for researchers to locate what they needed.

Once computers became powerful and sophisticated enough to analyze the contents of entire books in nanoseconds, the need for research tools capable of capitalizing on this raw power became obvious. More was needed than simply old wine in new bottles, though, because the reach of the new technologies was orders of magnitude greater than the conceptual grasp of the card catalog metaphor.

The advent of the Internet, and more particularly the hyperlinking capability of the World Wide Web and the development of highly effective browser technology,

[1] Disclosure: the author and his law firm have acted as legal counsel to a number of entities mentioned in this article, including the Association for Cooperative Operations, Research and Development (ACORD), Open GeoSpatial consortium (OGC), Organization for the Advancement of Structured Information Standards (OASIS), and XBRL International, Inc. (XII).

raised the ante even more substantially. Soon, the volume of data becoming accessible presented a veritable Black Hole of knowledge in reverse, exposing gigabytes of new information to virtual access on a daily basis.

Simple access to data, however, does not equate to an ability to make practical use of that information. A way was needed to make the text of digitized documents transportable and able to be displayed with its original formatting intact on the proprietary computer equipment sold by every vendor. And in order for the information contained in electronic documents to be automatically extracted and combined with other data of a similar type, information of particular types needed to be identifiable as such (e.g., as annual profits, or lab test results or census data). Otherwise, we would simply drown in our own newly accessible data, no better off than before.

Simple access to data does not equate to an ability to make practical use of that information

Making data practically useful therefore required the development of new technical means to make information machine-readable, so that computers could manipulate it most efficiently, in the first instance, and do useful things with it in the next. Since the goal was to be able to access, share and redisplay information anywhere on the Internet, data needed to remain human-readable as well. In short, making the data and knowledge of the world electronically accessible required new standards - and lots of them.

Today, there are a variety of standards that deal with information on the Internet. But in many ways, the DNA that lies at the heart of categorizing, storing, accessing, manipulating and presenting electronic information is a single standard. Or, more properly stated, a single matriarch standard that has given birth to, and continues to nurture, an almost limitless number of specification descendants. That standard is the Extensible Markup Language – more familiarly known simply as "XML."

What is XML? Simply stated, it is a text format standard, with supporting tools, that allows structure and meaning to be given to electronic documents (broadly construed) through the use of machine-readable, standardized tags. Most importantly, XML is a flexible standard that not only allows a programmer to give digital meaning to words and sections of text, but also to decide what that meaning should be. The result is that everyone from biochemists to Old Testament scholars to ad writers uses XML to create languages unique to their needs.

In this article, I will review the origins and development of XML, the organizations that develop and maintain XML-based standards, and the explosive growth and influence of XML in the decade following its formal adoption in 1998.

I XML Origins and Development

XML did not spring full blown from the minds of the World Wide Web Consortium (W3C) Working Group members that created it. Rather, it evolved from a number of predecessor specifications that systemized ways of electronically annotating text to give additional, computer-readable meaning to text of various types (i.e., “this is a table,” “this is a chapter heading,” and so on). In effect, these precursors played a role similar to an editor’s traditional blue-pencil marginal comments on pre-print text – and hence the name "markup languages." But unlike an editor’s penciled marks on fixed copy, markup language text can remain(to human eyes) invisibly embedded forever, available to instruct future processors how to more usefully work with the text in question.

GML and SGML: The immediate predecessor to XML was the Standard Generalized Markup Language, or SGML, adopted by the international Organization for Standardization (ISO) in 1986. SGML, in turn, was based in part on an earlier specification called the General Markup Language (GML). That language was developed within IBM, beginning in the 1960s, by a team led by Charles Goldfarb, Edward Mosher, and Raymond Lorie.[2] The goal of the project was to create a robust means for governments and document-dependent private industries (such as the legal sector) to create large documents that would be able to retain their structure over long periods of time. But while each markup language relied on the innovations of its predecessor, each in turn also broke new ground.

In the first evolutionary step, GML’s creators sought to provide structure, but not directions, to documents. For example, GML tags would identify appropriate text as a table or a section heading, but would not go on to tell a computer how a table or a section heading should be formatted. That was left to the specific software application that might be used to open and print the document. As a result, text could more easily be exchanged among users with different software, and those users could also make independent decisions on matters such as font choice, table size, and so on.

Goldfarb went on to become a member of a committee formed in 1978 and administered by the American National Standards Institute (ANSI), called the Computer Languages for the Processing of Text committee. Later, he was asked to chair the working group chartered to create SGML. The first working draft of SBML emerged in 1980, and in 1983 the sixth draft was formally adopted by the Graphic Communications Association as GCA 101-1983. Soon, government agencies like the U.S. Internal Revenue Service and Department of Defense were using the new standard.

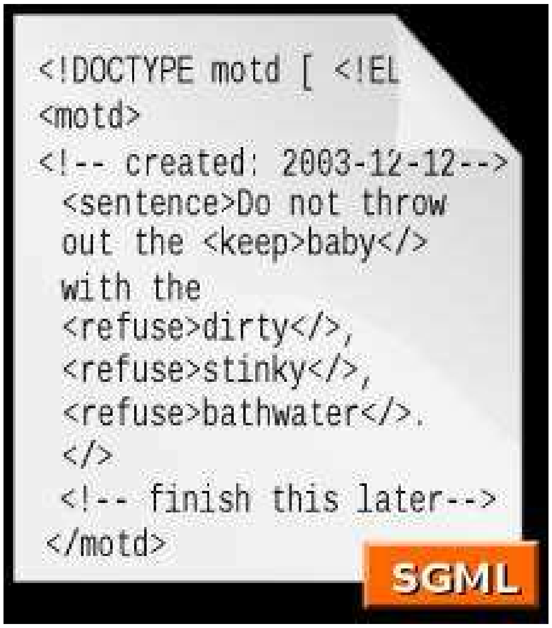
Development work on SGML continued in the ANSI committee as well as within a new working group (SC18/WG8) organized under Joint Technical Committee 1 of ISO and the International Electrotechnical Commission (ISO/IEC JTC 1). Goldfarb was project editor for each, and in 1986, the international SGML standard was published as ISO 8879:1986.

But while SGML was robust, it was also complex and restrictive. In order to conform to SGML, each document was required to conform to a “Document Type Definition” (DTD), made up of the “markup declarations” that the document could (and must) utilize. The result worked well for setting up templates for standardized documents, but the requirement to either use, or create, a DTD for each document was both limiting and burdensome.

[2] It was no coincidence that the last name initials of the three engineers also happen to be “GML.”

While the advantages of SGML played well to a world of centralized computing systems managed by experts, the highly regimented standard was not nimble enough to meet the demands of the often self-trained, hacker-mentality developers that were building out the Internet and the Web. As noted by one on-line commentator not long after XML was formally adopted by the W3C:

> The Web culture is about "freedom". To most Web geeks, DTDs are just dull and boring stuff, intolerable obstacles to their creative freedom. Joe Webmaster wants bouncing logos, blinking commercials, 3D buttons, flashy fonts, background music like in Starwars. Joe Webmaster had extensive training with MS InterDev, JavaScript, ActiveX. He had no training with SGML and does not plan to…Joe Webmaster's freedom is immense. Do you want an indication of how immense it is? Just have a look to the shelves at your favorite computer store. *Awsome [sic]*, isn't it? (Then try to find the SGML books -- if there are any....) [3]



```
<!DOCTYPE motd [ <!EL
<motd>
<!-- created: 2003-12-12-->
  <sentence>Do not throw
  out the <keep>baby</>
  with the
  <refuse>dirty</>,
  <refuse>stinky</>,
  <refuse>bathwater</>.
  </>
  <!-- finish this later-->
</motd>
```

Moreover, some features that added flexibility to SGML became disadvantages on the Internet. For example, SGML could function as an umbrella standard for other markup languages, which could be referenced in the DTD for the document being created. The result was that not every processor would be able to read every SGML compliant document – a significant disadvantage when seeking to read a document available at a hyperlinked site.

In order to preserve the benefits of SGML while addressing these issues, a less restrictive subset language was needed that would allow, but not require, the use of DTDs, and address other matters of concern. But while some recognized the need for such a standard, others did not, or believed that formalizing an SGML equivalent for Web pages was more important.

The eventual result was the development of two new standards – one inspired but not so faithfully derived from SGML, to be used for uniformly rendering Web pages (HyperText Markup Language, or HTML), and some time later, a second one, more narrowly based on SGML, for use with documents accessible via the Internet (XML).[4]

[3] *See,* Sabarthez, Laurent, Some Notes on the History of XML (August 1998), *at*: http://www.users.cloud9.net/~bradmcc/xmlstuff.html All Web site cited in this article were last accessed on December 9, 2009.

[4] HTML, like XML, is a formatting language that allows information (both fixed text, as well as dynamic elements, such as video) to be displayed. Proper use of HTML in the creation of a Web page allows any browser that makes proper use of the same standard to display information as originally

The development of XML: SGML traced its origins to the world of mainframe computers, and as a result it was developed within the leisurely, and often more exacting, process of the standard setting infrastructure that had evolved over more than a hundred years. The birth of XML, however, was far different.

