The design of an International Web Font extension for Cascading Style Sheets

Chris Lilley, W3C
www.w3.org/people/chris

Abstract

A new W3C specification extends the font mechanisms in CSS1 to permit improved client-side matching of fonts, enable font synthesis and progressive rendering, and allow font download over the Web in response to font requests in a style sheet. This allows web page authors to describe, or provide links to, fonts for any Unicode characters, and addresses many shortcomings of the proprietary 'font face' HTML tag.

1 The problem

This paper describes a new way to use fonts on the Web. Of course, fonts are used on the Web all the time, otherwise nothing would be visible! Given certain restrictions, such as a total lack of typographic control and limitation to the Basic Latin and Latin-1 Supplement characters, the current system works surprisingly well.

The use of the ISO 8859-1 (Latin-1) character encoding for document transmission, rather than ISO 646-IRV (or US-ASCII) was undoubtedly a factor in the early success of the Web, particularly in Europe; it permitted some common European languages to be easily expressed and gave an unaccustomed cross-platform portability. Documents could readily be transferred between highly heterogeneous systems and viewed and printed at wildly different display resolutions and capabilities.

The limitations became apparent when people naturally wanted to have more typographic control, or to use other languages. Early methods of transcending these limitations had some success, but also introduced other problems which are now becoming increasingly burdensome.

1.1 Fancy Layout

Typographic control was achieved by convoluted use of tables, proprietary HTML extensions, and transparent 'spacer' images, and by setting type in other applications and including a picture of it on the Web page. The benefit of this approach was that the Web became a more attractively designed place, particularly if the reader used the same computer, operating system, browser, and screen size as the document author.

The downside was that people using different models of computer, different browsers, and different default fonts found that the assumptions made by page designers did not hold; text would be too big or too small, text would be partially visible, overlap, or only be visible when scrolling, images would not line up, pages would not print well. Pages started to include instructions to resize the window to a particular width, or to switch to a different browser, to force the reader to more closely approximate the assumptions of the document authors.

The use of images of text for headings was particularly troublesome; the contrast between headings and body text would vary wildly across systems, text would be illegible when printed, and search engines would fail to retrieve relevant pages because the important headings contained no text so that pages were not indexed correctly.

Visually disabled users, meanwhile, were finding that Web pages had transformed from simple, marked up text documents to content-free, complex descriptions of a visual layout which were inaccessible to plain text or speech-synthesis browsers; the information had been replaced by a picture of a document.

1.2 More languages

Support for non-Latin scripts was simply achieved by setting the default font (in a preferences file, registry, X resource, hacking the program or by whatever method) to be a font which used a different font encoding vector and which was intended for the display of a different script. By this means, Web pages could appear to be in Polish, Greek, Russian and similar languages with a small character repertoire. The benefit of this was that different communities were able to access information in their own languages, provided that documents were only circulated within that community and given that the ability to correctly display Latin-1 documents had been lost.

The drawback was that indexing, searching, spell-checking, and cross-platform portability had been lost; even on the same platform, the reader had to have a font with exactly the right name or the page would display as gibberish.

The MIME type for HTML is `text/html`, and all text types can have an optional parameter which indicates the *character encoding* used to transmit the document. For historical reasons this is termed the `charset` parameter. For example, a Russian page could be labelled `text/html; charset=iso-8859-8`. Unfortunately most pages were not so labeled; partly because within a closed community using a single character encoding the sytsem appeared to work without it, and partly because early browsers treated the entire MIME type as an opaque string and would thus break when confronted with a correctly labelled page which included a charset parameter.

The downside was that the meaning of the bytes in the file (as assigned by the document author) was incorrectly labelled so indexing engines, speech synthesisers and anyone not using the precise font used by the author (or another one with exactly the same font encoding vector) would see random meaningless strings of characters. Also, each document could only contain languages which used the same script.

