Update on World-Wide Web Consortium activities in internationalization

Bert Bos
W3C internationalization coordinator
INRIA/W3C, France
http://www.w3.org/people/bos – bert@w3.org

Abstract

The World-Wide Web (The Web) interconnects machines and people all over the world. It is steadily becoming better at representing all the different languages and cultures of those people, although much still needs to be done. The World-Wide Web Consortium (W3C) is coordinating the development efforts in such areas as HTML, HTTP, style sheets, fonts, electronic payment, etc. This article gives an overview of some of the activities that are currently going on in W3C, in particular: XML, HTML, and CSS.

1 The World-Wide Web Consortium

The World-Wide Web Consortium (W3C) is a consortium of about 180 companies and institutions from all over the world, who have come together to further develop the Web. The activities of the consortium are roughly divided into three domains, called *Architecture, User Interface* and *Technology & Society.*

Architecture deals with the protocols between computers (such as HTTP), with the methods to identify documents and other resources (in particular URLs), with real-time audio and video, object-oriented data models, and mobile code. Natural language doesn’t play a large role in these areas, but there are still cases where the underlying technology needs to provide support for language-related features at the user interface level. For example, when a text document is transported over the Internet, the protocol must have a way of labeling the encoding and passing it on to the receiving end.

The new language XML, for text-based representations of arbitrary structured data, is also part of the Architecture domain. Being text-based, it obviously *does* have many internationalization aspects.

The User Interface domain contains the activities in HTML, style sheets (including CSS), graphics, fonts, and colors. Since all of these have direct interaction with human beings, internationalization is a key factor in their design.

Technology & Society groups together the areas of privacy, digital signatures, electronic payment, demographics, disabilities, etc. This domain is probably the one where cultural differences are most obvious, especially since it is the domain that touches directly on politics, which tend to exaggerate cultural differences.

Of course, none of these areas is completely separate from the others. The internationalization activity itself is an example: although it is officially part of the User Interface domain, it has connections to almost all areas.

W3C recently opened offices in Japan. After the US (at MIT, Cambridge, MA), and Europe (at INRIA, Sophia-Antipolis/Grenoble/Paris, all in France), this is the fifth W3C site. It is hosted by Keio University in Tokyo. One of the tasks for the new team is internationalization for the “CJK” (Chinese, Japanese, Korean) languages.

2 The Web model of information exchange

Just like programs, information also benefits from being modular, and for the same reasons: you can re-use modules elsewhere and small modules are easier to maintain than monolithic documents.

The art is to find the right places to split. A large document is split into chapters and sections, but – to make it harder still – in hypertext the sections usually have to be quite small. The sections themselves can be split further along other dimensions: into different media types (text, sound, graphic, video), and into text, structure, scheduling, style and various metadata.

The split isn't completely possible yet on the Web. In particular the metadata (things like the document's age, access rights, digital signatures, subject classifications, etc.) and the scheduling are in a very early stage. In some cases, also, the split is not carried out rigorously on purpose. For example, HTML consists of text and structure intermingled. It even includes part of the metadata. The main reason for not splitting is usually to make authoring easier. Sometimes there are historical reasons.

Every module, of every type, has to be encoded as a sequence of bytes, send over the network, and decoded again at the other end. Because the Web is such a large and heterogeneous network, the encodings have to be well-known and rigorously specified, and each transferred module must be labeled with enough information to enable a receiver to decode it again.

In addition, in order to support a wider range of networked devices still, there are certain provisions for presenting alternatives and negotiating the best ones. The alternatives can be of the same type, but don't have to be. An alternative for a sound module can be a lower quality sound module, but might be a text file.

The labeling of the modules is partly handled by making them self-describing. As an example, an HTML file can have embedded information that tells an application which version of HTML is used and what the language of the text is. PNG (a format for bitmapped graphics [Boutell96]) is another example. It can contain information ranging from the size and number of colors to the copyright and date of creation.

