The Evolution Of Web Protocols

Brian Kelly, UK Web Focus, UK Office for Library and Information Networking, University of Bath, Bath, BA2 7AY, UK

E-mail: b.kelly@ukoln.ac.uk
Phone: +44 1225 323943
Fax: +44 1225 826838

Abstract

This paper outlines the evolution of World Wide Web protocols. The paper reviews the original protocols developed for the Web, in addressing, transport and data formats. A review of developments of the protocols is given, including developments of web data formats (HTML 4.0, cascading stylesheets and XML), transport (HTTP/1.1 and HTTP/NG) and addressing (URLs). The paper describes how the web initially lacked a metadata architecture and outlines the emergence of a metadata architecture for the web.

The paper includes a review of Web technologies which have a social impact on our society, including the Web Accessibility Initiative, the Digital Signature Initiative and the Platform for Privacy Preferences Project.

Background to Web Protocols

The World Wide Web (often referred to as WWW or the Web) has been defined as "*the universe of network-accessible information, the embodiment of human knowledge*" [1]. The web is based on a set of protocols and conventions which initially covered the data format of resources, addressing of resources, the transport of resources across the Internet:

> **Formats**: The native format for resources on the Web is HTML, the *Hypertext Markup Language*.
>
> **Transport**: The *HyperText Transfer Protocol* (HTTP) provides the mechanism by which Web browsers and other user agents access resources.
>
> **Addressing**: *Uniform Resource Locators* (URLs) provide an address for resources on the Web.

The development of these protocols is given below.

Data Formats

Early Days

HTML (Hypertext Markup Language) is an application of SGML (Standard Generalised Markup Language). The first release, HTML 1.0 [2] provided the hypertext linking which Web users today will be familiar with. HTML 1.0, in keeping with the spirit of SGML of defining the structural elements in documents, included the basic structural elements this used today, such as paragraphs (the <P> element) and headings (<H1> to <H6>) as well as a small number of formatting elements, such as italic <I> and bold <B>.

HTML 2.0 [3] introduced a number of innovations introduced by the Mosaic web browser, developed at NCSA (The National Center for Supercomputing Applications), University of Illinois at Urbana-Champaign. Innovations introduced in HTML 2.0 included forms and inline images.

At the first international WWW conference held in CERN, Switzerland in May 1994 David Raggett (one of the authors of the HTML 2.0 specification) outlined a roadmap for future developments of HTML. HTML 3.0 [4] (which was initially known as HTML+ [5]) would include a range of new features such as tables, richer forms and support for mathematical equations. HTML 3.0 was submitted to the Internet Engineering Task Force (IETF). Unfortunately it failed to be standardised, due to a failure to reach consensus within six months. This failure was due partly to the size and complexity of the proposal and also due to the lack of interest from commercial web browser vendors – with only the Arena browser providing a proof of concept.

In October 1994, the first version of the Netscape browser was released. Although Netscape proved tremendously popular, it also, controversially, announced support for a number of HTML elements which have not featured in discussions of developments to HTML such as **`<CENTER>`**, **`<FONT>`** and the infamous **`<BLINK>`** element [5].

By 1995 Microsoft had become aware of the importance of the web. The initial release of their browser, known as Internet Explorer, was based on a licensed version of the original Mosaic browser and was limited in functionality. By the time Internet Explorer 3.0 was released (by which time the software was being developed in-house), Microsoft were beginning to compete strongly with Netscape for browser market share. This competition resulted in both companies announcing a variety of new HTML elements, with, for example, Microsoft responding to **`<BLINK>`** with their **`<MARQUEE>`** element for displaying scrolling text and Netscape introducing the **`<FRAME>`** element which provided multiple separately scrollable areas within the browser window.

Although the battle for market share helped to stimulate innovation and was warmly applauded by many end users, the innovations to the HTML language were causing difficulties in other areas: vendors of authoring tools were having to choose whether to develop software which supported Netscape's extensions or Microsoft's or both; information providers were faced with the dilemma of developing rich, attractive interfaces which would not be universally accessible and the web protocol community was concerned that the developments were not interoperable across applications and would hinder the evolution of HTML, in particular by the introduction of a range of layout features into a language which was intended to define the document structure.