Between the adoption of SGML and the chartering of the XML working group a revolution had occurred not only within the information technology (IT) industry, but within the standard setting community as well. Beginning in the late 1980s, IT vendors increasingly opted to form variously formal and informal new organizations to develop the new specifications they needed. Their motivations included a mix of frustration with the slow speed of traditional standard setting, as well as the desire to exercise greater control over the specifications that emerged from their combined efforts.

By the time the Internet began to be massively used, there were already hundreds of these "consortia" in existence, many of which had become institutionalized, as well as the centers of domain effort within their self-appointed areas of competence. As a result, while a great deal of IT standard setting continued within traditional standard setting organizations, the majority of activity, and hence the center of influence, in IT standard setting had passed to the consortium world.[5]

One of the most successful and respected of these new consortia was the World Wide Web Consortium (W3C), conceived and founded in 1994 by Web inventor (now Sir) Tim Berners-Lee at the Massachusetts Institute of Technology (MIT) Laboratory for Computer Science. The immediate inspiration for the effort was the evident need for a single, universally implemented HTML standard, but over time the W3C became the venue of choice for the development of a variety of standards directed at improving the effectiveness of the Web, and ensuring that its benefits could be shared throughout the world.

As a result, when the need became clear (to some) for a new version of SGML that would be optimized for use on the Internet, the W3C seemed like the logical venue in which the effort should be launched. But as the Internet became more economically important, the design of the technology underlying it became more strategic, and particularly so after the meteoric rise of Netscape Communications Corporation, which enjoyed one of the most successful initial public offerings in history on August 9, 1995. Suddenly, the question of what "SGML on the Web" should mean became a subject of significance, although only a few fully appreciated that fact.

Microsoft was one of those that did, in part because for some time it had famously missed the importance that the Internet and Web would assume. As a result, when Netscape came to own the suddenly hot market for Web browsers, Microsoft was left scrambling. In response, it mounted an urgent effort to counter the instant success of Netscape's Navigator browser with its own hastily launched Internet

intended. The faithful use of additional standards ensures that those with vision, hearing or other disabilities will also be able to access the same information.

[5] The most extensive available list of active and inactive IT consortia is maintained by the author, and can be found at: http://www.consortiuminfo.org/links/ Over time, ISO/IEC developed processes that allowed consortium-developed standards to be submitted to, and approved by, JTC1 working groups, thus allowing these standards to gain the imprimatur of the traditional standards regime.

Explorer software. But while Microsoft came late to the Internet party, it recognized the importance that an Internet-optimized subset of SGML could play.

The result was the formation within the W3C of the "Generic SGML Editorial Review Board" in 1996, chaired by Jon Bosak, of Sun Microsystems. At the same time, a "Generic SGML Working Group," was formed, which in turn was supported by a Special Interest Group. The design goals of the new working teams were as follows:

> *Incredibly, the first working draft of the XML standard was released only twenty weeks after work began*

1. XML shall be straightforwardly usable over the Internet.
2. XML shall support a wide variety of applications.
3. XML shall be compatible with SGML.
4. It shall be easy to write programs which process XML documents.
5. The number of optional features in XML is to be kept to the absolute minimum, ideally zero.
6. XML documents should be human-legible and reasonably clear.
7. The XML design should be prepared quickly.
8. The design of XML shall be formal and concise.
9. XML documents shall be easy to create.
10. Terseness in XML markup is of minimal importance.[6]

As with all other W3C Recommendations, the final product would be distributed for free.

Work began in July of 1996, with James Clark as technical W3C Technical Lead, and Tim Bray and C. Michael Sperberg-McQueen as co-editors. Although the SGML standard did not provide the sole reference point for the new standard (the Working Group also borrowed ideas from the HTML and HTTP standards, among other sources), XML was primarily intended to be a selective, narrower profile of SGML.

Incredibly, the first working draft of the XML standard was released only twenty weeks later – on November 14, 1996. The final version of XML 1.0 was formally adopted as a W3C "Recommendation" (i.e., standard) on February 10, 1998. Along the way, there were intense efforts by both individuals (with strong feelings on technical matters), as well as competing companies (with huge investments riding on their ability to successfully navigate the rising tide of the Internet), to influence the final form of the standard.[7] In all, eleven individuals comprised the original working group, laboring through weekly teleconferences and via email, while as many as 150 others participated in the active email discussions via the working group listserv.

[6] "1.1 Origin and Goals," Extensible Markup Language (XML) 1.0 (Fifth Edition) *at:* http://www.w3.org/TR/REC-xml/#sec-origin-goals

[7] For a highly personal account of the rough and tumble development of XML and profiles of the individuals involved, *see* XML is Ten Years Old Today, posted by XML co-editor Tim Bray to his blog in 2008 at http://www.tbray.org/ongoing/When/200x/2008/02/10/XML-People. Some of the sharpest elbows were thrown when Bray took a consulting position with Microsoft arch-competitor Netscape midway through the development process. According to Bray, Microsoft insisted that he be removed as a co-editor; in an eventual compromise, he retained his role, but Jean Paoli, a Microsoft employee, was added as a third co-editor.

Following the adoption of XML 1.0 the W3C (and the wide adoption of the standard in the field), XML development work was expanded and restructured under the direction of a newly chartered XML Coordination Group and XML Plenary Interest Group. The actual design work would now be conducted in five new XML working groups addressing topics such as XML Schema and XML Syntax; internal liaison relationships were established with other W3C working groups active in technically adjacent areas to ensure overall architectural coherence.

Despite its rapid development, XML proved to be remarkably durable. Today, XML 1.0 is in its fifth edition, but the changes to it have been minor. A version 1.1 of XML was adopted in February of 2004, but its use has been less widespread. The stability of the standard is due in part to the fact that XML has proven to be robust and useful, and also because, once implemented, any standard is difficult to change without introducing incompatibility problems with documents and applications already created.

What it is: In concept, XML is disarmingly simple. Like its partner in digital presentation, HTML, XML employs "tags" to separate content and assist computers in more usefully and easily dealing with that data. Tags are identifying labels contained within angle brackets, as follows: "<tag>." There are three basic types, two of which are used to enclose the content to which they relate (called start tags and end tags, such as <address> and </address>) and "empty element" tags (e.g., <line-break/>), which separate content. Tagged content, and empty element tags, can be nested within and between other tags, permitting the marking up of content within content (e.g., minor heading content within major heading content).

Tags can also serve other purposes, such as assigning "attributes" to text by pairing tag information of one type with a value. For example, tags can not only identify a line of text as a section of a document, but also assign a section number to the text enclosed by the start and end tags. Unlike the section number that is visible to the reader, but meaningless to a computer program, the value associated with a section heading tag can be machine-readable. This allows the software application in which the document is composed to take appropriate actions in relation to the tagged information, such as automatically adding a similarly numbered section line to the index of the same document.

Tags can also add important search capabilities to documents, by giving machine-readable meaning to individual words. For example, in an XML language for legal documents, a party can be designated as the plaintiff rather than the defendant. A legal search using this "metadata" can therefore find (for example), only cases where Company A is a plaintiff, and not a defendant, and where Company B was a defendant. Tags can also indicate the technical nature of content (e.g., a picture would be identified with the tag "<img>." A search of a news archive could therefore be performed to find only pictures of a specific person in a boat, and exclude all text references to the same individual, whether or not in a boat. The tagged data in question therefore "knows" what it is, and can identify it as such to a search function.[8]

Tags take advantage of another fundamental standard called Unicode. The Unicode is the product of a long-running project with the mission of ultimately making every character set of the past, present and future machine-readable.[9] XML tags therefore are allowed, with limited exceptions, to include only Unicode characters.

When a programmer properly adds XML tags to an otherwise appropriate document, the result is a file that can properly be read by any software application running on any operating system that has itself been developed in a compliant manner. But while the result is easily read by a computer, it is not so obviously interpreted by anyone not skilled in the programming arts. The following is a very simple example taken from the Wikipedia:

When a programmer adds XML tags to a document, the result is a file that can properly be read by any compliant software application running on any operating system

```
<?xml version="1.0" encoding='UTF-8'?>

<painting>

  <img src="madonna.jpg" alt='Foligno Madonna, by Raphael'/>

  <caption>This is Raphael's "Foligno" Madonna, painted in

  <date>1511</date>-<date>1512</date>.</caption>

</painting>
```

When interpreted by a work processing application, the above code would display an image of a painting by Raphael, together with the caption "This is Raphael's 'Foligno' Madonna, painted in 1511-1512." Moreover, the document would "know" (and therefore a search function could be informed) that the image was of a painting, that "Foligno" in this case (because the caption lies within the start and end tags for "painting" was the name of the painting, and that the work of art was created during the years noted – something not otherwise possible without the added information provided by the tags.

