



Daniel Rivers-Moore

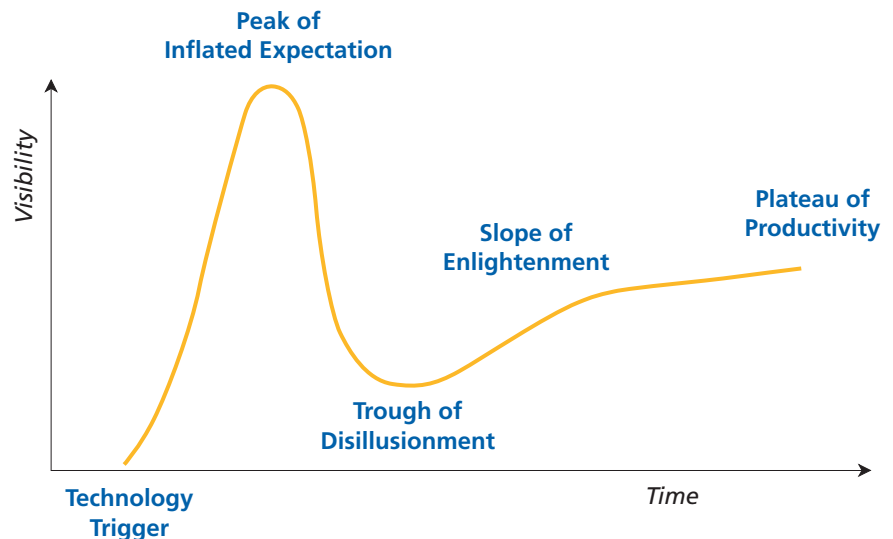
RivCom

Kicking off this series of three articles on XML technologies in broadcasting, the author offers us an introduction to the eXtensible Markup Language, which was the subject of an enthralling EBU Seminar in Geneva earlier this year (EBU members can access the proceedings at http://www.ebu.ch/ptech_xml_sm01.html).

Once in a very great while ...

Technologies come and technologies go. Often there is a great fanfare as pundits sing the praises of the next revolution that will change our lives. New buzzwords are created. Journalists describe how our lives will be changed. But soon all discussion on this topic is drowned out by the clamour surrounding the next new technology that “will change our lives”.

Of course sometimes new technologies *do* change our lives. For better or worse, we are all affected by the car, telephone, word-processor, spreadsheet and, now, e-mail.



We have a pattern here: exaggerated claim, followed by cynicism, *sometimes* followed by real change.

The Gartner Group has described this phenomenon in their *Hype Cycle* (see the diagram above), where we see the transition from the *Peak of Inflated Expectations*, through the *Trough of Disillusionment* to the *Plateau of Productivity*.

This is echoed in Geoffrey Moore’s books *Crossing the Chasm* and *Inside the Tornado*, in which he describes how many really useful technologies never become established within the mainstream, no matter how good they are. This is because none of us wants to be left with an 8-track cassette, the Betamax video or the word-processor that no one else uses. A chasm is created, which many technologies fail to cross. If the chasm can be crossed,

technologies enter the “tornado”, with almost limitless demand for the technology that is now defined as “proven”.

So how are we to respond when we see statements in the press like this:

*Once in a very great while we find ourselves on the cusp of a major shift in computing technology
Tomorrow's IT may hinge on XML today.*

(InfoWorld, July 1998)

Will XML join HTTP, TCP/IP and HTML as a ubiquitous standard for information exchange? Or are statements like this – and there are very many statements like this being made about XML – just over-blown hype?

What is XML?

First of all we need to understand what XML is.

XML (eXtensible Markup Language) is a standard developed by the W3C that was approved in February 1998. It is a simplified form of SGML, the information markup language that gave rise to HTML, and enabled the Web to exist. Just like SGML and HTML, XML provides a standard way of adding “tags” to information. These tags may not be displayed, but they provide “metadata”, or information about information, that enables applications to process content in helpful ways.

Comparison to HTML

To understand XML, let's consider both its similarities to, and differences from, HTML.

