Publishing to XML and XSL W3C Recommendations

Bäck, Asta[1], Kangas, Sonja[1] & Kautto, Vesa[1]

Keywords: Publishing, XML, XSL, W3C, Printing

Abstract: Publishers face challenges as publishing becomes more diversified with printing complemented by electronic media (WWW and CD-ROMs) and vice versa. Also the nature of publishing is changing with an increasing tendenct to personalization. This concerns all publishers who have adpoted the role of a provider of continuous information services.

XML -related standards have inspired great expectations in publishing especially with regard to web publishing. This paper reports two case studies of the possibilities of XML (Extensible Markup Language) and XSL (Extensible Stylesheet Language). The first case is a directory publication and the analysis was made in the summer of 1998. The second case is the study of a news bulletin in the first quarter of 1999. Many new software tools were brought into the public domain or into the market between the two case studies, but still there is not yet enough program support to fully apply XML and XSL in publishing. The time has come to test and prototype the new technologies in order to succeed in publishing with multiple media and increased personalization.

[1] VTT Information Technology, Printed Communications, P.O.Box 1204, FIN-02044 VTT, Finland. email: asta.back@vtt.fi; fax +358 9 455 2839.

Introduction

To succeed in the publishing business, the created content must be easy to use for various publications and media. This is indeed important also with a single medium, but it has gained significance with has increasing number of publishing channels. Many publishers are already using or complementing printed products by WWW, email, Intranet and CD-ROM publishing. The next big step is the widespread use of mobile devices, such as cellular phones, multimedia phones and electronic books. According to some estimates [Pemberton], by the year 2002 some 75 per cent of document viewing on the Internet will be on these portable platforms. Also digital TV will gather ground in the next couple of years.

These channels present different opportunities for carrying information. Some have only a small display area with a low resolution and few colors (ebooks, mobile phones) while others handle all types of information with colors, video and audio (PC, TV). At least with encyclopedias, trade literature, hobby related literature, news and current affairs, and educational materials have a lot of potential for using the same content elements in different publications and media and at different times.

It is vital that publishers make the right decisions about their publishing channels. The more channels they use, the greater their market potential is. Their biggest challenge is to manage all of these channels at a reasonable cost. Content creation and design processes for the supported channels should be as integrated and automated as possible with a minimum of medium-specific efforts. The publishing processes of the future must be media agile.

It is not enough to manage the technical differences of the future channels, because there is a growing demand for personalization and customization of information according to the users' expectations and requirements. With personalization and customization it is more convenient and effective to get information from both the sender's and the user's point of view. Personalization is a tremendous challenge because we do not always know exactly what we expect nor are we willing or able to explain our requirements to the providers of information services.

Traditional mass communications are not likely to disappear overnight, but growth will focus on the new services which will hopefully make our life more pleasant by providing the information we need quickly and in proper form.

Customization and personalization may be based on the information obtained in one of the following ways:

- Selective presentation of information based on information the user has expicitly given
- Selective presentation of information based on previous user behavior, such as buying or reading
- Selective presentation of information based on the reference group's behavior

Also an opportunity of making searches may be viewed as a means of customizing information. It is simpler than the other ways of personalizing information, because no immediate knowledge, assumed or factual, is needed.

Customization may relate to the content, the layout and the presentation, or a combination thereof.

In short, publishing is undergoing major changes. Most publishers cannot afford to confine themselves to using only one medium. They must try to find the right target groups and provide them with the required services. XML and other related recommendations have been proposed as a solution to multi-channel publishing. This paper analyzes application potential of XML and XSL recommendations, their status and applicability on the basis of the two case studies. It also the requirements for XML to become the future tool of multichannel publishing.

Tools and technologies - SGML, HTML

The SGML standard was endorsed in 1986. Its main ideas are separating the content and presentation and describing the content using explicit tagging. SGML did not become a mainstream publishing technology and it is mainly used for technical documentation. This is explained by the fact that SGML has to do with managing knowledge rather than with publishing.

SGML's poor success may also be explained by the number and the cost of the special software tools required. Different tools are needed for

- generating DTDs (DTD=document type definition)
- converting documents from one DTD to another, or from an unstructured format to a DTD
- parsing
- storing documents
- editing and viewing documents
- making a printed version of the document, and
- making the documents viewable on the Internet.

Since technical documentation has been the main application area, the available software tools tend to support the complicated editorial process.

SGML's biggest success came in the early 1990s, i.e.with the HTML. When HTML was defined, the principle of separating the content and the presentation was not followed, and most of the tags exist to control the presentation. The loss of flexibility was offset by simplicity and popularity.