Unfortunately, many tags are not so intuitively named. As a result, properly tagged text, and especially highly formatted text, rapidly becomes incomprehensible to anyone other than a programmer. The following is a sample of a simple document header instruction:

[8] A far more ambitious effort to improve the abilities of computers and search engines to automatically perform searches more intelligently is the long-ongoing Semantic Web effort of the W3C.

[9] The staff of the Unicode represents a group of unsung heroes doing yeoman service for the betterment of all mankind. For more on the significance of the Unicode, see my blog entry, The Unicode Standard 5.0: an Appreciation, *at:* http://www.consortiuminfo.org/standardsblog/article.php?story=20061017163856508

<h level="3" class="test">The Editor in the Hat</h>[10]

Complexity and purpose: XML formats can be as complex or as simple as the task at hand requires. They can also be used for a variety of different purposes. Both points are well illustrated by one of the major standards wars of the last decade, which played out across two consortia, ISOIEC JTC1, and the national standards committees of scores of countries around the world.

The subject material for the drama was the ubiquitous office productivity software suite, which includes word processor, spreadsheet, slide presentation, and database modules, each of which is expected to be able to exchange data with the others. With such heavy formatting needs, the effort required to create a robust standard capable of preserving so much detail can be great indeed. Nevertheless, over time, there was increasing consensus that XML-based formats should provide the foundation for each of these modules.

The value of such a conversion was clear, for both internal as well as external reasons. For a proprietary vendor like Microsoft, the conversion of its Office software suite to an XML-based format would make it easier for Microsoft to upgrade its products in the future, and also make it simpler for the many independent software vendors (ISVs) that are part of the Microsoft Office ecosystem to keep their own products interoperating successfully with new versions of Office. This would in turn make it easier for customers of Microsoft and Microsoft ISVs alike to exchange information among the products they purchased from each.

As a result, Microsoft began to move away from its historic, binary formats to a new XML-based format that it named Office Open XML format, or OOXML. Microsoft offered OOXML to ECMA, a European-based consortium in 2006. After adopting OOXML, ECMA proposed the resulting specification to ISO/IEC for adoption the following year. In 2008, OOXML became ISO 29500.

But a second XML-based document suite format standard, called OpenDocument Format (ODF), had already been created by a consortium called OASIS (discussed in greater detail below). ODF was approved by the members of OASIS in 2005, submitted to ISO/IEC JTC1 later the same year, and adopted as ISO 26300 in 2006. The ODF standard had been created for quite a different purpose: to allow multiple, independent office suites (and other types of software) to coexist, each able to exchange documents with the other. Customers could thus choose among a variety of competing, but compliant, products without concern over being "locked in" to the products of any vendor, since they could easily move to the offerings of a competing vendor any time they wished.

[10] While simpler than SGML, XML is still no treat for newbies. *See*, for example, this sentiment from a book on coding using XML and Perl (another programming language):

> Many people, understandably, think of XML as the invention of an evil genius bent on destroying humanity. The embedded markup, with its angle brackets and slashes, is not exactly a treat for the eyes. Add to that the business about nested elements, node types, and DTDs, and you might cower in the corner and whimper for nice, tab-delineated files and a split function.

Ray, Erik T. and McIntosh, Jason, Perl and XML, Section 1.2 (O'Reilly 2002) *at*: http://docstore.mik.ua/orelly/xml/pxml/ch02_01.htm.

Each standard was therefore intended for the same general purpose – to allow information to be exchanged within documents and software products, and for documents to be exchanged between systems, in each case without loss of data integrity or formatting. But since the business goals underlying the creation of each standard were different, the resulting standards were as well.

Because ODF was intended to enable documents to be exchanged between a wide variety of software products, both desktop based as well as remote (i.e., "in the cloud"), and proprietary as well as open source, the creators of ODF sought to strike a traditional balance between level of detail and freedom to innovate above the level of standardization. As a result, the final specification was c. 700 pages long – quite lengthy for a standard of any type, but most standards are not required to address such a long and detailed list of parameters.

OOXML, on the other hand, had a different goal: to permit the faithful replication of every aspect of a single proprietary product - Microsoft Office - down to the finest detail. The result was a specification that weighed in at over 6,000 pages, filling six binders that, piled one atop the other, stood four feet high.

Happily for all, most XML languages and formats can be far shorter in length. But the example of ODF and OOXML nonetheless illustrates the extremely wide range of requirements to which XML can accommodate.

The OOXML Standard Stack

Schemas and more: While XML was intended to be a narrower and more restrictive version of SGML, over time it replaced SGML for almost all purposes, both on and off and off the Internet. In a recursive twist, the W3C created its own narrower version of the SGML DTDs that had helped inspire the XML development effort to begin with. The specification for creating what were called "XML Schemas" became a W3C Recommendation in May of 2001, and enabled the creation of shared XML vocabularies and rules to define the structure, content and semantics of an XML document. True to the original spirit of creating XML as a simpler subset of SGML, far more XML documents continued to be created without reference to schemas.

Today the XML environment is supported by a variety of other W3C efforts, including 10 standing working groups.[11] These working groups create standards and tools intended to make XML documents more useful, such as tools to assist in formatting, exchanging and searching XML documents.

[11] Current W3C XML Working Groups are listed at the home XML page, found here: http://www.w3.org/XML/ The status of current efforts are listed at the XML Activity Page, found here: http://www.w3.org/XML/Activity

II The World of XML Implementation

The advent of the Digital Age has presented information-handling challenges at every level. But the inescapable need to wrestle with issues such as formatting and the reality of proprietary hardware and software has also offered a unique opportunity to increase the ability to exchange information across all languages, cultures and distances. Through the steady progress of the Unicode, for example, a single computer can work with documents composed not only in modern German, Japanese and Arabic, but in ancient Sumerian and Babylonian as well – each in their own unique scripts.

XML likewise provides the means to take a major step towards putting the vast archive of human history and creativity into a more universally usable form. It does so by providing, not a single language, but the computer-linguistic (as it were) tools to create an infinite number of domain-specific languages. The response in the marketplace has been nothing less than phenomenal.

The secret behind the success of XML lies in its simultaneous rigidity and flexibility. Rigid, because the way that XML tags are created, used and read must remain the same so that software can use a single methodology to address them. But flexible, because anyone can create a new XML language, for whatever purpose she may wish, and generic computer systems will nonetheless be able to work intelligently with those tags.

A language for every purpose: The result has been an explosion of efforts to create custom tag sets comprising new XML languages that can be used to work more efficiently with data of any kind imaginable, from sports information to periodical advertising to financial information. Most significantly, tags can be used not only to designate factual data, such as street addresses and section headings, but anything else as well. Once labeled, information can be selected, manipulated, combined, and displayed more intelligently.

The use of XML permits the automatic population of charts, tables and spreadsheets with data that otherwise would need to be extracted, totaled and manually reentered as a separate step. Moreover, using XML related tools (such as XML Query), XML tagged data can be combined not only from text documents of a similar type, but from all of the following at the same time: text documents, databases, Web pages, and spreadsheets.

XML can be used to address the narrowest of niches as well as the broadest. At the universal end of the spectrum, one can find XML languages such as XBRL, created to permit multinational and national enterprises the world over to identify and present financial data in a uniform fashion, thus enabling global financial information to be more easily consolidated within multinational corporations, and for regulatory filings to be made and examined more easily. The narrow end of the spectrum is suggested by the following excerpt from an article that appeared in the financial press less than three years after the formal approval of XML:

> XML has taken a strong foothold in the financial services industry, and the weather derivatives market is next in line for a standard trading protocol of its own. The Weather Risk Advisory, a software

> and consulting company focusing on weather derivatives, is leading an initiative to develop WeatherML, an XML-based data protocol that looks to be a standard for the electronic processing of weather derivatives….
>
> Currently,…weather derivatives trading [is] a tedious and manual process for all parties involved. "For example, trader A has to enter the details of a transaction on the system and the system holds an internal representation of that transaction….Counter parties to that trader would have their own systems with internal representation of the transactions data." In other words, there is no standard to integrate the two systems and foster automatic communication between trading parties. This manual system, in turn creates operational risks with the re-keying of information into each separate system by the traders. "Once in place, WeatherML would allow straight-through processing and the users could connect components within their overall architecture to each other in a seamless manner,"…[12]

XML development organizations: While XML languages can be created without enormous effort in most cases, they only become broadly useful if they are widely implemented. Since pervasively implementing an XML language represents a substantial commitment by a document creator, a new language therefore needs to have credible support from the outset in order to be successful. In order for that to happen, potential implementers will look to whether the language will be maintained (and ideally promoted) over the long term by a credible organization. The result has been the development of a multi-tiered ecosystem of standards consortia. The following is a selection of currently active consortia, grouped by the role that they play in supporting both the XML ecosystem and those that are dependent upon it.