HTML provides a (theoretically) *fixed* set of tags that are primarily used to determine how information should be *presented* and *linked*. Here is some HTML:

```
<H1 ALIGN="CENTER">Captain Corelli&#146;s Mandolin</H1>
<P ALIGN="CENTER"><I>Louis de Berni&egrave;res</I></P>
<P ALIGN="RIGHT"><FONT SIZE="-1">&copy; 1994 by <A HREF=mailto:ldb@lb.com>Louis
de Berni&egrave;res</A> </FONT></P>
<HR>
<P ALIGN="CENTER">To my mother and father</P>
<HR>
<H2>Dr Iannis Commences His History and is Frustrated </H2>
<P>Dr Iannis had enjoyed a satisfactory day... </P>
```

Here we see the use of standard HTML tags such as <H1>, <H2>, <P> and <HR>. We see that some of the tags include attributes such as ALIGN="CENTER". All of these tags are used to determine how the information will be displayed within a Web browser. In addition, we see the use of the <A> tag together with the HREF=mailto attribute to provide some behaviour – in this case to launch an e-mail addressed to ldb@lb.com, when the author's name is selected. We also see that, with HTML, special calls need to be made to the higher order characters such as the “è” in “Louis de Bernières”.

When I saved the text above as a text file with a .htm extension, and opened this with my browser, what I saw is given in the browser screen-shot shown on the next page.

Now let's look at the same text, tagged with XML:

```

<book copyright_date="1994" copyright_owner="author">
  <book_title>Captain Corelli's Mandolin </book_title>
  <author>Louis de Bernières</author>
  <dedication>To my mother and father</dedication>
  <chapter>
    <chapter_title>Dr Iannis Commences His History and is
    Frustrated</chapter_title>
    <para>Dr Iannis had enjoyed a satisfactory day...</para>
  </chapter>
</book>

```

We can immediately see some similarities between XML and HTML – the use of angle brackets, start tags matched by end tags (with a /), some tags containing additional attributes (`copyright_date` and `copyright_owner` within the `<book>` element in the XML example). We can also see that XML, which uses the full Unicode character set, can cope with the higher order characters such as “è”.

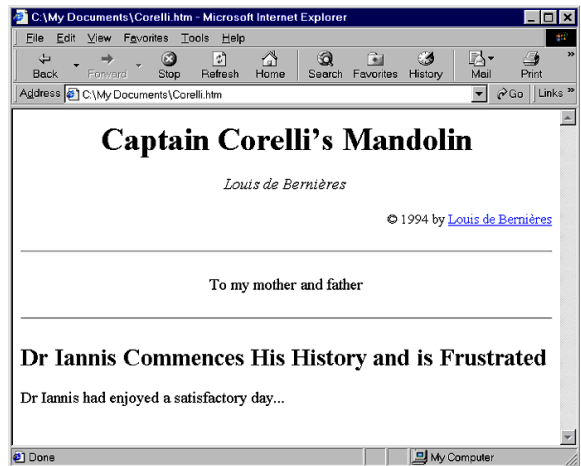
However there are two major and fundamental differences between XML and HTML – one of which is obvious from the examples above, and one of which may not be immediately apparent:

- **XML separates content from presentation**

You can see that the XML tags (`<book_title>`, `<author>`, `<dedication>`) describe structure and meaning, but not display, whereas the HTML tags (`<P ALIGN="CENTER">`, `<I>` (meaning italic) and ``) are all about display and contain no information about structure and meaning. With XML you determine how information will be presented in a separate “stylesheet” which will say how information within different tags should be displayed in different contexts.

- **XML is not a set of tags, but a set of rules for how you define your own tag sets**

It may not be apparent from the example above, but I invented the XML tags that I’m using here. While HTML provides a pre-defined set of tags, XML provides a set of rules for **determining your own tags**, relevant to the information you want to convey. Hence *Extensible* Markup Language.



So simple?

Can something as simple as a set of rules for determining markup tags really cause analysts such as Forrester Research, Cap Ventures and the Gartner Group to produce reports supporting the view that XML will become established as a key enabler for Internet-based enterprise information exchange and computing?

SGML — the unsung hero

HTML and XML are both closely related to SGML, the 11-year old ISO standard that has primarily been used for large-scale documentation projects. HTML is simply a defined set of SGML tags. XML is a sub-set of SGML with a few of the more complex aspects stripped out to make it easier to use over the web.

SGML is the unsung hero, which has spawned two famous children.