As the number of HTML documents and the number of potential web applications increased, the following problems arose:

- it is awkward and slow to define new HTML tags because this must be done through W3C procedures
- the HTML recommendation is very broad and web development is difficult to manage, because the different user agents (browsers) support different features
- HTML is confined to presenting information to people and it cannot be used for data exchange
- more freedom is needed to control the presentation of web pages.

The last mentioned issue of this list was first solved by the Cascading Style Sheets mechanism.

XML and XSL

The aim of XML was to combine the advantages of SGML and HTML, and to eliminate the complexity of SGML. XML is a metalanguage like SGML, which means that the user can define the semantics and the tags freely. The rules which the XML documents must fulfil were made very clear and precise, with no freedom of choice, making it easy to develop tools, such as parsers, for manipulating XML documents.

The original XML proposals were as follows:

- XML (Extensible Markup Language) for syntax and structure definition (November 1996),
- XSL (Extensible Stylesheet Language) for document presentation (August 1997) and
- Linking (April 1997).

XML version 1.0 became W3C Recommendation in February 1998 [Bray & al. ed.]. The other proposals are still under development, and are expected to be completed in the summer of 1999.

XML only addresses the structure of the document and does not say anything about how to show or present the data. XSL (Extensible Stylesheet Language) is being defined for defining presentation. By April 1999, two working drafts have been published and the princilpal ideas may be assumed to be those which have been presented already, although changes are indeed possible.

The previously mentioned Cascading Style Sheet mechanism and XSL have distinct differences: CSS is intended for defining web documents while XSL will cover both printed and electronic media. The XSL working group has announced as its goal is that it should be possible to define even complicated page layouts using XSL. Another difference is that CSS can only define the appearance of an HTML document as such, but XSL allows to change the structure of the document for presentation without changing the original document.

This is possible because according to the latest working draft [Clark, J. & Deach, S., 1998], the Extensible Stylesheet Language consists of two parts: the transformation language and the formatting vocabulary. With the transformation language, the original XML document can be transformed into any other structure. Also some new data can be inserted into the document and new elements created, such as a table of contents. The formatting vocabulary in the XSL Working Draft introduces the concept of formatting objects. These are generic layout and presentation related objects whose semantics and attribute values define how the object should be rendered. The idea is that the programs, such as browsers or print drivers, would know how to convert these formatting objects into viewable results. Some experimental XSL software tools have been brought into the public domain, but most of them make the transformation from XML into HTML, not yet into formatting objects.

Even though the work on XSL has not yet been completed, it is clearly a very important development. If it is accepted by most software providers, it will have a great impact on how documents and publications are designed and distributed to different media. It may be the tool that allows media agile publishing to become a reality. Bosak (1998) has pointed out that layout and presentation management is far more difficult than merely the exchange of contents made possible by XML. According to Marx (1998), much value is added to a document when the layout is designed, and it should be possible to conserve and reuse also the layouts. XSL seems to have the potential to make this possible.

Two CASE-analyses of XML-based publishing

Two case studies are reported here. The first analysis is of a reference book made during the summer of 1998. The second study was made during the first four months of 1999 of a news bulletin. The aim of these case studies was to assess the potential of XML from two angles: what is required to use it to produce the existing publication and what new opportunities does it offer.

CASE 1: A reference book

The analysis was made during the summer of 1998, when less than 6 months had passed since the issue of the XML recommendation and only the initial proposals had been published of the style sheets and linking mechanisms. Some preliminary and experimental tools were available at that time but only a few commercial software with XML support. The available products were modifications of the existing SGML software.

The contents of the analyzed reference book are kept in a database. At that time the contents had only been published in book form but there were plans were made for CD-ROM and web publication.

The information to be published in this reference book is gathered from selected people by sending them a form to be filled in. Much of the data only needs updating with the most recent information and only new entries are fed in completely.

The database was designed to store the data for the printed version and the main idea was to have the information as concise as possible and to store it ready for publishing. The page count of the book is limited and every effort was made to maximize the number of entries included. The book is not easy to read, because there are only a few subheadings per entry and a lot of abbreviations are used. Since there is no space in the book for complete terms or subheadings, no such information is stored in the database, and most of the data for a single entry is stored in one field.

In this case, XML could be introduced without showing it to the editors, because also with XML the data could be fed in using forms. During the analysis, we designed and proposed a detailed DTD to capture the structure of the information. If this DTD were taken into use, the editors would have to feed in the data into many separate fields. Extra structure would be useful with electronic publications: it would improve searches and it would make it possible to present the information in a user-friendly way.