In March 1997 Microsoft admitted that “proprietary HTML extensions from Microsoft and other vendors have confused the market, hampered interoperability and been ill-conceived with respect to the design principles underlying HTML” [7]. In Microsoft’s pledge they agreed to work with W3C (the World Wide Web Consortium, who are responsible for coordinating developments to web protocols) in further advancing the HTML standard.

HTML 3.2, which was announced in January 1997 [8] recognised that attempts to define developments to HTML independently of the commercial software vendors would fail. HTML 3.2 was based on the current accepted practices (for example it accepted layout features such as **`<CENTER>`**). Future developments to HTML, however, lay in project *Cougar* – W3C’s codename for the new version of HTML. In December 1997 W3C announced that HTML 4.0 [9] (as Cougar became known as) had been accepted as a W3C recommendation. The development of the HTML 4.0 specification was made by a W3C HTML working group which included representation from W3C staff together with software vendors including Netscape and Microsoft.

HTML 4.0 included enhancements in a number of areas, such as more sophisticated forms and tables. HTML 4.0 added features to make web resources more accessible by providing support for people with disabilities and for non-English speaking users. Although HTML 4.0 gave recognition to the widespread deployment of frames, it did not introduce a wide range of new features. It primarily provided hooks for embedding other resources within HTML documents, such as multimedia objects and scripting languages. In addition HTML 4.0 provided support for style sheets.

Style Sheets

As mentioned earlier, HTML was originally intended to define the structure of a document. It has always been recognised that the appearance of a document was important. However it was felt that the appearance should be held separately from the content of a document.

The initial recommendation for style sheets, Cascading Style Sheets 1 (CSS1), was announced in December 1996 [10]. However CSS1 was only partly supported in Microsoft's Internet Explorer 3.0 (which was available at the time) and was not supported in Netscape Navigator 3.0. CSS2 was announced in November 1997 and was released as a W3C recommendation in May 1998 [11].

CSS2 provides a great deal of control over the appearance of a document. CSS can be included inline within an HTML element or within the **`HEAD`** of a document. However for maintenance purposes, it is better if the CSS is included as an external linked file. Changing the house style for a collection of related resources which use the same style sheet will simply require changing a single file.

Dynamic HTML

Amongst the innovations introduced in the Netscape browser, it was the JavaScript scripting language which provided the means for developing some level of interaction and feedback with Web pages. JavaScript was used to develop simple applications and to enhance user interfaces. Microsoft responded by developing their own implementation of JavaScript, known as JScript, and providing support in their browser for VBScript – a development of their Visual Basic language. As with the HTML wars, the lack of standardisation in the scripting languages resulted in interoperability problems. This time, however, ECMA, the European Computer Manufacturer's Association, provided a forum for standardisation. In June 1997 the ECMAScript Language Specification was released [12].

However JavaScript and VBScript were limited in what they could do by the lack of a comprehensive object model for the HTML language. The Document Object Model (DOM) was developed by the W3C to provide a neutral, language-independent programming interface to HTML (and style sheet) resources [13]. The term *Dynamic HTML* (DHTML) has been used to describe the use of a scripting language in conjunction with the Document Object Model to provide rich functionality for web pages.

XML

Although HTML 4.0 used in conjunction with CSS 2.0 and DOM provide a rich environment for defining the appearance of resources and manipulating the appearance, there are still a number of limitations. The HTML standardisation process makes it difficult to introduce new HTML elements. Specialist HTML elements, required by, say, the mathematical community, would be inappropriate for inclusion in a general purpose markup language. In addition there are requirements for defining structural elements which will be specific to a particular organisation (such as a <STUDENT-NUMBER> element which could be needed in an educational context) which would be inappropriate for inclusion in a general purpose markup language.

The *Extensible Markup Language* (XML) has been developed to address these issues. XML has been designed to be extensible, so that agreement on a set of standard elements does not necessarily have to be achieved. For example, the mathematical community could agree that the following XML elements:

```
<apply><times/> <apply><plus/> <ci> F </ci> <ci> G </ci>
</apply> <ci> x </ci> </apply>
```

would define the expression $(F + G)(x)$. The definition of an element set does not have to be agreed by an international organisation. An educational establishment, for example, could defined XML elements such as **<staff-number>**, **<student-number>** and **<semester>** for use within the organisation.

Although XML 1.0 was only announced as a W3C recommendation in February 1998 [14], it is already becoming widely adopted in a number of areas. The Mathematical

Markup Language (MathML), which was announced as a W3C Proposed Recommendation in February 1998 [15], is an XML application, as is the Chemical Markup Language (CML) [16], the Channel Definition Format (CDF) [17], and the Synchronized Multimedia Integration Language (SMIL) [18].