Modules cannot be completely self-describing, however. Some amount of decoding must have occurred before any information can be extracted. A decoder must know the data-format that a stream of bytes represents before it can start interpreting any information embedded in it. Whether a byte stream is compressed or not must also be transmitted outside the stream itself.

Textual information, such as HTML files, can usually be encoded in several ways, since the formats are specified as sequences of characters, not as bytes, like image formats usually are. The encoding of characters into bytes is left undefined, meaning that it, too, must be transmitted to the receiver. This encoding, usually referred to as a *charset,* is tagged onto the type label. When HTTP is used as the transfer protocol, these labels look like MIME labels. Compare, for example

image/png (the type of an image in the PNG format), and

text/html; charset=iso8859-1 (the type and charset of a text in HTML).

The figure below shows the processing schematically. In the top left corner, an original document is shown, with its different images, text, structure, presentation, etc. The document is split into different modules, such as a text part, information about the structure and information about the presentation. Any text is encoded with a specific character encoding. Optionally, parts may be compressed as well.

The several parts are transported over the network, labeled with the character encoding, the compression used, and (not shown) the formats of each of the byte streams.

At the client side, in the lower half of the picture, the modules are decompressed, any modules labeled as text are decoded, and the different parts are combined to reconstitute the complete document.

Several variations of this general scheme are possible. For example, the decomposing only has to be done once, the document can be stored in decomposed form. There may also be a number of different decompositions, or encodings, so that a client can ask for the one it prefers. There may also be several different presentations for the same document.

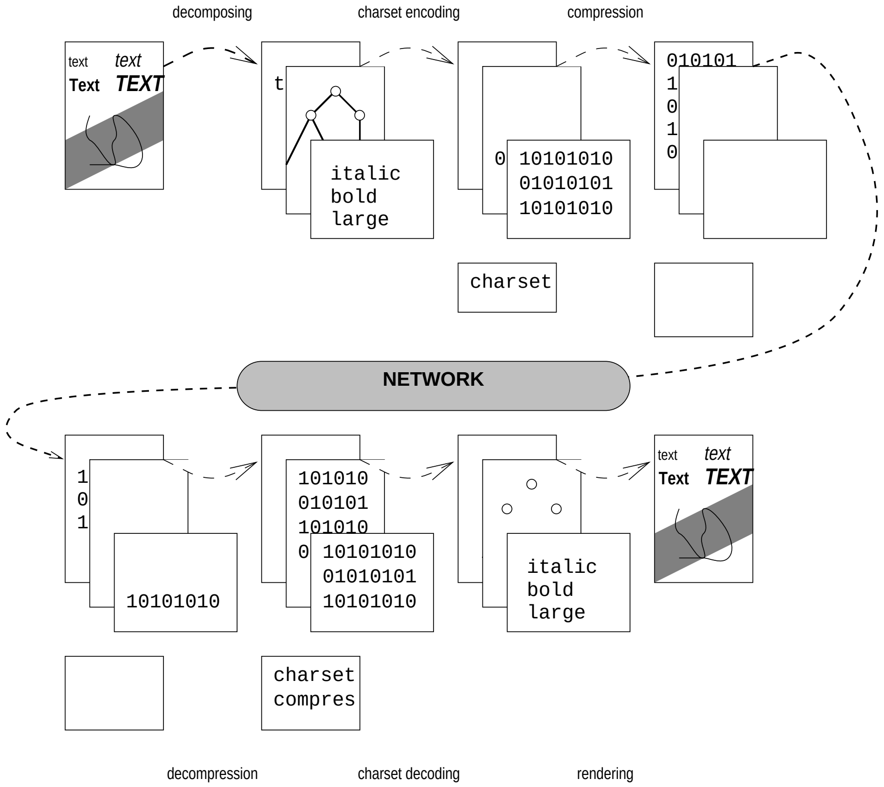
In principle, all formats in use on the Web, in so far as they contain text, should be able to contain any Unicode character. For all 'charsets' it must be possible in principle to define them in terms of how they map to Unicode.