In this case, it would not be necessary to store the data in the XML format. It would be possible to convert the data into XML as needed. We made a demonstration by storing the data in an SQL database. A fetched record was converted into XML and combined with a style sheet for converting the document into HTML (Figure 1).

We used an experimental piece of software, called msxsl by Microsoft, to convert an XML document into HTML according to the specifications given in the attached XSL style sheet. Using this software, we were able to view the information on an HMTL browser. There was also a corresponding ActiveX control to make the conversion on the fly when the XML document was requested by the browser.

This tool and the very preliminary XSL draft did not allow any complicated layout for the page, but we were able to produce an acceptable table to present the data (Figure 2). With XSL, different views can be given of an XML document without any changes in the XML document; it is possible to show only a part of the document, and/or to change the order of the data for the presentation. Text and graphics can be added to the page through the XSL style sheet. The msxsl program and the ActiveX control are no longer relevant or even available, as they were based on the original XSL proposal [Adler, 1997] and XSL has changed considerably since then.

For a printed version of XML data, one either needs a page layout tool that can process tagged text, or the tagged text needs to be preprocessed by removing the tags and/or by changing the tags so that a page layout tool recognizes them in a way relevant to the appearance of the printed page. In this case, a relatively simple tool could be used to remove and to convert the tags, and it was not necessary to use a special XML page layout tool. Some of the tags were removed completely, because the elements they marked required no special effect on the page. Some of the tags were replaced by rendering-specific commands. For example, the tag <surname> was replaced by a command which made the page layout software (Quark Xpréss) to render the content of the element bold.

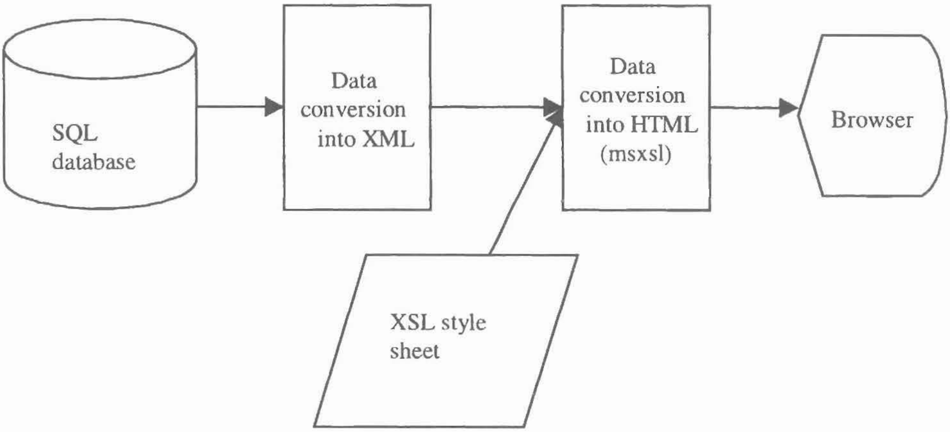
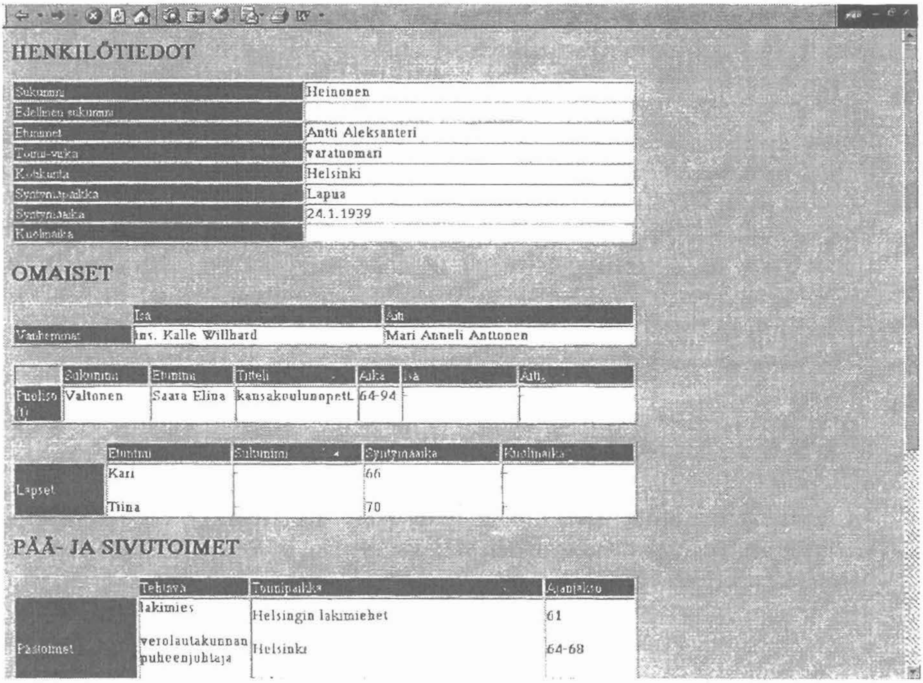