In addition to use within the scientific communities, XML is also being used within the web community to develop new architectural components to the web, especially in the area of metadata, as discussed later.

Transport

HTTP, the *HyperText Transfer Protocol*, governs the transfer of resources between a web server and client. Typically clicking on a hypertext link in a web browser will send a HTTP **`GET`** request to the server. The web server will then send back a series of headers, together with the resource, if it exists.

In the initial implementation of HTTP, HTTP/0.9, the web browser could process the files based on the file suffix. So, for example, a PostScript file with a **`.ps`** could be passed to a PostScript viewer for displaying. This, however, was not a scaleable solution. In HTTP/1.0 [19] files are sent as MIME attachments, such as **`text/html`**, **`image/gif`**, **`text/postscript`**, etc.

Although HTTP/1.0 has been widely deployed, there are a number of problems with it:

- HTTP/1.0 uses TCP, the Internet's Transmission Control Protocol, inefficiently. Since most resources are small, and HTTP/1.0 opens and closes a new TCP connection for each operation, there is a large overhead.

- HTTP/1.0 does not have sufficient facilities for compression.

- HTTP/1.0's caching is very primitive.

HTTP/1.1 [20] was developed to address these deficiencies and to fix a number of bugs in HTTP/1.0. The HTTP/1.1 specification provides support for multiple TCP connections and more efficient support for caching.

A W3C Note on "*Network Performance Effects of HTTP/1.1, CSS1, and PNG*" [21] confirms the performance benefits of HTTP/1.1.

Extending HTTP

Although HTTP/1.1 will provide performance benefits, the introduction of new facilities is still hindered by the standardisation process and the dangers of making HTTP more complex by the introduction of facilities which will be used by only small communities. HTTP faces similar development problems as does HTML.

Just as XML provides a extension mechanism for data formats, the Protocol Extension Prococol (PEP) [22] is designed to provide an extension mechanism for HTTP.

PEP examples which are given in the PEP draft specification include determining whether a server understands and supports the Distributed Authoring and Versioning (DAV) protocol extension and use of a micropayments scheme.

Content Negotiation

We have seen how PEP can be used to provide an extension mechanism for HTTP. Transparent Content Negotiation (TCN) [23] provides an extensible negotiation mechanism, layered on top of HTTP, for automatically selecting the "best" version when the resource is accessed. TCN enables new data formats and HTML elements to be smoothly deployed.

HTTP/NG

Although HTTP/1.1, together with PEP and TCN, are addressing a number of the deficiencies in the underlying transport protocol, we are still faced with a number of problem areas, including the complexity of HTTP, the poor scalability of HTTP when faced with todays network traffic load and the difficulty of introducing applications on the web, other than simple document retrieval applications.

HTTP/NG [24] will be a new architecture for the HTTP protocol based on a simple, extensible distributed object-oriented model. Work on HTTP/NG started recently. As yet there is little information publicly available.

Addressing

In comparison with data formats, developments in the area of addressing on the Web seem to be very slow. A Uniform Resource Locator (URL) [25] is used to define the location of a resource on the Web. For example the URL `http://www.w3.org/TR/WD-math` (normally) describes the name of the computer holding the resource (`www.w3.org`), the directory in which the resource is stored (`/TR`) and the name of the resource (`WD-math`). This is equivalent to defining a book by saying it is located in the Oxford University library, on the third floor, and it's the fourth shelf from the entrance. However the URL system is prone to failure. If an organisation reorganises its website, links to resources are likely to be broken. Similarly if an organisation changes its name, is taken over or sells part of the organisation, a reorganisation of its website to reflect the changes, including changes in the domain name, will also result in broken links.

There have been a number of proposals which attempt to provide a location-independent address for a resource including Uniform Resource Names (URNs) and Persistent Uniform Resources (PURLs).

URNs are intended to provide a persistent and globally unique identifier for web, and more generally, Internet resources. The Internet Engineering Task Force (IETF) has set up a Working Party to define a Uniform Resource Name (URN) framework and an initial set of components that fit this framework [26].