Foundational XML organizations: At the top of the stack are two principle organizations: the World Wide Web Consortium, which developed and maintains the core XML standard and related broadly applicable standards and supporting materials, and OASIS, the consortium that has been most active in developing a broad spectrum of standards, guidelines, profiles and tools based upon XML to facilitate eCommerce across multiple business domains.

- ✓ ***World Wide Web Consortium (W3C):*** As already noted, the W3C was formed to ensure that the Web evolved in a more orderly and standardized fashion. While its initial efforts focused on HTML, its work program expanded into a variety of areas supporting the Web, including XML, Web Design and Applications, Web Architecture, Web Services, Web Devices, and Browsers and Authoring Tools. It has also dedicated significant resources (and a great

[12] McEachern, Christina, A New XML-based Standard for Weather Derivatives Transactions Proposed, Wall Street Technology, December 22, 2000, at: http://www.wallstreetandtech.com/technology-risk-management/showArticle.jhtml;jsessionid=XFR3LRZCN1XU3QE1GHRSKH4ATMY32JVN?articleID=14704626&_requestid=629891

deal of missionary effort by Tim Berners-Lee) to develop and promote a new set of standards directed at enabling a new and richer layer of meaning to Web-hosted information.

The goal of that effort is to make possible the evolution of a “Semantic Web” of information, using a more sophisticated set of tags and tools to invest Web hosted information with more pervasive machine-readable information. As with XML, a Semantic Web document would identify data with additional attributes of various types, such as geographic location, business type (e.g., a theatre), and more (such as the theatre's hours of operation), so that users of the Web could perform far more sophisticated and useful searches (e.g., “find any Chuck E. Cheese restaurant in city X within three blocks of a theatre showing Willy Wonka and the Chocolate Factory between 1 and 5 this afternoon”).

As of this writing, W3C a total of 143 standards, guidelines and other deliverables (including serial versions of the same material) have been adopted by W3C members as formal Recommendations, with many more under development.[13]

- ✓ ***Organization for the Advancement of Structured Information Standards (OASIS):*** Formed in 1993 as SGML Open, OASIS was initially created to promote SGML rather than to undertake technical activities of its own. With the advent of XML, OASIS changed its name, and also began to undertake XML-based standards development efforts of its own. Unlike consortia that remain focused on a single standard and a limited work program, however, OASIS adopted a more open, "Big Tent" technical process philosophy that allows a small number of members to launch a working group effort within the broad perimeter of the overall OASIS mission. As a result, by 2004, there were 70 working groups in operation, and the number of activities in process at any time has remained high.

  Broadly stated, as W3C is to foundational XML tools, so OASIS is to developing XML languages and related tools to address specific domain needs. Its reputation in this area was augmented soon after the launch of XML when UN/CEFACT, a United Nations committee concerned with business standards, selected OASIS as its partner to develop XML-based standards to serve the evolving needs of eCommerce. The result was the eXtensible Business Language, or eBXML, which later became ISO 15000.

  Since then, OASIS has developed a wide variety of XML-related specifications serving the needs of commerce over the Internet, both broad and narrow. Examples of the broad variety include:

  - o Universal Business Language (UBL): UBL comprises a library of useful, standard electronic forms such as purchase orders and invoices that can be easily integrated with existing software without additional data entry

[13] The main information page for the W3C can be found at: http://www.w3.org/Consortium/ The main standards page can be found at: http://www.w3.org/standards/

    - o <u>Security Assertion Markup Language (SAML)</u>: SAML allows the practical exchange of authentication and authorization information for the benefit of a Web site user and a Web host through the use of a third party identity services provider, thereby enabling not only more secure use of the Internet, but also a "single sign on" convenience for the user.

More recently, OASIS has developed XML-based standards for a variety of other purposes and constituencies, such as the Common Alerting Protocol, to facilitate the rapid dissemination of emergency warnings, standards to enable the Smart Grid, and an end to end suite of standards and processes to facilitate electronic voting, including the Election Markup Language.

As of this writing, OASIS supports 83 adopted standards (again, including multiple versions of the same standard).[14]

Single Focus organizations: Many consortia have been formed to develop XML languages and related tools for use in a single industry sector, or to meet a specific need of businesses generally. The following is a very small, but representative sampling:

XBRL®
eXtensible Business Reporting Language

- ✓ ***International Press Telecommunications Council (IPTC):*** While the IPTC is heavily invested in the development and support of XML based standards, its standards development activities predate the existence of XML by two decades. Given its mission of developing standards for the interchange of news data, it is not surprising that XML-related efforts now represent the backbone of its efforts. Today, the IPTC supports a suite of XML-based languages, each tailored to the needs of working with a specific type of news data. Those standards include:

    - o <u>NewsML-G2</u>: An XML-based general purpose exchange standard able to deal with news of any kind and media of any type

    - o <u>EventsML-G2</u>: An XML-based standard for exchanging event-related data in a manner conducive to news reporting

    - o <u>SportsML</u>: An XML-based format tailored to sports statistics and other information

- ✓ ***HR-XML Consortium:*** HR-XML, as its name suggests, was created to develop and promote standards for use by human resource professionals across all industries, based on XML. More specifically, its mission is to develop a "standard suite of XML specifications to enable e-business and the automation of human resources-related data exchanges."

[14] Adopted <u>OASIS standards</u> can be found here: <u>http://www.oasis-open.org/specs/</u> OASIS also sponsors the <u>Cover Pages</u> Web site, the most exhaustive resource on the Internet relating to all things XML. The Cover Pages site can be found at: http://xml.coverpages.org/

- ✓ ***XBRL International, Inc. (XII):*** XII develops and maintains the eXtensible Business Reporting Language (XBRL) specification for global use in financial reporting, utilizing multiple W3C standards (i.e., XML Schema, XLink, XPath and Namespaces) in order to permit the degree of highly structured presentation of financial data that financial reporting requires. XBRL utilizes metadata included in taxonomies that define reporting concepts and interrelationships between concepts and semantic meaning. XBRLS, a simplified application profile of XBRL, allows non-XBRL experts to create XBRL metadata and reports. In order to permit both localization as well as uniform reporting tools for multinational corporations, XII is based upon a national membership model, with each "jurisdictional" member in turn having domestic members. Representatives of these members may in turn participate in XII Working Groups.[15]

XML-adjunct organizations: XML provides a useful mechanism that many consortia use in connection with some of their standards activities. While less obviously identifiable as "XML consortia," they nevertheless provide XML languages and other tools of importance to the marketplace, either in specialized areas, or more broadly, as befits their overall mission.

- ✓ ***Association for Cooperative Operations, Research and Development (ACORD):*** Unlike the other organizations discussed in this article, ACORD is an American National Standards Institute (ANSI) accredited standards development organization. While it does include technology companies as members, its core constituencies are insurance underwriters, brokers and other commercial enterprises in the insurance sector. As a result, rather than creating XML languages to be used by others to create products, it uses XML to create useful forms, frameworks, and guidelines that are directly usable by its members in their businesses, or which can be incorporated into their own internally generated tools.

- ✓ ***Internet Engineering Task Force (IETF):*** The IETF is one of the foundational consortia enabling the Internet, transitioning in 1991 from a government project to one with public participation. Famous for its "rough consensus and running code" philosophy, it has been well suited to developing and maintaining some of the core standards upon which the Internet is based, including the Transmission Control Protocol and the Internet Protocol (TCP/IP). When Web site syndication (i.e., the ability to be notified when new material is posted at a given Web page) gained in popularity, the IETF chartered a new XML-based activity to improve upon the RSS syndication feed, called the Atom Publishing Format and Protocol (AtomPub) Working Group.

- ✓ ***Open GeoSpatial Consortium (OGC):*** For the last fifteen years, OGC has served as the primary venue for standards activity addressing the rapidly evolving needs of government, defense, agriculture, science and many other domains to work efficiently with geospatial information. Its 28 (to date)

[15] Links to XBRL International taxonomies, specifications and best practices documents can be found on the left side of the XII home page, at: http://www.xbrl.org/Home/

adopted standards include many that are based upon XML, including the following:[16]

- Geography Markup Language Encoding Standard (GML): A grammar (schema and instance document) for expressing geographic features

- City Geography Markup Language (GML): A schema of OGC's GML adapted for 3D city and landscape models

- Geospatial eXtensible Access Control Markup Language (GeoXACML): An extension to the OASIS eXtensible Access Control Language (XACML) allowing the incorporation of spatial data types and spatial authorization decision functions.

- Keyhole Markup Language (KML): Based upon a contribution from Google, this standard is intended to standardize the use of geospatial data in on-line 2D maps and 3D earth browsers.

- Sensor Model Language Encoding Standard (SensorML): SensorML specifies models and XML encoding to permit the geometric, dynamic, and observational characteristics of sensors and sensor systems to be defined, from simple visual thermometers to complex electron microscopes and earth observing satellites.

Commercial languages: While XML languages are usually created by non-profit membership organizations, they can also be created and sold by commercial enterprises. A rather exotic example is the Spacecraft Markup Language, developed by SRA International, Inc., for the aerospace industry.