Figure 1. Converting data into XML, combining it with an XSL style sheet to transform it into HTML for display on a browser.



HENKILÖTIEDOT

Sukunimi	Heinonen
Edellinen sukunimi	
Etunimet	Antti Aleksanteri
Toimi-virka	varatuomari
Kotikunta	Helsinki
Syntymäpaikka	Lapua
Syntymäaika	24.1.1939
Kuolinaika	

OMAISET

	Isä	Äiti
Vanhemmat	ins. Kalle Willhard	Mari Anneli Anttonen

	Sukunimi	Etunimi	Titteli	Aika	Isä	Äiti
Puoliso (1)	Valtonen	Saara Elina	kansakoulunopett.	64-94	-	-

	Etunimi	Sukunimi	Syntymäaika	Kuolinaika
Lapset	Kari	-	66	-
	Tiina	-	70	-

PÄÄ- JA SIVUTOIMET

	Tehtävä	Toimipaikka	Ajanjakso
Päätoimet	lakimies	Helsingin lakimiehet	61
	verolautakunnan puheenjohtaja	Helsinki	64-68

Figure 2. The HTML page generated of an XML file according to the instructions given in an XSL style sheet.

Conclusions from Case 1

The case clearly showed that even if the data is in a database, it can be tailored to serve only one specific medium, the printed on in this case. In this case, the big issue is to decide, whether or not to add more structure to the data and to replace the abbreviations by complete terms. This would be quite a job, because it would not be done fully automatically. XML could be used in this case so that, if required, it would not be visible to the editors.

The analyzed publication is well suited for database-driven editing and publishing on the web. At the time of the analysis, there were many ways of publishing the content on the web directly in HTML. In this case, the application of XML becomes more interesting with a company-wide view. By using XML as the storage format or by providing automatic conversion from native formats into XML, components of different publications could easily be combined, and the same tools and methods used to produce several publications.

Case 2: A news bulletin

The bulletin is a newsletter which comes out eight times a year. It is published by VTT Information Technology. The bulletin carries printing and publishing related news in Finnish. Our research scientists write most of the articles, which indeed is not their primary duty. The principal publication channel is the printed. The bulletin has also a website, but it only contains the pdf version which is identical with the printed copy. An email message is also sent to the recipients to inform them that a new issue has come out.

This analysis was made during the first four months of 1999. Even though GT Bulletin this is a small publication, it contains many typical elements of magazine publishing. There are, however, no advertisements, the bulletin is printed in black and white, and the page layout is fairly simple. No proper editorial system is used.

The editors of the bulletin would like to make the following improvements on the WWW version of the paper:

- Increased searchability
- Presentation of the news according to the reader's interests (by means of a user profile stored and maintained in the system, or by allowing the users to make queries).
- A more effective use of the archives
- Links to other relevant materials published by VTT
- Possibilities of including new types of information, such as videoclips and audio

- It would also be good to automatically include the principal headlines of the latest issue could be included in the email messages sent to the subscribers.

The current production process is as follows:
1. Creation of text (office software for word processing, such as Word)
2. Creation of graphics (office software, such as PowerPoint and Excel)
3. Page layout (PageMaker)
4. PDF for digital printing and web publishing

Production of the printed copy

The process begins with the writing of the articles. In the analysis we had to consider that many of the writers only write a few articles per year. This means that large investments in software or training are not justified for writing the articles of the bulletin alone. Other important considerationsa are that the articles are usually written by only one writer and that articles of one issue are not directly interrelated. Links to previous articles on the same theme would, however, be useful.

We can start using XML for publishing at different stages of the operation:
1. the articles are written directly in XML
2. the articles are converted into XML as soon as the writers have completed them and before the articles move to the next production phase
3. the XML format is created after the printed copy is ,completed so that its main purpose is to support electronic publishing or archiving.