PURLs [27] were developed by OCLC (the Online Computer Library Center) as an interim naming and resolution system for the Web. A PURL acts as a URL which points to a resolution service rather than the resource itself. The PURL resolution service associates the PURL with the URL of the resource and uses a HTTP redirect to access the resource. Content providers are responsible for updating the PURL database when a URL changes.

More recently the Digital Object Identifier (DOI) system has been developed [28]. The DOI system has three components: the identifier, the directory and the database. The system allows identifiers to be assigned at various levels. The directory is a distributed system based on CNRI's Handle system which provides a mapping from DOIs to URLs. DOIs have initially been aimed at the 'traditional' publishing industry, and there are plans to use the DOI as the basis of copyright management systems.

However none of the proposals for replacing URLs have been widely deployed. This is, in part, due to the need for an organisational infrastructure for registering location-independent resources.

Metadata

Metadata can be described as the missing architectural component of the web. The initial web protocols enabled resources to be located (using a URL), transported (using HTTP) and displayed (using HTML). Although HTML enabled simple document structures to be defined, it failed to provide a means of defining, in a machine-parsable way, information about the resource.

The lack of a metadata architecture became apparent with the development of automated robot software which provide indexes of web resources. It is not possible to search for, for example, resources *authored* by an individual, rather than resources about the individual.

The failure of automated search engines to provide effective results resulted in an initiative to develop a set of attributes for resource discovery. The initiative identified 15 core attributes, which became known as Dublin Core metadata (after Dublin, Ohio, the location of the first meeting) [29].

Since HTML did not provide a mechanism for storing structured information, Dublin Core elements initially had to be embedded within HTML documents using the <META> element, as shown below:

```
<META NAME="DC.Creator" CONTENT=”Brian Kelly">
<META NAME="DC.Subject" CONTENT=”Web Developments">
<META NAME="DC.Identifier" CONTENT=”http://www.ukoln.ac.uk/">
```

Figure 1 Embedding Simple Dublin Core Elements in HTML Documents

However it soon became apparent that it was necessary to embedded hierarchical Dublin Core metadata, to include the creator’s name, email address, postal address, etc. Within HTML 3.2, this had to be achieved in a convoluted way, illustrated in Figure 2.

```
<META NAME="DC.Creator" CONTENT="(TYPE=name) Brian Kelly">
<META NAME="DC.Creator" CONTENT="(TYPE=email) b.kelly@ukoln.ac.uk">
<META NAME="DC.Creator" CONTENT="(TYPE=affiliation) UKOLN, University
of Bath">
```

Figure 2 Embedding Hierarchical Dublin Core Elements in HTML Documents

Although HTML 4.0 enabled Dublin Core metadata to be embedded in a more elegant way, work in related metadata areas helped with the development of a more general metadata architecture.

In 1996 the proposed *Communications Decency Act* in the United States hastened the need for a mechanism to enable parents to control access to offensive resources. The World Wide Web Consortium developed a technical standard called PICS (Platform for Internet Content Selection) which enabled information providers to electronically distribute descriptions of digital works in a machine-readable form. Web browser software can process these labels, automatically shielding users from undesirable material. PICS enabled parents and teachers to screen materials they felt were inappropriate for children, rather than censoring what is distributed.

The PICS protocols were developed quickly due to concerns amongst various groups such as software vendors, information providers and civil liberties communities over the effect of the proposed legislation. Although the legislation failed to be implemented, it was recognised that the PICS architecture had a valuable role to play, not only in labelling resources with various descriptions of dubious content, but as a more general mechanism for labelling resources. Configuring a browser to suppress access to resources by a controversial artist, for example, is clearly related to finding resources by the artist. During 1997 the World Wide Web Consortium began work on PICS/NG which would implement support for string values (PICS allowed only numeric rating schemes) which would enable it to provide a more general metadata architecture.

As the web grew in popularity, especially to provide commercial services, the lack of a *trust mechanism* became apparent. In the real world we can make informed decisions regarding the status and trust-worthiness of companies such as “Joe Bloggs Used Car Garage” or “Harrods”. On the Internet, however, we have no way of knowing the status of a website. It is also unclear whether statements we read on the web have any legal validity.

W3C's Digital Signatures Initiative (DSig) uses digital signature, certificate and metadata technologies to provide a solution to the problem of helping users decide what to trust on the Web. is the goal of the DSig project to provide a mechanism to make the statement: "*Signer believes statement about information resource*" [30].

