The Role of XML in Open Hypermedia Systems

Jacco van Ossenbruggen, Anton Eliëns and Lloyd Rutledge

Position paper for the 4th Workshop on Open Hypermedia Systems, Hypertext '98.

Abstract

The World Wide Web Consortium [W3C] has recently issued XML 1.0 as a Recommendation [XML]. XML, the Extensible Markup Language, is a data format for structured document interchange on the Web. In this paper, we put forward the position that XML, together with (future) standards based on XML, will play an important role in the development of open hypermedia systems. Many of objectives of the XML community are similar to the objectives of the open hypermedia systems (OHS) community. This paper discusses these similarities, but also stresses the differences between the architecture and protocol-oriented approach which is typical for the OHS research community versus the document oriented approach of the XML community.

Introduction

Open hypermedia systems are designed, quoting Uffe K. Wiil [OHSWGchair]:

> ... to introduce hypermedia technology into as many applications and components of existing computing environments as possible to evolve gradually current computing environments into a world-wide, unified hypermedia environment spanning multiple computing platforms.

While both the XML and the OHS community share these ambitious goals, they tackle the related problems from different perspectives. We will characterize these perspectives as document-oriented and system-oriented, respectively.

From a document-oriented perspective, an open hypermedia system is a system which does not have a single, fixed hypermedia document model. An OHS is able to process an extensible set of document types (all having a different markup scheme), to recognize the (possibly complex) hyperlink structures which are encoded into the documents, and to present the documents in an appropriate way to the user. From this perspective, the Web currently does not qualify as an OHS, because browsers can not be easily extended with new document types: there are only very limited facilities to tell the browser how to recognize links encoded differently from HTML links, or to define how new document types should be presented to the user. In contrast, open hypermedia document models focus on the facilities supporting structural, domain-dependent markup, facilities to use common link structures across different document sets, and generic ways of defining how to present the encoded information, usually in the form of style sheets.

From an architecture or protocol-oriented perspective, an open hypermedia system needs to be able to offer generic hypermedia services to different applications. From this perspective, the Web does not qualify as an OHS, because it requires other applications to adopt HTML as the main document format, which would require (at least) a major rewrite for most applications. In contrast, an OHS can be seen as a middleware component offering link services and/or storage facilities to a wide variety of applications, each with their own data models and document formats. Open hypermedia system models focus on the design of the OHS architecture, the interfaces and (link)protocols which are defined by the various components in the an OHS environment and the main component technology used (e.g. CORBA, DCOM etc).

The structure of this paper is as follows. First, we explain the role of XML and other document-oriented standards in open hypermedia systems, focusing on the areas related to document markup, hyperlinking and presentation. Secondly, we look at XML from a more system-oriented perspective, focusing on link and storage services and component technology. Thirdly, we discuss the current limitations of XML-based open hypermedia technology.

Hypermedia Document Formats on the World Wide Web

HTML-based systems can be characterized as being cheap, simple and intended for general use by a large audience; but too limited for more advanced hypermedia applications. On the other hand, SGML-based systems do not have the limitations of HTML. But SGML systems are considered by many as too expensive and too complex, and they are in practice often geared towards a more specific use by a smaller audience. As a result, the two research communities (that is, the Web community and the SGML community) were dominated by different research agendas and remained remarkably separate.

XML was developed to bridge this gap between HTML and SGML. XLink, XPointer and XSL are other languages closely related to XML. XLink [XLink] and XPointer [XPointer] define common link and addressing structures which can be used in XML documents, and XSL [XSL] defines a style language for defining how XML documents should be presented to the user.

Table 1: Structured document standards on the Web

	HTML family	**XML family**	**SGML family**
Markup	HTML	XML defined	SGML defined
Hyperlinking	HTML	XLink + XPointer	HyTime,TEI
Style and presentation	CSS	XSL	DSSSL

The relationship between XML, XLink, XPointer and XSL, and the languages related to HTML and SGML are listed in table 1. The following sections will discuss the three rows of this table.

Markup

Currently, HTML is used as the common markup and linking language on the Web. From the moment of introduction HTML has been severely criticized, both by the SGML as the hypertext community. The concerns of the SGML community addressed HTML's fuzzy separation between structural markup and style issues. This problem was (partly) solved by the introduction of the CSS style sheet language, which allows authors to separate these issues. However, the primary concerns of the SGML community were related to the "one size fits all" approach of HTML, which contrasted sharply with the many SGML document types in use within the SGML community, all tailored to a specific application. SGML requires applications to formally specify the elements they want to use in a document type definition (DTD).

To make delivery of SGML documents over the Web as simple as that of HTML, XML was designed as "lite" subset of SGML, in which many of the more exotic features of SGML have been removed. One of the more practical differences between XML and SGML is the fact that in XML, the document type definition is optional. This allows for lightweight documents and implementations which want to employ XML's flexible approach to markup, but do not need the overhead of parsing DTDs and document validation.