Other: The ease with which XML languages can be created and the appeal that such an exercise has to specialists with programming skills as well as professional software developers has resulted in an astonishing array of efforts, some of which have been transitory and others sustaining. The following is a very short excerpt from an impressively long list compiled, but not recently updated, by Robin Cover at the CoverPages Website. The examples below are meant to suggest the breadth, rather than (in some cases) the depth, of the efforts listed:

Chemical Markup Language
XML Common Biometric Format (XCBF)
Signed Document Markup Language (SDML)
Real Estate Transaction Markup Language (RETML)
Emergency Data Exchange Language (EDXL)
Mathematical Markup Language (MathML)
vCARD in XML and RDF (Electronic Business Card)
Historical Event Markup and Linking (HEML)
Telecommunications Markup Language (tML)

[16] The main standards page for OGC can be found here: http://www.opengeospatial.org/standards/tml

Robotic Markup Language (RoboML)
Physics Markup Language (PhysicsML)
Exploration and Mining Markup Language (XMML)
Navigation Markup Language (NVML)
Astronomical Markup Language
AdMarkup XML DTD for Classified Advertising
Printing Industry Markup Language (PrintML)
Tutorial Markup Language (TML)
SpeechML
Architecture Description Markup Language (ADML)
Theological Markup Language (ThML)
OpenText.org Papyrus Encoding Markup
LitML: A Liturgical Markup Language
FlowML: A Format for Virtual Orchestras
Staffing Industry Data Exchange Standards (SIDES)
Electronic Thesis and Dissertation Markup Language (ETD-ML)
Steel Markup Language (SML)
Marine Trading Markup Language (MTML)
Chess Markup Language (ChessML)
Mind Reading Markup Language (MRML)[17]

III The Future

As XML co-editor Tim Bray observed in the W3C press release celebrating the tenth anniversary of the adoption of XML, XML will not be the last platform-independent, vendor neutral standard that will be needed to manage the ever-expanding flood of data that we continue to create. New standards will be required to make better and more efficient use of data on the Web, and perhaps more sophisticated standards will be needed to manage the data itself as the volume and nature of that information changes.

But given the enormous amount of information that is already exposed to the Web, it will be far more difficult to implement new standards than it was with XML, when the Web was still young, and material was still being prepared for on line accessibility for the first time. To some extent, this challenge may be ameliorated by automatic tagging tools. But experience to date with the W3C's long campaign to inspire implementation of its Semantic Web standards demonstrates that the benefits of new tagging or other systems will need to be very demonstrable before broad implementation can be expected.

That being the case, the incredible success of XML becomes meaningful in a new way: as the standard we have, we need to commit to use it most effectively. The success of XML also makes obvious the enormous benefits that properly conceived and executed information standards can bring. Hopefully, this will provide the incentive to make the substantial investments that may be needed to retrofit the

[17] The complete (and seemingly endless) CoverPages list of XML Applications and Industry Initiatives can be found at the CoverPages Web site at: A more current, but much less entertaining, list of XML languages can be found at the Wikipedia, at: http://en.wikipedia.org/wiki/List_of_XML_markup_languages

Internet and the Web with the new standards that will inevitably follow, and that will further enrich our experience of all that the Digital Age has to offer.

Sign up for a free subscription to ***Standards Today*** at

At http://www.consortiuminfo.org/subscribe/2.php?addentry=1

INTERVIEW:

XML Past, Present and Future: An Interview with Tim Bray

Andrew Updegrove

There is essentially no computer in the world, desktop, hand-held, or back-room, that doesn't process XML sometimes…XML won't be the last neutral information-wrapping system; but as the first, it's done very well. - *Tim Bray, W3C.org press release, "XML is 10!" – 2-12-08*

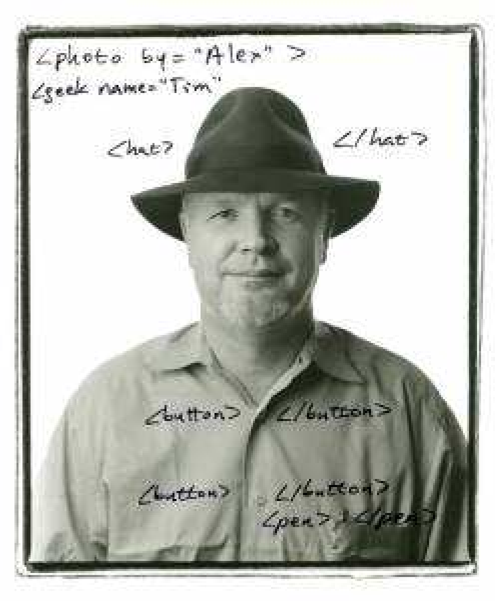


Photo by Alex Waterhouse-Hayward
alexwaterhousehayward.com

It may seem as if standards materialize out of nowhere, but of course that's never been the case. They are the product of a collaborative process that typically includes many experts, drawn from a variety of backgrounds. Every successful standards development effort also requires a few individuals willing to play a more central role, as working group chairs, to keep things moving efficiently, fairly and on the right course, and also as editors to control and write the text of the standard itself.

Serving as a standards editor is a highly technical task that can only be learned in the breach (there are no courses that teach it). At the same time, it requires not only satisfying the demands of those entitled to vote on whether to accept or reject a final draft, but also laying out what must be done in such a way that those with no prior contact can easily produce uniformly compliant implementations. Needless to say, the likelihood of a standard's becoming widely and successfully implemented in the marketplace can be greatly influenced (for better or for worse) by the skills of its editor.

In the old days of standards development, editing a standard was a leisurely process. That all changed as the pace of innovation ramped up exponentially in the information technology sector. In the late 1990s, even greater pressure was brought to bear to generate the standards needed to keep the accelerating locomotive of the Internet and the Web on track. In the fall of 1996, what must have been a new record was set when the first draft of an important new standard was produced in only twenty weeks.

Two co-editors made that possible (later there were three), one of whom was a Canadian raised in Lebanon who had already helped create one of the first successful Internet search engines. His name was Tim Bray, and the standard he helped create became one of the most influential standards of the Digital Age to date: the Extensible Markup Language, more commonly known simply as XML.

Tim's recruitment for that role was a combination of availability and capability. At the time what became the XML Working Group was chartered, Bray was an Invited Expert with the World Wide Web Consortium (W3C) and a friend of Jon Bosak, the project leader. He was also working as an independent consultant, making him the master of his own schedule. On the capability front, he had previously been the manager of a major text digitization project: the conversion of the Oxford English Dictionary. The rest, as they say, is history (you can read Tim's own highly personal account of the people, the times and the process here).

While Tim's primary role is as a technologist (his ongoing research is described at his Concur.Next Web site) he has continued to drive important Web-relevant standards efforts, including as a member of the W3C Technical Architecture Group (2001-2004), as co-editor of the Namespaces in XML W3C standard (1996-1999), and as the Co-Chair of the Atompub Working Group of the Internet Engineering Task Force (IETF) (2004-2007). When he's not editing standards, he serves as Distinguished Engineer and Director of Web Technologies at Sun Microsystems, Inc.

In this interview, Tim shares his thoughts on where XML has been, where it is now, and where it's going next.

I. The Past

AU: *For starters, what would the Internet and the Web look like today if XML had never been created?*

TB: I really have nothing beyond wild guesses; alternate histories are hard to make believable. On the downside, there is a huge amount of application integration with real business value-adds that would have been more difficult or impossible. In particular, the rise of REST [*ed:* Representational State Transfer, a distributed software architecture useful for the Web] might have been hampered if there hadn't been such a useful general-purpose format to ship around in the bodies of resource representations. On the upside, the huge waste of energy and investment that went into the failed WS-* project, which was originally presented as "XML Web Services", might have been prevented.

AU: *While XML was an outgrowth of SGML, it was a new start, rather than a new version of SGML. What were the problems you were trying to solve when you helped create XML that required a fresh start?*

TB: When you ship a new version of something, it's usually grown, compared to the previous version. XML, on the other hand, was radically smaller than SGML, so it could hardly be presented as a new version. Also, several of us were impressed with Tim Berners-Lee's then-new Web consortium and thought it might be a more fruitful place to get work done than the ISO SGML committee.

AU: *All standards need sponsors, usually from the business world. Who kicked the XML effort off, and why?*

TB: XML had some sponsorship from Sun, in that John Bosak's manager authorized him spending half his time on it. But the primary business-world backer was Microsoft, which saw the opportunity to do more business computing on the Web; it was pretty obvious that neither HTML or SGML was the right vehicle for this, but the space between them was crying out to be filled. Aside from Sun and Microsoft, there was support from some small SGML-community players, but none of the big corporates; until it became obvious that XML was catching on.

AU: *XML was created in the middle of a wild ride (the Internet Bubble) with enormous financial, technical and social dimensions that researchers will be studying for decades. What was it like working on XML – under tremendous time pressure – while all of this was going on?*

TB: It was about as much fun as you can get paid for; which reminds me that I should point out that a lot of the labor was volunteer: myself and James Clark at least. While Michael Sperberg-McQueen was employed, his employer, I suspect, didn't realize they were supporting the creation of what became XML.