Later, browsers had 'easy' pull-down menus to let users choose from various character encodings (on the assumption that most content was unlabelled), this tedious process was often described as 'choosing the font'. The concepts of *character encoding* and *font encoding* were being muddled, mainly because the concept of the document character set (see later) was not fully developed. This increased the difficulty for the reader. For example, it was common for users of Cyrillic documents to have two versions of each font, one using ISO 8859-8 as a font encoding vector and one using KOI-8 as the font encoding vector. The reader would switch between these manually, on a per-document basis, depending on the character encoding that had been used to transmit the document. A popular Russian browser, AMSD Ariadna, has five different Cyrillic encodings to select from. This is clearly a poor way to proceed; if the content is labelled and the browser can re-encode a font on the fly, only one font is needed and the reader does not need to manually select anything. User-driven font selection on a per-document basis is an attempt to mask the symptoms rather than treat the problem.

2 Font face

The `font` element is an HTML extension, now incorporated into HTML 3.2, which was originally for altering the color and the size of text. It soon acquired a new attribute, `face`, to select a particular font by name and this attribute is included in Transitional HTML 4.0. In some ways this was an advance; fonts could be applied to particular HTML elements rather than the entire document, and greater stylistic control was also possible.

The downside was that if there was no font of that name on a particular reader's system, and the document had relied on the font encoding vector to generate the appearance of non-standard characters, the meaning would be

lost. This problem was exploited in one round of the browser wars - pages appeared which used `font face` to set text in *WinDings*, a picture font, thus making it unreadable on browsers which implemented this extension.

The arrival of the font face extension prompted a resurgence of sites in other languages, because multiple fonts could be used in one document and thus the appearance of multiple scripts could be simulated. By producing free fonts with carefully selected ligatures, even more complex scripts such as Devanagari could be simulated in this way. Readers who had not downloaded the fonts, or platforms which were unable to use fonts in the supplied format, were unable to view the content at all.

This situation hilights a fundamental difference between publishing on paper and publishing on the Web. On paper, it does not matter which keystrokes were used to prooduce a particular glyph. Indeed it is common for designers to re-arrange fonts to get a keyboard layourt that suits them. There is no way to tell, looking at the printed result, what keystrokes produced it. On the web. in contrast, it matters a great deal.

3 International HTML

The Web uses HTML as its primary media type. HTML is an application of SGML, and each document type in SGML has a particular Document Character Set in which all computations and manipulations are performed. For HTML since version 2.0, this single Document Character Set has been ISO 10646 (although version 2 of HTML was restricted to the first 256 characters, in other words Latin-1). With HTML 4.0 the document character set was stated to not ionly be ISO 10646, but Unicode - which has the same code points but also implies additional functionality such as the Unicode bidirectional embedding algorithm.

A consequence of this is that, regardless of the c*haracter encoding* used to transmit the document, all numeric character references refer to the Document Character set not the character encoding used to transmit the document. For example, Κ represents the character at code point 922 decimal, U+039A, greek capital letter kappa. This NCR can be inserted in any HTML, regardless of the character encoding that happens to be in use. For example, the document might be in Japanese and sent using shift-jis; it can still contain any Unicode character.

Conceptually, the incoming document is converted from a stream of bytes into a stream of Unicode characters, using the character encoding information. It is then converted into a sequence of glyphs, using the font encoding vector information. Although both of these mappings can be 1:1 they need not be.

Both HTML and CSS rely on this distinction, which is well described in the ISO Technical Report on the Character Glyph Model, TR 15285.

In addition, HTML 4.0 includes the lang attribute, introduced in RFC 17666 and more tightly defined in RFC 2077, to specify the human language used by that element and its children.[1] This information can be used by the style sheet author to perform language specific processing. For example, a page containing passages in Chinese and in Japanese may well use the same CJK Unified Ideographs in both sections; the different language codes `zh-tw` and `ja-jp` may be used by the styule sheet implementation to select the appropriate Traditional Chinese and Japanese fonts, respectively.