Can a technology, that is substantially similar to an 11-year old standard that few have adopted, lead Microsoft to say: *“XML is the universal format for data on the web”*; Netscape to say *“we are 100% committed to XML”*; SAP to say it has *“embraced XML across its business framework as a format for data interchange”*; IBM to say *“XML is a key open standard which will help our customers and business partners realize the promise of e-business”*; Sun, whose John Bosak pioneered the development of XML, to say *“XML has been heralded as the next important internet technology, the next step following HTML, and the natural and worthy companion to the Java™ programming language”*; Oracle to say *“XML ... will be a key enabler for e-business. Adoption of XML will simplify information exchange between applications and provide great flexibility in how information can be presented and used by end users”*?

(Incidentally, when RivCom ran its first company seminar on XML a year and a half ago, we said *“XML is arguably the most significant development in computing since the advent of the Worldwide Web”*. They thought we were crazy. Now it seems that at least we are not alone in the asylum!)

Well, at its heart, XML is that simple. But there is a whole family of standards being built around the core XML standard which enable complex applications to be built (see *Appendix A*). We can think of XML being like the wheel, a simple but powerful technology. A lot can be achieved with the simple wheel but, in today’s complex world, we add lots of complexity to it through the addition of cogs, gears and cams to enable the engineering of highly complex applications. It is the same with XML – a lot can be achieved with just XML, but the most complex applications cannot be built with XML alone.

It is the status of these supporting standards that presents a major stumbling block for those trying to ascertain the value and potential impact of XML. The problem is that people use the term ‘XML’ interchangeably to describe both the core XML standard itself and to describe the whole family of XML standards. When claims are made for the prowess of XML, it is often unclear whether the assertion is really about the core standard, a companion standard or an application built on a number of XML-related technologies.

The same problem occurs when people express opinions about the readiness and maturity of XML. The core XML standard is not bleeding-edge technology. It is a stable standard, approved in December 1997 and closely linked to SGML, which has been used in the aerospace, heavy-manufacturing and pharmaceutical industries since the early 1990s. However, the supporting standards are at different stages of development and maturity. So care needs to be taken when considering how and when to apply more complex XML-based technologies.

What are the benefits?

When I compared XML and HTML above, I identified two key differences:

- XML separates content from presentation
- XML is not a set of tags, but a set of rules for how you define your own tag sets.

There are considerable benefits that arise from these distinctions.

Putting on the style

Many documents can be linked to a single stylesheet. Changes in design can then be implemented in a whole family of documents through a single change in the stylesheet.

Stylesheets can also be used to enable multiple views of the same information. This can be used to allow subscribers to see the whole document, while non-subscribers can only see the abstract; or to allow managers to be provided with an overview, while the engineer sees all the technical detail. This approach can also give the user much more choice about how they view information.

Not just for browsers – not just for humans

Separating format from content enables XML messages to be exchanged between different systems. The same XML document can have different formatting applied so that it can be processed by a browser or different types of printer, or be converted to any output format that is required. XML is being used extensively as a standard format for information exchange between systems without ever being displayed for human consumption.

Not just the Web

While HTML is purely a Web application, XML can be sent over any network.

Documents and data

XML will complement HTML as a publishing technology for documents. But it is at least as important as a generic standard for data exchange. In truth, the distinctions between documents and data are increasingly becoming blurred, as XML documents can contain XML data fragments.

Documents as applications

The powerful processing language that is provided by XSLT (see *Appendix A*), and potential action languages that will follow, will enable the generation of a whole new level of interactivity and processing within “documents”. Documents are becoming applications in their own right.

Robust – long-life information

Because XML documents and data are not tied to any proprietary software tool, they can have a much longer life-cycle than the software systems built to process them.

Effective searching

When undertaking searches with XML information, it is possible to be far more specific than with HTML, thereby reducing the number of redundant “hits”. With XML we can differentiate between books *by* Winston Churchill and books *about* Winston Churchill in our searches. Or find out about *Queen Elizabeth II*, the boat, without getting thousands of references to the current British monarch.

Cheap software tools

One of the design principles adopted by the group which is developing the XML standard was that it should be easy to build software tools to process XML. This, combined with the massive adoption of XML by software vendors, means that XML tools are coming on to the market which are much cheaper than their SGML counterparts. Indeed there are many excellent XML tools downloadable over the web (see <http://xml.apache.org/> for a repository of free XML software, or visit sites such as <http://www.ibm.com/xml>).