[Editing XML] was about as much fun as you can get paid for

The XML Working Group was a like-minded bunch and we had a mostly-shared vision, based on experience, of what needed to be built. We got along well and were blessedly free of problem personalities. Jon was a capable and efficient leader.

AU: *XML achieved wide use very quickly. Was there more to this than simply the rapid growth of the Web? In other words, why did XML take off so rapidly, while many other worthwhile standards don't?*

TB: Let me turn that question around: Why on earth did it take until the late Nineties before someone cooked up a neutral data interchange format? There had been some attempts, most notably ASN.1. The time was long-overdue and the need was huge. XML, seen in the rearview mirror, is far from perfect, but it could be made to work for interchanging more or less anything between more or less any two computing systems, and the world really needed one of those.

Another important reason is that in parallel with designing XML, we (James Clark, myself, people at Microsoft) were building open-source software to process it. So by the time people got around to looking at it, there was already reasonably-good free software that you could put to use right then.

I guess I shouldn't underestimate the importance of the fact that XML got internationalization right via its tight coupling to Unicode in a way that turned out to pretty well just work.

Finally, the fact that XML was quite useful for encoding documents, not just relational records or persisted objects, was a major value-add.

When we (chiefly Jon Bosak and I) went out on the road to sell XML, it was like hurling your weight against a door that wasn't even latched; everyone said "Oh yeah, we can use that."

AU: *The flip side of success for a standard is often contentiousness in its development and maintenance. You had a taste of that early on when you went to work for Netscape, resulting in a third co-editor (from Microsoft) being appointed. Ten years later, we had the ODF-OOXML saga. Do you think XML bears any lasting scars from the level of energy that major vendors put into its development and maintenance?*

TB: Netscape, despite the fact that they hired me, never put any energy in. My reports on its progress were more or less completely ignored. Netscape was already well into the progress of its case of terminal arrogance.

But the answer to your question is "yes." XML 1.0 itself was designed and shipped by a small group of experts who really had no motives aside from making it work. Once it became successful, the space around all the standards-building tables became crowded with company representatives, who had neither the same level of technical expertise, nor the same focus on doing the right thing. Examples of negative results were the low quality of specifications like XSD and WSDL; and, as you point out, the OOXML debacle.

AU: *Conversely, were there any benefits from this level of attention to help off set the frustrations?*

TB: Not that I'm aware of.

AU: *How did you expect XML to be used, and by whom? Is that what actually happened, or did it take on a different life of its own?*

TB: Our primary objective was that Web servers deliver payloads suitable for processing by computer programs, as well as display to humans. We also knew that most of what was being done with SGML could be done much more easily and cheaply.

Obviously, it took on not one but a hundred different lives of its own, many of them still frankly astonishing to me. I've noticed that people who help build general-purpose technologies are usually bad at predicting how they'll be used.

AU: *While XML is known as a remarkably flexible standard, every standard inevitably includes constraints. If you had it to do over again, are there any things you would do differently based on how you've seen technology and usage develop?*

TB: Oh, yes. The big thing we'd do is leave out DTDs. In the real world of data interchange and processing, schemas are second-class citizens. Also, DTDs brought along with them a bunch of features which turn out to be less than useful or arguably even actively harmful.

Also, XML Namespaces, which were done at more or less the same time as XML 1.0, get a lot of hate. There are a couple of pieces of that design that could be improved, and there’s also a case to be made that they actually could have been dispensed with.

There are a bunch of other pieces of fine-tuning that we can see in the rear-view mirror, but those are the big ones.

AU: *How might the Internet and the Web look and function differently today if you’d gone down that road instead?*

TB: Not much, to be honest. We’re fortunate in that it was possible, in practical terms, to either ignore or work around the irritants in XML.

AU: *Fundamental standards often influence many other standards decisions, and also how architectures evolve in a broader sense. Do you see such wider effects on how the architecture and/or infrastructure of the Internet and Web have evolved that you can trace back to the creation and success of XML?*

TB: First, XML is an existence proof of the possibility of data-interchange formats that are language-neutral, OS-neutral, database-neutral, and so on. We’ve had a couple more since XML that have gotten some traction: YAML [*ed.:* a much more human readable data format] and especially JSON [*ed.:* Javascript Object Notation, a data exchange alternative to XML, often used in Ajax programming]. I particularly like JSON for the things it’s good at, which pleasantly enough mostly happen to be things where using XML is awkward.

Why on earth did it take until the late ‘Nineties before someone cooked up a neutral data interchange format?

The notions of Web Services and especially REST depend crucially on the assumption that you can ship things around the infrastructure that any flavor of computing infrastructure can produce and any other flavor can consume.

These days, any time there’s an argument as to whether some information resource should be open or not, it is a pure policy argument; because of XML, there are typically only minor *technical* barriers to opening up information. That seems like the big deal to me.

II The Present

AU: *XML has been adapted to handle everything from sports information, to advertising handling, to human resource data, to financial reporting information. Where do you think its impact has been greatest?*

TB: The most successful application, in terms of volume of information and number of users, has been syndication: Atom and RSS. Also, offerings like Amazon Web Services depend crucially on XML. But look behind the firewall at any large enterprise, private or public sector, and you’re apt to find a whole bunch of XML sloshing back and forth being used to stitch different applications and components together; in many cases even when they weren’t designed for such integration.

AU: *Are there any areas where you're surprised that XML isn't yet being used to its full potential? Which, and why do you suppose that is?*

TB: Some of us hoped that XML would replace a lot of the usage of HTML on the Web, simply because dealing with real-world HTML is such a major pain in the butt. That hasn't happened, simply because the cost of HTML parsing is already a sunk investment, and so there was no real upstream pressure to produce XML.

XHTML has been a success and quite a few of the better Web designers use it just because that eliminates certain classes of problems you can run into. But now we see that the HTML5 project is moving in quite a different direction; its leadership is actively disdainful of XML.

The real answer to your question, though, is that XML is being used far beyond what any of us could have dreamed its full potential to be.

AU: *We both remember the ODF – OOXML competition well. If that process revealed any flaws in the standard setting infrastructure and process, what were they?*

TB: This was my first exposure to the ISO/IEC JTC1 process and culture, and I was horrified at the pervasive corruption and incompetence. I would prefer never to work in that context again. I would be eager to participate in a reform effort, if there were the political will to launch such a thing.

AU: *Do you think the existing IT standards development structure (e.g., the ISO/IEC process plus innumerable consortia) is sufficient for today's demands, or do you think we need new types of organizations, such as ones that would rate the "openness" of standards developers?*

TB: I have experience in the W3C, IETF, ISO, and OASIS contexts. Among those organizations, I find I generally prefer the IETF culture and process. Having said that, standards are created by people, and the individuals who end up as committee members, editors, and chairs end up having a huge influence.

I'm unconvinced that the world needs any new standards organizations.

AU: *The Obama administration in the US has pledged to spend tens of billions of dollars on several major technology based initiatives that involve masses of data – a major electronic health record initiative, as well as a total redesign of the electronic power grid that is intended to turn it into an interactive ecommerce platform. Similar efforts of varying size are in progress in other countries and regions. What role do you see XML playing in these enormously expensive undertakings? Will they require further development of XML?*

TB: Almost by definition, a high proportion of this information, especially in the health sector, takes the form of documents. If you want to represent documents in a form that's open, highly interchangeable, re-usable for unforeseen purposes, and free from vendor clutches, you really can't beat XML. So I'm assuming that it will be the default choice for a lot of this stuff.

On the other hand, some of the work, for example in "Smart Grid", seems to me like it involves interchanging numbers and database records rather than documents; something like JSON may be a much better fit.

Of course this doesn't mean that the costs, complexity, and openness in these projects won't be driven in the wrong direction by technology vendors and especially blue-suit consultant solution providers, whose business interests are not aligned with lightweight, open, flexible, technology deployments.

AU: *What's being done with XML 1.0 and 1.1 development today that you think people should be aware of?*

TB: Not much. XML 1.0 pretty well just works. The XML group at W3C continues to tinker with internationalization, mostly because Unicode is a moving target. I don't agree with some of the stuff they've done, but on the other hand it doesn't seem to be actively harmful. XML 1.1 was a mis-step, which fortunately has been largely ignored by the marketplace.

The big thing we'd do is leave out DTDs. In the real world of data interchange and processing, schemas are second-class citizens

III The Future

AU: *First we had XML 1.0 (in 1998), which is now in its 5th edition, then XML 1.1 (in 2004), now in its 2nd edition. Naturally, people talk about whether there should be an XML 2.0. In your view, should there be, or has XML, like SGML, reached a point where any thing significantly different should mark the launch of a new standard?*

TB: I don't think XML needs any more features. I suspect that opinion is widely shared. I have proposed something I called "XML-SW" where SW stands for Skunkworks, which is just a cleanup. See
http://markmail.org/message/hzxocbofmmmgxeah
and http://www.textuality.com/xml/xmlSW.html

It integrates three or four of the low-level XML 1.0 standards that everyone implements: namespaces, the information set, xml:base. Also it decouples DTDs. Finally, it reorganizes the XML specification to make it more readable and usable. I think the resulting document is quite a bit cleaner and more useful to implementors. But realistically, the world seems to getting along reasonably well without it.