4 International XML

XML is a language for writing document types; a Web-enabled successor to SGML. Whilst individual XML applications will invent their own element names, certain things are common to all XML applications. One of these is the use of Unicode as a document character set (XML does not need or use SGML declarations). Another is the `xml:lang` attribute which exactly mirrors the semantics and properties of the equivalent HTML 4.0 `lang` attribute.

5 Cascading Style Sheets

5.1 Style sheets, CSS, XSL

Style sheets are a way of gathering together all the presentational information in a document, leaving the actual document to concentrate on describing the structure. Style sheets can be external (in another file) or internal (inside the HTML file). For example, if a document has ten subheadings, the stylesheet could simply indicate that they are all to be in 16/18 point Helvetica Oblique rather than having some of this information embedded in the HTML and thus repeated ten times.

The term Cascading Style Sheets (CSS) refers to a style sheet format developed first at CERN, around 1994, and later at W3C. It is aimed particularly at online use, with HTML documents. The cascade refers to the way multiple style sheets - those linked to the document by the document author, the reader's personal stylesheets, and the browser internal default stylesheet - are

1.This attribute contains a primary code and a (possibly empty) series of subcodes, separated by hyphens. For example, de, `ja-jp`, `en-nz`, `no-nynorsk`, `fr-ca`. If the primary code is two letters, it is an ISO 639 language abbreviation; if a subcode has two letters, it is an ISO 3166 country code.

combined in a precise order to contribute towards the end result. Level 1 of the CSS specification, finalized in December 1996, is a W3C Recommendation; the specification for Level 2 is nearing completion. CSS is designed to work well with HTML and can also be used with XML.

The eXtensible Stylesheet Language, XSL, is another style sheet format which aims to include the functionality of both DSSSL, the ISO style sheet language, and CSS2. A W3C working group was formed in January 1998 to develop XSL, which is specifically targetted at comples, data-rich XML documens which require extensive reordering and computation for display.

5.2 Font features of CSS1

The CSS1 specification allows the setting of various font properties on HTML elements. The properties include family name (eg Times, Arial), weight (eg normal, bold), style (eg italic) and size (eg 12pt). It allows these properties to be inherited and then modified by child elements. For example, various font properties may be set on a paragraph, and a bold element within that paragraph can be made to have the same font - but in a heavier weight - by simply altering the font-weight property.

In CSS1, fonts are assumed to be present on the client system and are identified solely by name. Several choices of fonts may be listed and are tried in order until a font is found that can display the required characters. If a font is available to the client that is a close stylistic match to the requested font but has a different name, it is not possible for a CSS1 implementation to select it. Generic font families such as ‘serif’ and ‘script’ are available as fallbacks if none of the listed fonts are available.

Because CSS honors the character-glyph model, it is *not* possible to apply, say, the Symbol font onto Latin text to get a semblance of Greek. The font will fail to match and the next font in the sequence will be used. This property allows document authors to specify several fonts for a single element, and the appropriate one will be used automatically. For example

```
P {font-family: Palatino, Bukinst, “Heisei Mincho W3”, serif }
```

Palatino covers Basic Latin and Latin-1 Supplement, Mincho covers some of the CJK Unified Ideographs using Japanese glyphs and Bukinst covers Cyrillic. Bukinst is placed before the mincho font because some Japanese fonts also contain glyphs for cyrillic and we want the ones from Bukinst to be used in preference. In a mixed French, Japanese and Russian document, the correct font will (if available) be selected for each character without the necessity of special markup around each run of characters from a particular script. The last font in the list, serif, is a generic fallback font which is defined to exist by the CSS specification.

6 Beyond name matching

The first additional feature provided by the Web Fonts extension in CSS2 is intelligent matching.

Intelligent matching involves using more information than just the name of the requested font to select an existing, accessible font that is the closest match in appearance to the requested font.The metrics might not match, resulting in different line breaks. The matching information includes information about the kind of font (textual or pictorial), the style of serifs (or analogous stroke terminations), weight, cap height, x height (where these can be defined), typographic ascent and descent, slant of vertical strokes, and so on. This is essentially an open-ended and extensible set of font characteristics, initially derived from Latin typography.