It is indeed also possible to have separate routes for the electronic and the printed versions after the text is completed: the non-XML format may be used for page layout, and the articles may be converted into XML for achiving and electronic publishing. In this case, the printed and the electronic articles may become different.

Direct text creation in XML requires software support. Several tools are available for that. These tools can be grouped as follows:
- add-ons to office software (Word, WordPerfect)
- structure editors
- DTD-specific Java applications

Add-ons are usually suitable when the DTD used is not very complicated and does not have many attributes. Most of the actual structure editors are expensive and many of them are not very user-friendly. The majority of the software available in these two categories were created for SGML and some have been modified to handle XML. Some new XML editors have been announced in the last six months and they should be brought into the market soon.

To our knowledge there is only one publicly available application in the third group. It is called Xeena and was developed by IBM [s1]. It creates an editor according to a specified DTD. Also attribute values can be added to the XML document. The program as such seems to work well but it is not a very practical tool for writing long text documents, because the text is fed in by using a separate window that contains one element, such as a paragraph, at a time.

As mentioned before the complexity of the DTD has an effect on tool selection. In this case, there was no existing DTD that should be used. The XML recommendation acknowledged two kinds of XML documents:

1. Well-formed documents which follow the XML rules but do not have a DTD, and
2. Valid XML documents which conform to a DTD.

We find that a DTD is needed in a publishing application. With the help of the DTD it is possible to make sure that the articles have the required elements. A common DTD is also needed to automate page layout and to manage the knowledge in the documents.

Yet another issue that needs to be defined is the level of the DTD: do we only have a DTD for the article, or do we make a DTD (also) for the issue, containing several articles and other elements needed for the publication.

When planning the DTD, we also have to decide what kind of metadata is useful to store, and where and how this metadata should be stored. XML allows it possible to tag the document so that the tags describe the content of the document. It may, however, be impractical if we have to search through the entire document to find out what it is all about. To make document management easier, the metadata can be tagged in a metadata element, or a separate metadata document is created.

Since in this case the grouping of the articles into the issue is not very central after the pdf file has been created for printing, perhaps the right solution would be to focus the system on individual articles. The information about the publication data of each article in the news bulletin can be stored in the article metadata. The articles of the GT Bulletin are now grouped into 14 categories according to their contents. These categories will be used as the starting point for storing the metadata and customisation of the web site.

In this case, it was not yet possible to set up an XML-based publishing process for the printed copy. For text creation, we propose to keep on using our normal office sotfware in a controlled way, saving the documents in a public format (such as RTF or HTML) and converting them into SGML! The reason for this is that at the moment the most suitable page layout tool can only import SGML files to make the page-layout for the printed copy. This tool can, however, export the files to XML, so that we can use the XML format for archives and for web publications. We have designed a relatively simple DTD that fulfils our basic requirements, making sure that it can be used with these tools.

Compared with the publishing process we began with, the new one requires an extra operation (conversion into SGML), which is partly compensated with the more automatic page layout. The principal justification for using SGML/XML is found in electronic publishing and archiving.

New opportunities for web publishing

In this case, the work with XML can begin after the articles have gone through the page layout software.

Since the March 1999, it is possible to publish XML documents directly on the web: one of the available browsers (Internet Explorer 5.0 by Microsoft) supports XML and also partly XSL. We could, at least in principle, require that our intranet users used this browser to see our GT Bulletin web pages. This is not the right approach, if we bear in mind the general trend towards publishing in different media. We should, on the contrary, try to support as many user agents and media types as possible.

One way to make it possible for non-XML browsers to render XML documents is to convert the XML file into another format at the server. Experimental software products have already been published at least by IBM (the XMLEnabler) [s1] and the Java Apache Project/ The Cocoon-servlet [s2] for such a conversion. These servlets check the requesting client type using the user-agent field of the HTTP header. The requested XML file is linked to several XSL files, and the appropriate file is chosen based on the type of the requesting browser. In this way we can have only one original file that contains the information, but serve it in different versions. Thanks to the transformation capabilities of XSL, the elements that cannot be properly presented on some browsers may be removed without changing the original document. If and when the formatting objects are implemented, the number of necessary style sheets will decrease.

This example shows how we can already work with XML and XSL, and move on to multiple media publishing. At the moment, this is still an experimental way of publishing on the web but it has a lot of potential for future development.

Personalization is made based on the content. XML gives us the tools to manage the knowledge once the tags and the metadata are there. There is a limit to how much manual work can be put into tagging and describing the documents. Our plan is to test how the articles in the XML format could be automatically tagged or how the describing metadata extracted from them in order to make it cost-effective to describe the data.