PNG [Boutell96], a format for bitmapped graphics, is an example where text is still restricted to Latin-1. PNG was developed in 1995, before the IAB workshop on character sets [Weider97], in which it was decided that the Internet should move towards ISO-10646 as the preferred character repertoire and UTF-8 as the preferred 'charset.' It was also before RFC 2070 [Yergeau97], that set the model for HTML.

XML [Bray97] is an example of a new format, which is based completely on Unicode. As XML is expected to become the base for a number of other, derived formats, it looks like the model outlined above, and the use of Unicode, is establishing itself on the Web firmly, and with increasing speed.

3 HTML

HTML (HyperText Markup Language) is the data format for most of the textual documents on the Web. It allows an author to add some structure to a text and has room



Schematic view of the Web model of (character) data interchange.

for hyperlinks to other documents on the Web. The structure consists of a number of predefined elements such as headings (six levels), divisions, paragraphs, lists, quotes, addresses, (computer) code, bibliographic references, etc. New structures can be created by subclassing from existing ones.

The first draft of HTML 4.0 was released in July 1997. It includes all the HTML extensions proposed by RFC 2070 [Yergeau97]. Still under discussion is the markup for *Ruby,* a short annotation above one or more characters in a Chinese or Japanese text, usually to explain how the main text is pronounced, sometimes also to explain its meaning.

4 XML

XML [Bray97] is a new format for encoding any kind of structured information. It is an abstract format, in the sense that for any practical use, additional specifications have to be written. XML provides a syntax and part of the semantics, but the semantics are barely enough for a simple text document. However, for text documents that use XML, it often suffices to associate a style sheet with the document (XML will define a standard way to do that), but for other kinds of documents, semantics and presentation rules specific to the type of document will have to be defined.

An XML document looks superficially like an HTML document. Here is an example:

```
<card>
<name>Xander M. Lafitte</name>
<address>1997, World Ave</address>
<home-page href=”http://xml.org/~xander”/>
</card>
```

It has tags, just like HTML, and it even uses ‘href’ attributes for linking, again just like HTML. But in contrast to HTML, XML does not define what elements *mean.* The name ‘card’ is just a name, with no implicit semantics. When the above is used in an application, the semantics, as well as the intended presentation, will have to be specified somewhere.

One other difference with HTML is the final ‘/’ in the ‘home-page’ tag. This indicates that the element doesn’t have content. We could have written `<home-page></home-page>` instead. In XML, there are no ‘empty elements’ as such. In HTML `<img>` and `<hr>` are defined to be always empty, and cannot even have an end-tag. But in XML, an element is empty simply if it doesn’t have content. There must always be either an end-tag, or a final ‘/’.

The XML specification is still being written, but a good idea of two of its three parts can be found by reading the working drafts. The two parts that are written are the basic syntax, which all XML-derived formats will have to share, and the (hyper)linking specification, which is for use by applications that need to link several documents together. The third part will describe how to associate one or more style sheets with an XML document. Only applications that use XML for encoding books, articles, letters, etc., will have a use for this, but that may still be a considerable number. CSS will be one of the possible style sheet formats, as will DSSSL (see below), but a third choice is also in study, called DSSSL-online. This will be a style sheet language with less power than DSSSL, but more than CSS. However, it is also possible that DSSSL-online and CSS will be merged.

DSSSL (Document Style Semantics and Specification Language) is an ISO standard [DSSSL96] that defines a language for transforming and formatting SGML documents. The syntax is based on Scheme. ISO is working on modifications to

SGML (and if necessary DSSSL) to make DSSSL usable with XML as well. DSSSL has support for bi-directional as well as for vertical text.