Problems come from inconsistent definitions of basic typographical terms on different platforms or with different font formats. These differences have always been there, of course, but are less obtrusive in a paper-oriented, non-distributed environment where document generation and document rendering happen on the same computer and where manual verification and correction of the layout is possible before generating a fixed printed result.

Panose-1 is a widely used classification scheme based on defined measurements of particular Latin characters. A Panose-1 measurement can be used to select a font which is similar in character to another font, provided both have Panose-1 measurements. Panose-1 could also be used for some non-Latin scripts such as Greek and Cyrillic, but the measurements to do so have not been defined.

Panose-2 is a proposed more extensive classification scheme which can relate different measurements made on glyphs from different scripts. It may be useful in the future for automatically selecting similar fonts for different scripts.

Other problems are due to characteristics being specific to particular scripts. For example, height of flat-topped unaccented lowercase letters (the x-height), as shown in Figure 1, is a very useful indicator of the style of a Latin font, particularly when expressed as a ratio of the x-height to the height of flat-topped capital letters (the cap height). Fonts of dissimilar x-height will look very different.

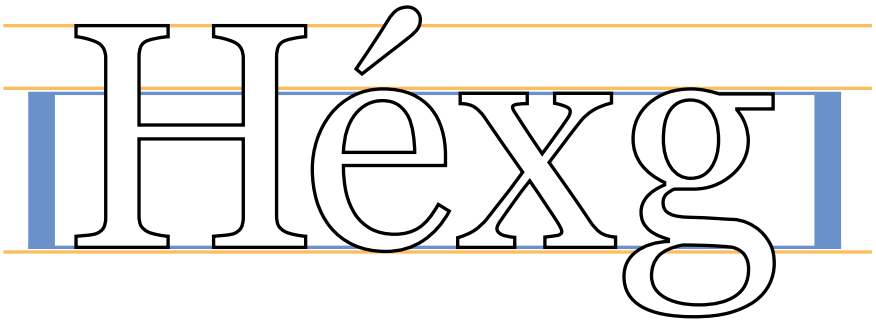

Figure 1 : x-height is a useful characteristic for bicameral fonts

Most scripts are unicameral - they have only a single case. They are often set at a height midway between the x-height and cap-height, as shown in Figure 2. Dropping the x-height and cap-height properties on the grounds that they only apply to some scripts, however, would penalise those scripts for which these characteristics are important. Another alternative is to assign the x-height and cap-height the same values, so that the ratio (ie, 1) can be compared with that of bicameral scripts.

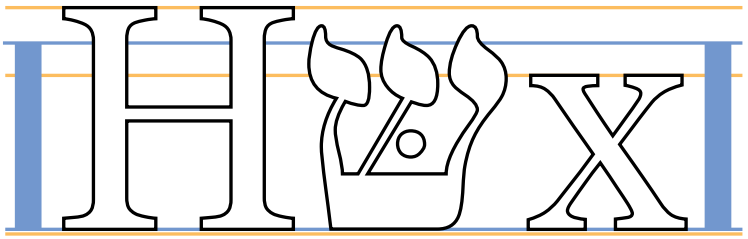

Figure 2 : Glyph from a unicameral script

This allows Latin fonts with low x-height to be chosen for use with Arabic, and Latin fonts with high x-height for use with Hebrew. The best solution is for such matching to be done by the stylesheet designer, using similar or contrasting faces that work well together from a design standpoint. The automatic matching is then a fallback if the requested fonts cannot be used for whatever reason.

Another characteristic that is script dependent is alignment. Latin scripts have a clearly defined baseline, with descenders. Other scripts have no descenders,

or are aligned on a centerline - for example, CJK Unified Ideographs - or a top line (hanging baseline) - for example, North Indian scripts. The Web Font extension allows all these alignments to be defined on a per-font basis.