Conclusion from Case 2

Since our first case study, there have been many new developments in commercial and published experimental XML-related software. But in this case, XML could not yet be applied quite as we had hoped. It is, however, possible to use it and to get firsthand experience in XML-based publishing. The biggest challenges are to find ways to tag and to describe the articles automatically to tailor the site for different user groups, and to handle page layout generation for multiple media with XSL.

Concluding discussion

During the past few years a lot has happened in web publishing. With XML and XSL, web and print publishing processes can be made more integrated than before. The XSL and XLink recommendations have not yet been finalized. For these recommendations to have a real impact, it is not enough that they are completed, but they must also be implemented in commercial software tools. Some 15 months have passed since XML version 1.0 was approved. During the last eight months many big players in the software industry have publicly committed themselves to XML and several have produced practical implementations. The fastest progress in XML development has been recoreded in transaction and data exchange applications while publishing has advanced more slowly. We hope that this will change when the recommendations are finalized and approved.

XML was created as a simplified version of SGML. It is simplified in the sense that it is easier to develop programs that handle XML data than it is for handling SGML data. It will also be less complicated to publish XML data on the web, which is a major improvement. But when it comes to managing knowledge, XML is not any easier than SGML. Designing DTDs may require as much planning with XML as it did with SGML, so considerable efforts may be needed up-front. However, XML allows a more straightforward approach, when we deal with publications which do not require a detailed inherent structure. In these cases, a relatively simple DTD could be used (such as Core XHTML) and more structure could be added gradually if the publishing applications really require additional structure. Additional structure is useful particularly with electronic and customized publishing.

A common DTD should be applied at least at company level and most preferably industrywide. As an example, we can mention the DTDs which have been defined for news articles [Anon. 1999a], [Anon. 1999b]. If these DTDs gain ground, special tools will be available to work with documents that conform to these DTDs. A common DTD, at least at company level, is also to be recommended to facilitate the management of style sheets and to maximize their reuse. Innovations and good implementations of layout tools are needed to support the making and management of style sheets. This is a challenge [Karatal, 1998].

Even though publishing to XML and XSL W3C Recommendations is still at experimental stage, it is already possible to work with them and to start creating processes that can produce publications to multiple media. In addition to technical issues, customized publishing and publishing to multiple media require a lot of work in defining what kind of content is suitable to these different media and how this metadata can be added to the content as automatically as possible. Even though many publishers have long ago made the transition to digital processes, it does not automatically mean that the content is useful for other media than the printed one. In these two analyzed cases it is necessary to change the process starting from the content creation stage to make it possible to publish the data meanlingfully and efficiently in different media.

Literature Cited

Adler. S., & al.

1997 A proposal for XSL submitted to W3C on 27 August 1997. http://www.w3.org /TR/NOTE-XSL.html

Anon.

1999a NITF development update. Newspaper techniques, vol. No, March 1999. p. 44. See also: International Press Telecommunications Council (IPTC), <News Industry Text Format> in XML. http://www.iptc.org/iptc/

1999b About XMLNews. http://www.xmlnews.org/intro.html

Bosak, J.,
1998 Media-Independent Publishing: Four Myths about XML. IEEE Computer, vol 31, No 10, October 1998, pp. 120-122.

Bray, T. & al. editors
1998 Extensible Markup Language (XML) 1.0. W3C Recommendation 10-February-1998. http://www.w3.org/TR/1998/REC-xml-19980210.

Clark. J. & Deach, S., editors
1998 Extensible Stylesheet Language (XSL) Version 1.0. World Wide Web Consortium Working Draft 16-December-1998 http://www.w3.org/TR/WD-xsl

Karatal, K.,
1998 Web Template Design: The Need for New Tools. XML '98 Conference. GCA. 15.-18.11.1998. Chicago, IL, USA. 6 pp.

Marx, A.
1998 Case Study: Editorial perspectives on content creation/storage and workflow management for multiple media publishing. Maximising the value & impact of publishing assets through effective Content Management. Pira Internation Conference, 29.-30.9.1998 London, UK. 1 p.

Pemberton, S. & al.
1999 W3C Workin Draft XHTML™ 1.0: The Extensible HyperText Markup Language A Reformulation of HTML 4.0 in XML 1.0. http://www.w3.org/TR/WD-html-in-xml

Software

[s1] http://www.alphaworks.ibm.com/tech/XMLEnabler

[s2] http://java.apache.org/