Examples of new formats for Web-based information that are proposed and that are based on XML include: CML (Chemical Markup Language) [Murray-Rust96], CDF (Channel Definition Format) [Ellerman97], MCF (Meta Content Framework) [Guha97], PICS-NG [PICSNG97], MathML (Mathematical Markup Language) [Ion97], and others.

PICS-NG is the proposed successor to PICS [Miller96], which is a format for metadata-labels. Such labels can be stored independent of the documents they describe, even on different machines. The associated protocol (based on HTTP) allows a browser or other program to ask several servers for any information they have on a certain document. (See under Meta-data below.) Whereas PICS used a Lisp-like syntax, PICS-NG will be based on XML.

Of the above, MathML will probably be able to use style sheets, the others contain very little text and are meant to be input to specialized applications, that may or may not interact with a user.

XML is based entirely on Unicode. Its preferred (and default) encoding is UTF-8. It defines case-folding of element names in terms of the case-mapping tables of Unicode, and it allows direct reference to any Unicode character by means of its hexadecimal number (for example: &x037A;, note the ‘x’, which makes it hexadecimal). These hexadecimal character references are independent of the encoding (charset) of the document when it is sent over the network.

5 CSS

CSS1 (Cascading Style Sheets level 1) [Lie96] doesn’t do much for internationalization. It supports font sets and describes how to use them for multi-lingual documents, but for the rest it tries to stay neutral with respect to scripts, at least for horizontal writing directions. There isn’t anything that prevents right-to-left or bidirectional (bidi) text, but there isn’t anything special for it either: there are no style properties that apply only to non-western text, and there is very little guidance for implementers of how CSS1 should be applied to bidi text.

CSS2 tries to improve that situation. At the very least it will support the bidi features of HTML 4.0: set the default direction of an element or set a direction override.

Some of the style properties of CSS1 depend on the language of the text, although in CSS1 it is not explained where the language comes from. CSS2 will explain how the language of an element can be set, for example from the LANG attribute of HTML.

A few new typographical features are also proposed for CSS2, features that are uncommon for Western typography, or don’t even apply.

In Japanese text, for example, one sometimes finds a row of dots over the letters as a way of emphasizing the text. This is a simple addition to the *text-decoration* property, which in CSS1 accepts such things as underline and overline.

Also very common in Chinese and Japanese, although one finds it in Western typography as well, is to space out a text to fill a certain length. This is different from justified text, since in justified text the last line is not stretched. In Western typography, instead of spacing the letters, one often sees that the font size is increased instead.

Ruby (see ‘HTML’ above) will also have to be added to CSS. Formatting Ruby correctly is sometimes tricky, especially when there is a line break in the middle.

An effect that may not make it into CSS is the Japanese style where up to four characters are reduced to one quarter their size and put into a single square. Unicode has a number of precomposed such characters, and they will obviously be supported.

The big challenge is vertical text. The easy solution will be to require that a text is either completely vertical or completely horizontal. In that case the formatting model of CSS1 can be left essentially intact, except that it will be rotated 90 degrees for vertical text. Mixed horizontal and vertical text can then be supported once CSS can do frame-like layouts, since each frame can have its own direction. (The recently proposed positioning extensions [Furman97] for CSS have a different function.)

6 Meta-data

There is no good distinction between "data" and "meta-data." It depends on the way you use data whether you consider it meta-data or not. If you use the date of publication to search for a particular article, then that date is used as meta-data: it is a handle that helps you find the data that is your final goal.

Since computers cannot interpret documents as easily as humans can, there is a constant search for machine-readable meta-data formats that are at the same time not too unnatural for humans. The formats are usually referred to as *labels*. If it is too hard to create the meta-data labels, too few of them will be created. Since it is also impractical to have many different formats, the search is for a single one, or a small number, that can describe maybe 80% or more of the features people want to use in selecting information.