The "PICS Signed Labels (DSig) 1.0 Recommendation" [31], which was published in May 1998, defines a standard format based on extended PICS 1.1 labels for making digitally-signed, machine-readable assertions about a particular information resource.

Together with the work in developing an infrastructure for managing resource discovery (Dublin Core), content selection (PICS) and trust (DSig) there were several other activities taking place related to information around resources, including:

- User navigation: new ways of navigating through websites.
- Collaborative authoring of documents.
- Indexing, off-line browsing and printing of collections of related resources.

Work in this area included Netscape's proposal on "*Meta Content Framework Using XML*" [32] which provides a specification for describing information structures (metadata) for collections of networked information using XML and Microsoft's "*Web Collections using XML*" [33] proposal for providing a metadata framework which can be used for a variety of applications, such as sitemaps, distributed authoring and content labelling.

Both of these proposals recognised the importance of XML for representing the syntax of the metadata. The proposals, together with other related work, led to the development of RDF, the Resource Description Framework, which provides a framework for metadata giving interoperability between applications that exchange machine-readable information on the Web [34].

At the time of writing (July 1998) work in developing RDF is still at an early stage. However RDF does seem to provide a mechanism for pulling together the various related metadata components and adding a new architectural component to the Web.

The Web and Society

The work in developing protocols for managing Internet content selection and developing a trust mechanism forms part of W3C's work in addressing the social impact of the Web. Other work in this area includes addressing privacy issues, public policy and accessibility.

W3C's Platform for Privacy Preferences Project (P3P) [35] aims to develop a specification and demonstration of an interoperable way of expressing privacy practices and preferences by Web sites and users. Users will be informed about and given the opportunity to decide how their data is used. P3P will enable statements about privacy practices and preferences to be made in a machine-readable way. Privacy statements can be made using a digital signature. The P3P work will be closely linked with the work of the Digital Signatures Initiative.

W3C's work on Public Policy [36] is addressing a wide range of policy matters, including content selection and censorship, intellectual property rights, copyright, anonymity, liability and trust and legality.

The PICS protocol for managing content selection has been described previously. The use of PICS as a mechanism for expressing intellectual property rights has been suggested [37]. Although this Internet draft has expired it provides an example of how machine-readable metadata statements about resources have the potential for managing intellectual property rights and other social aspects of the web. The NRC/CSTB/Information Systems Trustworthiness Project has addressed the area of rights management and outlined scenarios for the evolution of rights management over the Internet [38].

Conclusions

This paper has reviewed developments of web protocols from the initial three underlying protocols areas of addressing, data formats and transport. The deficiencies in these protocol areas have been described and a summary of how the protocols are developing in order to address these deficiencies has been given. The development of a new protocol areas of metadata has been given, with an outline of metadata protocol developments.

References

1. W3C. *About The World Wide Web*. 1992. http://www.w3.org/WWW/ (visited 9 July 1998).

2. W3C. *HTML*. 1992. http://www.w3.org/History/19921103-hypertext/hypertext/WWW/MarkUp/MarkUp.html (visited 9 July 1998).

3. W3C. *HTML 2.0*. 1995. http://www.w3.org/MarkUp/html-spec/ (visited 9 July 1998).

4. W3C. *HTML 3.0*. 1995. http://www.w3.org/MarkUp/html3/CoverPage.html (visited 9 July 1998).

5. W3C. *HTML+*. 1993. http://www.w3.org/MarkUp/HTMLPlus/htmlplus_1.html (visited 9 July 1998).

6. Netscape. *Extensions to HTML 2.0*. http://home.netscape.com/assist/net_sites/html_extensions.html (visited 9 July 1998).

7. Microsoft. *Microsoft Pledge on Standards*. 1997. http://www.microsoft.com/internet/html.htm (visited 9 July 1998).

8. W3C. *HTML 3.2 Reference Specification*. 1997. http://www.w3.org/TR/REC-html32.html (visited 9 July 1998).

9. W3C. *HTML 4.0 Specification.* 1998. http://www.w3.org/TR/REC-html40/ (visited 9 July 1998).

10. W3C. *Cascading Style Sheets, level 1*. 1996. http://www.w3.org/TR/REC-CSS1.html (visited 9 July 1998).

11. W3C. *Cascading Style Sheets, level 2*. 1998. http://www.w3.org/TR/REC-CSS2.html (visited 9 July 1998).

12. ECMA. *ECMAScript*. 1997. http://www.ecma.ch/stand/ecma-262.htm (visited 9 July 1998).

13. W3C. *Document Object Model (DOM)*. 1998. http://www.w3.org/DOM/ (visited 9 July 1998).

14. W3C. *Extensible Markup Language (XML) 1.0*. 1998. http://www.w3.org/TR/REC-xml (visited 9 July 1998).

15. W3C. *Mathematical Markup Language (MathML) 1.0 Specification*. 1998. http://www.w3.org/TR/REC-MathML/ (visited 9 July 1998).