7 Font Synthesis

More than selecting an existing font, synthesis means creating a new font with a particular appearance, on the fly.

The aim is to create a font which is not only a close match in appearance, but also matches the width metrics of the requested font so that the line breaks occur in the same places. Information required for font synthesis is similar to that required for matching, but generally needs to be more precise. In particular, synthesis requires accurate width metrics and character to glyph substitution and position information if the layout characteristics of the specified font are to be preserved.

As implemented in the current draft, this implies that only fonts which have a 1:1 character to glyph mapping can be succesfully synthesised, because there is currently no way to describe individual glyphs or multiple, differently sized glyphs for the same character. AFII glyph IDs are a possibility, but are not very user freindly. There also comes a point where the cost of downloading extensive font description information, to be used as input to a font synthesis engine, approaches the cost of downloading the actual font.

8 Font Download

The Web Font extension, now fully integrated into CSS2, allows URLs to be added to the style sheet which point to fonts. There are techniques such as site locking, digital signatures, and format transformation which can be used to protect the intellectual property rights of the font designers; these techniques are not addressed here.

The stylesheet can also indicate, on a per-font bases, the range of Unicode characters for which it has some glyphs. Most fonts have sparse coverage of Unicode. This property is used to determine whether a font might have glyphs and thus whether to download it or search it.

Other metainformation about the font can be added, such as the size of the design grid, the position of the various baselines (low baseline for Latin, Greek and Cyrillic; central baseline for Ideographic scripts and top baseline for Indic scripts), the x-height and ca-height, Panose-1 number, and so on. Other descriptors can readilly be added to better describe fonts for scripts that are presently not covered so well, such as Arabic.

Multiple font formats are in use on many different platforms, so in consequence HTTP content negotiation is required for downloadable fonts. The same stylesheet can point to multiple fonts in different formats, indicating which format is available at which URL; this is particularly handy given that the first two implementations of Web Fonts chose different font formats.

9 Example

This example is for font downloading, no information is provided to enable font synthesis or matching. It defines a composite font split over three files.

```
<STYLE>
  @font-face {
  font-family: Excelsior;
  src: Excelsior Roman, url(http://site/er) font/opentype
  unicode-range: U+00xx /* Latin-1 */
  }
  @font-face {
  font-family: Excelsior;
  src: Excelsior EastA Roman, url(http://site/ear) font/intellifont;
  unicode-range: U+01xx-022x /* Latin Extended A and B */
  }
  @font-face {
  font-family: Excelsior;
  src: Excelsior Cyrillic Upright, url(http://site/ecu) font/truedoc;
  unicode-range: U+04xx /* Cyrillic */
  }
</STYLE>
```

10 Conclusion

A font architecture has been produced which allows the salient properties of fonts to be described in stylesheets and applied to HTML and XML documents, which are considered to be a collection of Unicode characters. This solution is cleaner and more scalable than preceeding, ad-hoc solutions. It is a first step towards true multilingual typography on the Web.

11 References

Association for Font Information Exchange (AFII)
http://www.isc.rit.edu/~afii/

Cascading Style Sheets, level 1
http://www.w3.org/TR/REC-CSS1

Cascading Style Sheets, level 2
http://www.w3.org/TR/WD-CSS2

Character Set considered Harmful
http://www.w3.org/MarkUp/html-spec/charset-harmful.html

HTTP Content Negotiation
http://gewis.win.tue.nl/~koen/conneg/
draft-ietf-http-negotiation-01.html

Panose-2
http://www.w3.org/pub/WWW/Fonts/Panose/pan2.html

Web Fonts extension to CSS
http://www.w3.org/TR/WD-font.html

ISO 639:1988 (E/F) - Code for the representation of names of languages - The International Organization for Standardization, 1st edition, 1988 17 pages .

ISO 3166:1988 (E/F) - Codes for the representation of names of countries - The International Organization for Standardization, 3rd edition, 1988-08-15.