AU: *If it's time to go to some sort of "next generation" XML, what do you think its mission should be?*

TB: I don't think it's time. The world is reasonably well-served by the XML and JSON tandem for information and document interchange. Let's invest at the higher level, in applications and data resources that impact users, not the boring stuff in the engine room.

AU: *To what extent, and how, do you see the Semantic Web as an extension of XML?*

TB: Not in the slightest. The fact that RDF has an XML syntax is an unfortunate historical accident, because XML was definitely flavor-of-the-month at the time RDF was being built. Also unfortunate because that syntax is horrid; hard to read, hard to write, hard to work with. I am well-known to be generally a Semantic-Web skeptic anyhow; there has been considerable energy and hype going into the project for a decade or so, and remarkably little useful software coming out. By the way, it seems that the Semantic Web has now been rebranded as "linked data."

AU: *Looking way out into the future - say 10 or 15 years – where do you see data creation and sharing headed? What should we be able to do in the future that we can't do now, and what role will standards have to play in order to make that possible?*

TB: That, quite properly, is a matter of policy not technology. The barriers for sharing information are not technological in any crucial way. Where there is the political will or business case for sharing information, you can start now; no need to wait for technology.

I think the most interesting thing going on in the world of information sharing is the advent of low-cost mobile-phone technology in the underdeveloped portions of the world, bringing the benefits of the Internet, albeit in a less-polished forms, to a couple of billion people who stand to realize benefits which will impact their lives more than the Internet has impacted ours.

AU: *I wrote a piece recently called Digitization and the (Vanishing) Arts of the Book. As the world moves more and more from fixed to electronic media, do you think that we need to make more room for aesthetics in standards development? If so, how would we go about that?*

TB: In response to your piece, I have to point out that electronic display media have been playing catch-up these last few decades. Paper display technology offers immensely higher resolution and a vastly larger palette of colors compared to any electronic medium, and has been more convenient to carry around and use. With things such as the Kindle, we're making progress on convenience, but I think we're still years and years from catching up on resolution and color.

Now, if you're reading my blog posts, a popular novel, or a Humanities textbook, who cares? I live in a part of the world where forestry is an environmentally fraught issue, and I have no patience with the cutting down old-growth timber to print Stephen King (and I like Stephen King). I'm a book lover, my house is stuffed with them; but in the future, the preserve of books will (properly in my opinion) be the antiquarian domain and those places where high-quality display is essential: Art, coffee-table books, graphically-intense textbooks; perhaps poetry.

Now, you asked about aesthetics in the standards domain. Engineering aesthetics are a different kind of beast; we worship at the temples of simplicity, flexibility, and minimalism. Which are only occasionally appropriate in the world of human aesthetics.

AU: *This has really been great, so just one last question: How long have you been wearing the <hat>?*

TB: A couple of decades. I've always like wearing a hat. On top of which I'm a pale white bald guy, and developed some lesions on my head that had to be blasted off with liquid nitrogen; ouch! So thus my fashion sense and medical advice are pointing in the same direction.

STANDARDS BLOG:

Smart Phones, eBook Readers, and "Same Old, Same Old"

Andrew Updegrove

Plus ca change, plus c'est la meme chose
- French Proverb

Ah yes - "The more things change, the more they stay the same." Isn't that how the old saw goes? Or, in the more impatient parlance of today, simply "Same old, same old." So perhaps it should be no surprise that the old proverb would also hold true in the rough and tumble world of standards. And that in fact is the case, not only generally, but more particularly in the suddenly hot war over eBook reader formats. This time around, though, there are a few new and interesting twists (on which more later).

It's 2009. Do you know where your Betamax is?

What's the "same old" part all about? There are two alternate behavioral flavors: (1) try and set a *de facto* standard that you control, perhaps even obtaining a near monopoly in the process (the "winner take all" strategy), and (2) pit your standard against another, where your standard gives you some relative, if not absolute, advantages (the "our team vs. their team" strategy).

In this case, it looks like Amazon is attempting to pull off the first, but in fact it's hard to tell whether they are serious, or just adopting a flawed strategy. Either way, I believe they will eventually have to admit defeat.

Here's the background, if you haven't been following the eBook contest. For years, some companies have tried to sell the public on the concept of electronic books with little success, largely because reading a book on a laptop or desktop simply isn't as appealing as reading from the old fashioned paper format. What was needed was a brand new "form factor" (as the consumer electronics industry calls it) that made sense.

Which is what eventually came along. Just as Apple's iPhone validated the concept of the Smart Phone, leading dozens of other vendors to accelerate their own development efforts to hop on the bus, Amazon's addition of wireless capabilities (and low book prices) to its Kindle eBook reader finally hit the right buttons. The Kindle sold enough units to prove that the public would indeed read eBooks if they had the right platform, and the land rush towards exploiting this new digital device opportunity was on.

Just how successful has the Kindle been? Jeff Bezos reports that in the case of titles Amazon offers in both paper and electronic form, 48 copies are now digitally shipping for every 100 that leave the loading dock in tangible form (that ominous creaking sound you hear is the sound of a venerable paradigm shifting).

There’s just one catch: if you buy a book from Amazon, you’ll only be able to read it on a Kindle (unless you can find another device that has licensed the rights to use Amazon’s proprietary formats). That’s because Amazon decided to use a proprietary format, called AZT, to display the books it sells for viewing on the Kindle.

And that, of course, is where the “same old, same old” part comes in. Did Amazon need to come up with it’s own format in order to sell books? Of course not. There have been a number of open standards, and standards sets, developed over the years, one of which (ePub, a suite of publication, packaging and container standards) is both mature and widely implemented in other devices. The Sony Reader supports ePub. So does the new Barnes & Noble Nook. Even the iPhone supports ePub. Adobe Digital Editions? Yup, even though Adobe obviously supports its own PDF formats for eBooks as well.

The ePub format is the work of the International Digital Publishing Forum

But not Amazon. Why?

Perhaps Jeff Bezos is trying to take a page from the Steve Jobs playbook. After all, the Apple App Store has been a huge success. Apple Apps are sold only by Apple, and developed only to run on Apple products. But there are a number of significant differences. Consider these, for example:

- ✓ Apple makes its money selling hardware; Amazon hits its revenue numbers by selling books.

- ✓ iTunes and Apps are bringing real money to the Apple bottom line, it's true, but their principle value is to increase the attraction of the far more profitable iPhone and Touch; the Kindle’s biggest value is as a way to lower Amazon fulfillment costs and increase the number of books each customer buys each year.

- ✓ You can buy music from anyone and play it on an iPhone, so there is no reason not to buy an iPhone even if you already own music; but for now you can only buy books from Amazon if you want to read them on a Kindle.

And there’s where the clue about Amazon's standards strategy presumably comes in: a customer will only be likely to buy one Kindle, but will hopefully buy many books, year after year, from Amazon. If that customer doesn’t want to buy multiple eBook readers (who would? They're expensive), that means that she will need to buy all of her books from Amazon. Conjoin that with Jeff Bezos’ stated desire to offer every book every written in electronic form, and there you have it: a plan to set a standards-baited trap to conquer the world of book selling. One format. One seller. One dominant vendor – Amazon as the Google of book selling.

Can Amazon pull it off?

The answer, I think, is no. And by trying to go for the gold, Bezos may throw away his early lead in eBook sales to boot. The reasons rate another short list, again using Apple for comparison:

- ✓ Apple's business model has always been to make lots of money per unit sale, pulling in fat margins on products that have become commodity items for its competitors. That means that even if Apple only gains a small percentage of the global smart phone market it can still be enormously successful – as it already has been. It can sustain this edge by out-designing and out-innovating the competition, exploiting a competence it has consistently demonstrated for many years. Amazon, on the other hand, can only sell books on eBook readers – a commodity item tailor made for a race to the bottom price war. Unless Amazon can be as successful at designing eBook readers as Apple is at designing computers of all sizes, Amazon will lose. Apple, of course, has been designing hardware and software since the dawn of the PC Age, while Amazon, well, you get the picture. Result? Amazon will only be able to compete on price, using the Kindle as a very expensive loss leader.

- ✓ eBook readers aren't the only platform on which people will read books – they are already doing so on cell phones and smart phones. There are not only formidable competitors in the smart phone marketplace (including Apple), but the phone market has other drivers and players (most notably the telecom companies) that are far larger than Amazon. On this chess board, Amazon barely rates pawn status.