Abbreviations

EDI	Electronic data interchange	TCP/IP	Transmission control protocol / Internet protocol
HTML	Hypertext markup language	W3C	World Wide Web Consortium
HTTP	Hypertext transfer protocol	XML	Extensible markup language
SGML	Standard generalized markup language		

All these terms, and many more, are defined at:

<http://whatis.techtarget.com/>

Vocabularies for all

The ability to define your own tags makes it possible to code information that is appropriate for you. For company-wide information it becomes possible to develop tags relevant only to one specific business; life gets more interesting when information needs to be exchanged across the supply chain. Here it becomes helpful to develop shared vocabularies relevant to a complete business sector. There are many initiatives underway to develop DTDs, schemas and vocabularies relevant to complete communities, from healthcare to transport, from engineering to the voluntary sector. OASIS's XML.org and Microsoft's Biztalk are providing repositories for these shared applications of XML.

And graphics, maths, mobile phones, voice recognition – the list goes on and on

In addition to vocabularies for different business sectors, a vast range of XML-based standards are being developed to meet specific computing needs:

- SVG (Scaleable Vector Graphics) – a language for describing two-dimensional graphics in XML;
- MathML (Mathematical Markup Language) – an XML application for mathematical and scientific content;
- WAP (Wireless Application Protocol) and WML (Wireless Markup Language) – XML-based technologies for allowing Internet access by mobile phones and other hand-held devices;
- VoXML – voice recognition markup language;
- ... and many more.

XML representations of existing standards

As XML is becoming *the* standard interchange format, many initiatives are underway to map existing formats to XML. For example, initiatives are underway to harmonize the STEP standard with XML, and XML representations of the modelling languages UML and Express are being developed.

XML and EDI

Initiatives such as the European Union XML/EDI Pilot Project demonstrated how Electronic Data Interchange (EDI) based on XML standards can extend electronic-commerce to small and midsize companies that cannot afford to implement systems based solely on traditional EDI standards such as EDIFACT. Following on from these pilot projects, OASIS and UN/CEFACT – the United Nations organization responsible for the EDIFACT standards, have joined together in the ebXML (electronic business XML) project to define a set of standards and protocols, all based on XML, that together provide a standardized infrastructure for carrying out electronic business. The ebXML suite of specifications was approved in May 2001, and has been successfully piloted by a number of companies. The coming months will see early adopters beginning to use it to do real business with their trading partners.

Consumer Relationship Management

XML is being adopted as a core technology within the field of Consumer Relationship Management, where integration and data exchange are key barriers to effective Internet-based operations.

Distributed applications

XML is not only being seen as a core technology to enable interoperability between applications, but also as a key to enabling data exchange within the latest generation of distributed Web-based applications.

What about the costs?

It can be seen that XML's growing importance and hype comes from its potential for computing as a whole, not just as an alternative standard for presenting documents. There is a wide range of potential, and emerging applications, for: Web-based e-commerce; enterprise content management; the brokering of content and real products; information discovery; workflow applications; application integration, and inter-application and inter-server communications.

What are the costs of this unprecedented flexibility of information use? I think that there are potentially three key costs:

- **There is an increased requirement for clarity in the definition of the meaning of the information being exchanged**
XML lets us code the information for its meaning. To reap the full benefits of this, we need to be clear about what we mean, which entails the need to develop effective models of our business; models of the information exchange requirements, and models of the system–system and system–user interfaces that we will use.
- **We need to develop new working relationships**
XML is being touted as the lingua franca for documents, data and applications. This has a profound cultural impact on the custodians of information and the developers of tools. “Document people” will now need to work with “data people”. People working on departmental projects will now have to work on an enterprise-wide basis. And organizations will need to consider whether to work with their competitors to develop common XML vocabularies.

For more information ...	
Graphical Communications Association (a volunteer, non-profit membership association which supports XML, SGML and other standards)	http://www.gca.org
IBM's XML site	http://www.ibm.com/xml
ISO, International Organization for Standardization	http://www.iso.ch
Microsoft's XML site	http://msdn.microsoft.com/xml
OASIS, Organization for the Advancement of Structured Information Standards	http://www.oasis-open.org
Oracle's XML site	http://www.oracle.com/xml
RivCom's XML site	http://www.rivcom.com/knowledge/xml.html
Robin Cover's SGML/XML page (most complete listing on XML)	http://www.oasis-open.org/cover/
XML UK User Group	http://www.xmluk.org
The World Wide Web Consortium	http://www.w3.org
The Whatis site (for a brief description of XML)	http://whatis.techtarget.com/
XML.COM (features a rich mix of information and services for the XML community)	http://www.xml.com
XML.ORG (information about the application of XML in industrial and commercial settings and a reference repository for specific XML standards such as vocabularies, DTDs, schemas and namespaces)	http://www.xml.org

○ **We must do more because we can do more**

Finally there is the cost associated with any new and useful technology. The technology enables us to do more. Doing more needs resources. But if we do not do more we will become less efficient and less competitive.