The two forms of meta-data that currently work on the Web are PICS [Miller96] and the HEAD part of an HTML document. Both methods are limited. PICS can provide labels for any resource on the Web, but it can only express things that can be represented as numbers. That includes information such as catalogue numbers, statistical information, and quality ratings along a numeric scale, but excludes such data as the names of author or publisher, keywords, and links to other URLs. The HTML HEAD can only contain data about the HTML document itself. Moreover, very little of what the HEAD can contain is sufficiently standardized.

The chosen solution is called PICS-NG, which builds on PICS, but uses an XML-derived syntax instead of the Lisp-like syntax of PICS 1.1, and which allows text strings as well as numbers. Since it contains text, it will need language codes as well.

The main part of a PICS-NG (or PICS) label is a list of keyword-value pairs. The other parts contain information to identify the label itself, including digital signatures of the people who entered the meta-data or are willing to vouch for its accuracy.

Every PICS-NG (and PICS) label refers back to the *schema* (or schemas) that defines the criteria and range of values for each part of the label. The schema is partly machine-readable itself, to allow an application such as a Web browser to present a reasonable user interface. The rest of the schema is in a natural language; it is usually an HTML document. The schema may be provided in several languages.

References

Boutell96 T.Boutell (ed). *PNG (Portable Network Graphics) specification.* October 1996 (http://www.w3.org/TR/REC-png)

Bray97 (1) T. Bray, C. M. Sperberg-McQueen. *Extensible Markup Language (XML): Part 1. Syntax*. W3C Working Draft, June 1997 (http://www.w3.org/TR/WD-xml-lang-970630)
(2) T. Bray, S. DeRose. *Extensible Markup Language (XML): Part 2. Linking.* W3C Working Draft, June 1997 (http://www.w3.org/TR/WD-xml-link-970630)

DSSSL96 Document Style Semantics and Specification Language (DSSSL), ISO/IEC 10179, 1996 (http://occam.sjf.novell.com:8080/dsssl/dsssl96)

Ellerman97 C. Ellerman. *Channel definition format (CDF).* March 1997 (http://www.w3.org/TR/NOTE-CDFsubmit.html)

Furman97 S. Furman, S. Isaacs. *Positioning HTML elements with Cascading Style Sheets.* January 1997 (http://www.w3.org/TR/WD-positioning-970131)

Guha97 R. V. Guha, T. Bray. *Meta Content Framework using XML.* June 1996 (http://www.w3.org/TR/NOTE-MCF-XML/)

Ion97 P. Ion, R. Miner (eds). *Mathematical Markup Language.* May 1997 (http://www.w3.org/pub/WWW/TR/WD-math/)

Lie96 H. W. Lie, B. Bos. *Cascading Style Sheets, level 1.* W3C Recommendation, December 1996 (http://www.w3.org/TR/REC-CSS1)

Miller96 (1) J. Miller (ed). *PICS label distribution label syntax and communication protocols, version 1.1.* W3C Recommendation, October 1996 (http://www.w3.org/TR/REC-PICS-labels-961031)
(2) J. Miller (ed). *Rating services and rating systems (and their machine readable descriptions), version 1.1.* October 1996 (http://www.w3.org/TR/REC-PICS-services-961031)

Murray-Rust96 P. Murray-Rust. *Chemical markup language (CML) version 1.0.* 1996 (http://www.venus.co.uk/omf/cml/)

PICSNG97 PICS Next Generation (draft in preparation) expected Autumn 1997 (http://www.w3.org/PICS/NG/)

Weider97 C. Weider et al. *The report of the IAB character set workshop held 29 February –1 March, 1996.* RFC 2130, April 1997 (ftp://ds.internic.net/rfc/rfc2130.txt) W3C took part in this workshop.

Yergeau97 F. Yergeau, G. Nicol, G. Adams, and M. Dürst, *Internationalization of the HyperText Markup Language.* RFC 2070. January 1997 (ftp://ds.internic.net/rfc/rfc2070.txt)