16. Virtual School of Molecular Sciences. *Chemical Markup Language* . http://www.venus.co.uk/omf/cml/intro.html (visited 20 July 1998).

17. Ellerman C. *Channel Definition Format*. http://www.w3.org/TR/NOTE-CDFsubmit.html (visited 20 July 1998).

18. Hoschka, P. Synchronized Multimedia Integration Language (SMIL) 1.0 Specification. http://www.w3.org/TR/REC-smil/ (visited 20 July 1998).

19. Berners-Lee, T., Fielding, R. and Frystyk H. *HyperText Transfer Protocol - HTTP/1.0*. http://www.w3.org/Protocols/rfc1945/rfc1945 (visited 9 July 1998).

20. Fielding, R., Gettys, J, Mogul, J., Frystyk H. and Berners-Lee, T. 1997. *Hypertext Transfer Protocol HTTP/1.1*. http://www.w3.org/Protocols/rfc2068/rfc2068. (visited 9 July 1998).

21. Frystyk H., Gettys, J., Baird-Smith, A., Prud'hommeaux, H. Lie, H. and Lilley, C. *Network Performance Effects of HTTP/1.1, CSS1, and PNG.* 1997. http://www.w3.org/Protocols/HTTP/Performance/Pipeline (visited 9 July 1998).

22. W3C. *PEP – An Extension Mechanism For HTTP*. 1997. http://www.w3.org/Protocols/PEP/ (visited 9 July 1998).

23. Holtman, K. *Transparent Content Negotiation in HTTP*. http://gewis.win.tue.nl/~koen/conneg/ (visited 9 July 1998).

24. W3C. *Hypertext Transfer Protocol - Next Generation*. 1998. http://www.w3.org/Protocols/HTTP-NG/ (visited 9 July 1998).

25. Berners-Lee, T., Masinter, L. and McCahill, M. *Uniform Resource Locators*. 1994. http://www.w3.org/Addressing/rfc1738.txt (visited 9 July 1998).

26. IETF, *URN Working Group Charter*. http://www.ietf.org/html.charters/urn-charter.html (visited 9 July 1998).

27. OCLC. *PURLs*. http://purl.oclc.org/ (visited 9 July 1998).

28. The DOI Foundation. *The Digital Object Identifier (DOI) System*. 1998. http://www.doi.org/ (visited 9 July 1998).

29. OCLC, *Dublin Core*. 1997. http://purl.oclc.org/metadata/dublin_core/ (visited 9 July 1998).

30. W3C. *DSig Activity Statement*. 1998. http://www.w3.org/DSig/Activity.html (visited 9 July 1998).

31. W3C. *PICS Signed Labels (DSig) 1.0 Recommendation*. 1998. http://www.w3.org/TR/REC-DSig-label/ (visited 9 July 1998).

32. Netscape Communications. *Meta Content Framework Using XML*. 1997. http://www.w3.org/TR/NOTE-MCF-XML/ (visited 9 July 1998).

33. Microsoft. *Web Collections using XML*. 1997. http://www.w3.org/TR/NOTE-XMLsubmit.html (visited 9 July 1998).

34. W3C. *Resource Description Framework*. 1998. http://www.w3.org/RDF/ (visited 9 July 1998).

35. W3C. *Platform for Privacy Preferences Project*. http://www.w3.org/P3P/ (visited 9 July 1998).

36. W3C. *Public Policy Page*. http://www.w3.org/Policy/ (visited 9 July 1998).

37. Reagle, J. *Using PICS for Copyright Notice and Control*. http://www.w3.org/IPR/work/draft-reagle-pics-copyright-00.txt (visited 9 July 1998).

38. Khare, R. and Reagle, J. *Rights Management, Copy Detection, and Access Control*. http://www.w3.org/IPR/work/NRC-v1.htm (visited 9 July 1998).