Linking

While the SGML community criticized HTML for its poor markup facilities, the hypertext community criticized HTML's limited support for hyperlinking. HTML's simple linking facilities could not compete with the more advanced linking found in many of the more traditional hypertext systems. The Web's simplistic linking model was especially problematic for building open hypertext systems: to support linking from existing applications (word processors, spreadsheets etc) and to integrate more powerful hypertext systems into a Web environment. Out-of-line links, for instance, are not supported in HTML, but are generally regarded as an essential ingredient to open hypermedia systems.

As a result, the hypertext and Web communities were also dominated by different research agendas, and the two communities can still not be characterized by intensive cooperation (the video-link between the '97 Hypertext and World Wide Web conferences did only partially make up for the fact that the two conferences were scheduled in the same week, but on different continents).

However, the need for markup supporting the more complex linking requirements of the hypertext community has been recognized by the SGML community (see the second row of table 1) for quite some time. Both the HyTime standard and the Text Encoding Initiative (TEI) have developed common conventions for the encoding of advanced link

structures (See, for instance, [OHS97]). But again, these solutions never reached a larger (Web)audience, due to the complexity of the standards involved.

So the gap between the simple, but limited markup of HTML and the advanced, but more complex markup of SGML is rather similar to the gap between HTML's simple and limited linking facilities when compared to the advanced, but complex linking support of HyTime and the TEI. XLink and XPointer have been developed to bridge this gap by defining a small set of link structures and addressing methods, combining the best of HTML, HyTime and TEI-based linking.

Style and presentation

The way HTML documents need to be presented was initially hard-coded into a Web-browser. Only recently, this has been changed by the introduction of CSS. In more open document systems, such as those based on SGML, presentation information cannot be hard-coded into the browser because of the wide variety of document formats which are accepted by the system. Therefor, these systems need a stylesheet mechanism such as DSSSL (see the third row of table 1) to provide presentation and style information. XML-based browsers, which also accept different document types, also need a style language. But (again!) the style language of the SGML community was considered to be too complex for widespread usage on the Web. Additionally, the strict separation between structure and presentation of DSSSL did not match current HTML-based Web-applications, where this distinction is less clear. XSL has been developed as a lightweight subset of DSSSL, with additional facilities to operate in a HTML-based environment.

XML in Open Hypermedia Architectures

From the document-oriented perspective, the XML-family of languages fulfills the basic requirement of open hypermedia systems as far as markup, linking and presentation is concerned. In the next section, we will focus on the use of XML from an architecture and protocol-oriented perspective.

A standardized repository format for open hypermedia link-servers

The most obvious member of the XML-family which is relevant to the OHS community is XLink, the XML Linking Language. Since XLink supports out-of-line links by concepts based on proven linking technology from open standards as HyTime and the TEI, it can provide a well-founded basis for defining a standardized format for a OHS link-server component. XLink extends HTML's binary, one-directional, navigation-only links by generic, n-ary and multi-directional relationships, which are sufficiently powerful to encode the hyperlink functionality required by most contemporary hypermedia systems. A standardized repository format makes hyperdocuments less dependent on the specific link-server used during authoring, and will increase the interoperability of the link-servers of different OHSs in general.

Additionally, XPointer, the XML Pointer Language, provides a powerful language for addressing into the internal structures of XML documents (or basically any hierarchical structured document). XPointer is an obvious candidate for a open hypermedia anchor language.

XML and data interchange

XML is, like its "big brother" SGML, promoted as a language for structured document markup. However, from the very beginning, the hidden agenda of XML has been the development of a standard interchange format for structured data on the Web. Currently, to enable data-interchange over the Web, data-oriented applications need to convert their data to HTML (often losing important structural information) or extend HTML by introducing proprietary extensions to HTML. From this (data-oriented) perspective, XML was designed to avoid that every EDI application on the Web needs to define and implement its own extensions to HTML in an ad-hoc fashion. From an open hypermedia system perspective, the use of XML has the additional advantage that the hypermedia data structures of a specific OHS can be manipulated and transformed to other formats by using standard XML technology.

XML component technology

Since most information is typically encoded in XML (including the documents themselves, link information, style sheets, meta-data etc), the XML parser will play a crucial role in an XML-based hypermedia system. Therefore, its is important that the components of a open hypermedia system can access the services of the parser in a standardized way. Currently, there are two important proposals for standard interfaces to XML parsers. The first interface is the tree-based Document Object Model [DOM]), the second is the event-based Simple API to XML [SAX]. For DOM, the interface is defined using OMG's IDL, and direct language bindings for Java and ECMA Script are provided as well. The tree-based interface of DOM provides a good basis for XPointer implementations. Not all applications need the full, random access to the document tree that DOM provides, and such applications typically benefit from the more lightweight approach of an event-based API. SAX provides such an API, and SAX drivers for all major XML parsers implemented in Java already exist. Additionally, there is a Java implementation of DOM [SAXDOM], build upon the SAX interface. As a result, applications can currently access all major XML parsers independent of the chosen interface paradigm.