Conclusions

- XML creates a powerful bridge between the *document and database* worlds.
- It provides a common language for encoding data and applying to it *rule-bound processing*.
- It blurs the distinction between *documents and applications*.
- It provides new ways of looking at the relationship between *documents and data*.
- It lays the groundwork for the next generation of *distributed software applications*, and will enable the enterprise to take advantage of secure, robust, global and cheap Internet computing.
- It is simple, powerful and easy to implement.
- XML will not go away.

Sure, some of the claims for XML will prove to be exaggerated and some of the supporting standards will not stand the test of time. But the simplicity of the core standard, combined with the support from the software vendors, together with the hype that has been generated by the press and the analysts, leads to the conclusion that XML will indeed (to use the Gartner Group analogy) transcend the *Trough of Disillusionment*, climb the *Slope of Enlightenment* and reach the *Plateau of Productivity*.

So what should we do now? We should analyse our requirements; track how XML is being used in our business sector; establish some pilot projects that bring business benefits in their own right; and prepare for a world where XML makes the Internet and the Web even more central to our day-to-day activities.



Daniel Rivers-Moore is Director of New Technologies at RivCom, a UK-based consultancy specializing in helping companies and organizations adopt leading-edge technologies for information management and delivery. He was actively involved from the outset in the development of XML and its related technologies. He has served as Joint Project Leader of the STEP/SGML Harmonization initiative for bringing together technical documents with engineering data, and is editor of NewsML, the XML-based standard for the management and delivery of multimedia news.

A founder member of TopicMaps.org, Mr Rivers-Moore served as chair of the subgroup that developed the XML Topic Maps Conceptual Model. He is a member of the board of management of Knowledge on the Web (KnoW), a collaborative initiative aimed at furthering the development of the latest generation of Web technologies in the service of knowledge management and knowledge sharing.

Appendix A: The XML Family – a partial list

DTD (Document Type Definition)

The DTD says, “*this is what you are going to get*”.

It provides the rules that are applicable to a set of XML documents, covering:

- the tags, or element names, that can be used;
- what attributes can be attached to the elements;
- the sequencing and structure of the document.

A DTD can be an external file shared by many XML documents and/or included within the XML document itself.

XSL (Extensible Stylesheet Language) CSS (Cascading Style Sheets)

XSL and CSS are different ways of saying, “*this is what that should look like*”.

To view the XML example given in the main part of this article, we would associate the XML file with an XSL or CSS stylesheet which would define how the tags (<book_title>, <chapter_title>, <para> etc.) would be displayed. XSL is a rich and powerful stylesheet language, developed specifically for XML. CSS is a simpler language that was originally developed for use with HTML.

XSLT (XSL – Transformations)

XSLT says, “*this is how that should be processed*”.

Originally developed as part of XSL, XSLT provides a powerful transformation language. It allows XML files to be manipulated into different structures or formats to enable interchange between different systems and output to different devices.

XLink and XPointer

Xlink and Xpointer are standards which say, “*this is related to that in this way*”.

- **XLink** provides a mechanism for asserting that a relationship exists between a point in one document and one or more other points in the same or other documents, and for providing information about the nature of the link.
- **Xpointer** provides a way of unambiguously specifying the spot within an information container (document) to which you want to link.

Together they provide the mechanism to build virtual documents made up of XML fragments from multiple sources.

XML Namespaces

XML Namespaces provides “*look there to see what this means*” functionality.

Given that XML allows users to define their own tags, it could be that the same tag name has different meanings. I could invent the tag `<title>` to mean the title of a book, while you could also invent the tag `<title>` to mean the title of a person. XML Namespaces provides a mechanism for removing any ambiguities in the meaning of these tags – essential if they were combined into one virtual document.

XML Schema

XML Schema provides an enriched DTD in XML syntax.

The DTD is part of the original SGML standard, and was designed to provide the rules needed for complex document structures. XML Schema will provide a richer language for setting rules for the kinds of complex data-centric applications that were not envisaged when SGML was invented.