- ✓ There's another little company out there called Google, and Google has launched its own smart phone operating system – it's called Android, it's taking off, and it could run an eReader, too. Google is launching its Chrome operating system soon as well. Finally, Google is well along in the process of digitizing every book in the world – and it isn't doing so out of public spirit. It's planning on you and me actually reading those books - in the Cloud - and it's expecting you to read some Google ads on your way to get there.

But wait – there's more. Remember that new twist I mentioned? As it happens, two of the most implacable enemies in the world are also uniting behind the ePub standard. I refer, of course, to the Peoples Republic of China (a/k/a, "Mainland China", to those that live in Taiwan) and Taiwan (a/k/a "Formosa" to those that live in the PRC). It seems that the two arch enemies have decided to ignore their differences when it comes to, yes, eReader standards.

That's what the two governments announced last week, and the facts of interest run deeper. For example, Prime View International, a Taiwanese company, is the contract manufacturer of the Kindle. It also recently bought E-ink Corporation, the developer and patent holder behind the displays used by just about every eReader today. Prime View and other Taiwanese manufacturers are already selling eBook readers in China. One sales partner is China Mobile – with 513.5 million subscribers.

In short, the world is moving to the ePub standard, with Amazon as, apparently, the primary holdout. That means that it will be in everyone's best interests to optimize every device (except the Kindle), and to convert every book, to the ePub standard. Amazon, on the other hand, will need to support its hardware and format standard all by itself. Of course, not being a hardware or software company, it will need to rely on…oh yes…the Taiwanese to supply them with the sort of cutting edge technology to be able to beat….Hmmm. That may be a problem.

The moral of the story, of course, is the usual one: standards are a great way to create big markets fast, for everyone's benefit, vendors and customers alike. If competitors get together and all adopt the same standard, they can then compete in other ways – cooler hardware and software features, better service offerings, and so on. The result? Everybody competes for a fair share of a much bigger pie, and customers get a broader and less expensive range of products and services to choose from.

There's another old proverb that goes like this: those that forget history are doomed to repeat it – and that holds true in commerce as well as in international relations. So I'd think twice before spending too much of your Christmas budget on a Kindle for that someone special.

It's a shame, though, that Jeff Bezos doesn't get it. And it will be an even bigger shame for Amazon's stockholders if he doesn't wise up soon.

Bookmark the Standards Blog at http://www.consortiuminfo.org/newsblog/ or set up an RSS feed at: http://www.consortiuminfo.org/rss/

Sign up for a free subscription to ***Standards Today*** at

http://www.consortiuminfo.org/subscribe/2.php?addentry=1

CONSIDER THIS:

#61 Jazz, Jazz Standards, and Open Source

Andrew Updegrove

Could Louis Armstrong have out-programmed Bill Gates?

As anthropologists now realize, a new human species began to emerge approximately thirty years ago. Its appearance was typified by careless and informal dress (T-shirts; jeans; old sneakers). Some specimens were as likely to be nocturnal as diurnal (and sometimes both). Many shared common food (pizza) and drink (highly caffeinated beverages) preferences. Their spoken language was efficient within their societies (when they chose to communicate by speaking, which wasn't often). But traditional *homo sapiens* found their dialect hard to parse. A few demonstrated a less than ideal level of interest in matters of personal hygiene.

But the members of this new evolutionary branch were uniquely well adapted to exploit several economic niches that were themselves evolving. Indeed, in some business sectors they were to us as *Cro Magnons* were to Neanderthals – and they prospered accordingly. So it was that as the last millennium came to a close, these newly emergent gods of the business world had left their peers in the dust. The immutable laws of Charles Darwin had asserted themselves once again.

I speak, of course, of computer programmers.

Alright, so hordes of people with a special knack for coding didn't suddenly mutate out of the primordial ooze that was us. In truth, they just became much more noticeable as a group when they began making a gazillion dollars on stock options in companies like Microsoft, Netscape and Sun Microsystems. But that only leads us to a different mystery: *what did all the software programmers do before there was software to program?*

Back in the early 1980s I posed that question to a friend of mine who had taken the software route, teaching himself how to program after dropping out of physics grad school. He gave me a pitying look, and stated what to him was obvious: "They were car mechanics and jazz musicians."

Well, sure, once you thought about it. And high school chemistry teachers, too, I expect. It's just that when they were spread around so widely, it was not so obvious to us, and a whole lot less remunerative for them.

All of which brings us to the point where we can consider this equally obvious analogical statement: *Standards are to classical music as open source is to jazz*.

Dig?

No question about it. Classical music, like technical standards, is complex and developed through a painstakingly exact process. Needless to say, people wouldn't pay a dime to hear a musician implement a Beethoven sonata unless she did so with precision and utter faithfulness to the score. Even "Pops" orchestras choose to "up" orchestrate popular music rather than monkey with the presentation of classical material. Instead, orchestras compete through the quality of their implementations, the ingenuity of the programs of music they assemble, and the talent of their ensembles and soloists.

The world of classical music also believes in copyright, as anyone who plays, or sings, it knows. We're not comparing classical music to consortium standards here, either – we're talking ISO. Join a chorus this holiday season, and you'll be asked to hand over a not insubstantial check to pay for your sheet music. No photocopying here, please – ASCAP is watching.

Jazz, of courses, is open source all the way – it's the ultimate freedom machine. Once you've grasped the melody line and basic chord structure of any song, you're on your own, encouraged to take the author's initial inspiration anywhere you wish. A jazz musician isn't judged by the faithfulness of his rendition but by what he codes at the musical keys.

THE NEARNESS OF YOU

A Real Fake Book

Even the legal underpinnings of jazz are different, at least in the trenches. No one who is really serious about jazz goes out and buys, say, an Oscar Peterson, Miles Davis or Mahavishnu John McLaughlin song book, setting down note for note what the great musician played. How could you? They played it different every time.

What you would do is buy a good "fake book," chock-a-block full of hundreds of jazz "standards" – songs with that certain ineffable magic that has led musicians the world over to include them in their repertoires for decades. Why a "fake" book? That's a detailed story for another time (if you can't wait, the Wikipedia page is here), but it's partly because many fake books are bootleg, *samizdat* compendia, usually hand-scored, and typically with the sort of cheap, insertable plastic spine that allows you to produce and sell it for cheap. And also because all you'll see when you open a fake book are the melody line, lyrics and chord names – after that, you've got to fake it yourself.

The fake book I picked up back around 1977 has 300 pages of the best music you could ever hope to riff on, from winsome WWI era favorites (such as "After You've

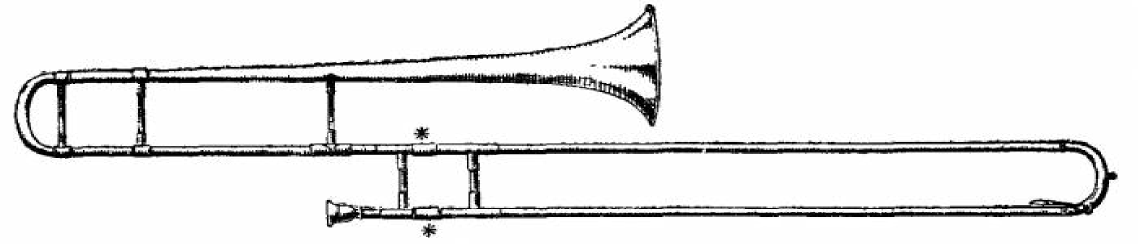
Gone") to Age of Swing crowd pleasers (e.g., "One O'Clock Jump") to more contemporary classics (like "The Girl from Ipanema" and "Take Five"). Each one is a great foundation upon which to build.

As with open source, whatever recorded magic anyone builds on top of the kernel of a melody line goes back into the pot. Anyone who wishes can incorporate their runs and flourishes into their own interpretations, each musician adding to what came before and helping further weave the entirety of the jazz experience and musical techniques into a constantly evolving continuum of inspiration.

All of which has more to do with open source software than you might think. Indeed, the jazz analogy may help you understand open source software better than more prosaic explanations. For starters, just as there are pedestrian (like me) and master (like Thelonius Monk) musicians, there are master coders that are viewed as geniuses by their programming peers. In the eyes of those skilled in the software craft, the code of a master programmer is viewed as a work of art.

And what good is a work of art if no one can see it? Small wonder, then, that so many people are willing to create open source code for the appreciation of others, while proprietary programming gets done on the clock. Nor any wonder that many of the jazz musicians of yesteryear might find the keyboard of a computer as appealing today as that of a Hammond B-3 organ of yore.

So it is we see that classical music is indeed to a standard as jazz is to open source, and that computer programmers are not so recently evolved as they may have seemed at first to be. Perhaps classical musicians will have their own turn in the technology sun some day – and the stock options to go with it. But they'll certainly have to leave the long black dresses and tails behind.



Read more *Consider This***...** entries at: http://www.consortiuminfo.org/blog/

Sign up for a free subscription to ***Standards Today*** at

http://www.consortiuminfo.org/subscribe/2.php?addentry=1