Limitations of XML-based open hypermedia

One of the drawbacks of using XML for building an open hypermedia environment is the fact that XML, and especially the standards related to XML, are in a very early state of development. While XML 1.0 has been approved as an official W3C Recommendation, and many XML parsers exist today, additional support for XML is not very common yet. Especially the link and style languages discussed above are not widely implemented. Additionally, while XML provides application the flexibility to

define their own markup language, it does not provide any solutions for making existing, non-XML applications "hypermedia aware".

Another fundamental limitation of the XML-family is its focus on text-based applications. Especially XSL has a pure page-based output model. This will be a serious problem for more advanced hypermedia systems supporting spatial [VIKI] or time-synchronized hypermedia [Jamming]. For instance, our group has participated in the development of SMIL [SMIL], the Synchronized Multimedia Integration Language. During the development, both XSL and XLink did not suffice to capture the semantics of presentation and hyperlinking of multiple streams of media. As a result, SMIL players need to be implemented as dedicated applications.

Another approach to avoid the limitations of XSL is by using a more powerful style sheet language. We have developed a hypermedia Web browser that used a general purpose scripting language as a style sheet to define the link and presentation semantics of its SGML and XML documents [Animating], [SGMLweb], . The advantage of this approach is that one has flexible access to all the (multimedia) support built into client, without requiring a fixed hypermedia document model. A disadvantage is the rather low level of programming required in the style sheet to define the link and presentation semantics of the hypermedia documents.

Conclusions

Despite some limitations, XML and related standards provide the basic building blocks for designing open hypermedia documents which can be delivered over the Web. Additionally, it can provide a starting point for standardized and interchangeable data structures for various OHS components, including link servers. In many ways, XML has bridged the gap between:

- the simple, and limited markup on the Web versus the advanced, but complex markup of SGML.
- the simple, but standardized data formats on the Web versus the more advanced but proprietary formats used for EDI.

Our claim is that XML can also bridge the gap between the Web community and the OHS community, and turn the Web into the advanced, open and global hypermedia system we wished it to be from the very beginning.

References

Animating

Animating the Web --- An SGML-based Approach. Anton Eliëns, Jacco van Ossenbruggen, and Bastiaan Schönhage. In: Proceedings of the International Conference on 3D and Multimedia on the Internet, WWW and Networks, 17-18 April 1996, Bradford. British Computer Society, April 1996. Available at http://www.cs.vu.nl/~dejavu/papers/bradford96.ps

DOM

Document Object Model Specification, Version 1.0. Work in progress. Available at http://www.w3.org/TR/WD-DOM/

Jamming

Jamming (on) the Web. Anton Eliëns, Martijn van Welie, Jacco van Ossenbruggen, and Bastiaan Schönhage. In: Proceedings of the 6th International World Wide Web Conference --- Everone, Everything Connected (pages 419--426) O'Reilly and Associates, Inc., April 1997. Available at http://www6.nttlabs.com/HyperNews/get/PAPER38.html

OHS97

Applying the HyTime Model to the Open Hypermedia Protocol LocSpec. Lloyd Rutledge, Lynda Hardman and Jacco van Ossenbruggen. 3rd Workshop on Open Hypermedia Systems, Hypertext '97, Southampton, England, April 6-11, 1997.

OHSWGchair

Message from the OHSWG chair. Uffe K. Wiil. Available at http://www.csdl.tamu.edu/ohs/intro/chair.html

SAX

The Simple API for XML. David Megginson. Available at http://www.microstar.com/XML/SAX/

SAXDOM

http://www.docuverse.com/personal/saxdom.html

SGMLweb

Web Applications and SGML. Jacco van Ossenbruggen, Anton Eliëns, and Bastiaan Schönhage. Proceedings of the Sixth International Conference on Electronic Publishing, Document Manipulation and Typography. Editors: David F. Brailsford and Richard K. Furuta (volume 8(2 & 3) June and September 1995) Electronic Publishing --- Origination, Dissemination and Design, John Wiley & Sons, Ltd., 1996 Available at http://www.cs.vu.nl/~dejavu/papers/ep96.ps.gz

SMIL

Synchronized Multimedia Integration Language (SMIL) 1.0 Specification. Editor: Philipp Hoschka. Work in progress. Available at http://www.w3.org/TR/PR-smil/

VIKI

Spatial Hypertext and the Practice of Information Triage. Catherine C. Marshall 1 and Frank M. Shipman III. Proceedings of the Eighth ACM Conference on Hypertext: Hypertext 97, Southampton, UK. ACM Press (1997).

W3C

http://www.w3.org/

XLink

XML Linking Language (XLink). Editors: Eve Maler and Steve DeRose. Work in progress. Available at http://www.w3.org/TR/WD-xlink

XML 1.0

Extensible Markup Language (XML) 1.0 Editors: Tim Bray, Jean Paoli and C. M. Sperberg-McQueen. W3C Recommendation 10-February-1998. Available at http://www.w3.org/TR/REC-xml.

XPointer

XML Pointer Language (XPointer). Editors: Eve Maler and Steve DeRose. Work in progress. Available at http://www.w3.org/TR/WD-xptr