L2/10-389

Reply to L2/10-327: “Comments on L2/10-280, ‘Proposal to Add Variation Sequences... ’”

This document consists of the reply from 2010-08-09 given on the Unicore mailing list, regarding the mail from Peter constable there from 2010-08-08, which is documented as L2/10-237.

Date: Mon, 9 Aug 2010 03:23:38 +0200
From: Karl Pentzlin <karl-pentzlin@acssoft.de>
To: Peter Constable <petercon@microsoft.com>
CC: "unicore@unicode.org" <unicore@unicode.org>
Subject: Re: comments on L2/10-280, "Proposal to Add Variation Sequences..."

Am Sonntag, 8. August 2010 um 20:45 schrieb Peter Constable:

>> ... In particular, A.1
>> attempts to make the case that alternatives to variation sequences
>> are inadequate.

No, I only considered *one* special alternative inadequate:
the user being required to select between different versions of a font
to get some special characters displayed according to their needs.

>> The argumentation, in overview, runs as follows:
>> ...
>> 3. Many Internet sites are maintained by non-pros who do
>> not have skill needed to use server-side technologies that could
>> provide the needed, non-widely-used fonts to the reader’s system
>>
>> It appears that the proposer may not be aware that such
>> server-side functionality is being integrated directly into the
>> latest versions of W3C recommendations ...

I *am* aware of such technologies (and mentioned it in the proposal),
but I also am aware of the fact that a technology will not
automatically come into use only because it exists. Non-pros (not all)
will continue to ignore such technologies just because they are non-pros.

>> 4. OpenType provides general mechanism to select alternate
>> glyphs within a font, but not standard conventions to specify a particular glyph.
>>
>> OpenType specifies clearly-enough a mechanism ...
[to specify culture-specific glyphs]

There is a mechanism to select a particular glyph *within the font*,
but afaik no mechanism that selects a glyph *according to an external
character specification* besides considering a character code itself.

E.g., I do not know of a general standard OpenType statement which *explicitly* says:
"use a glyph for the Latin small a which employs a two-storey form",
in a way independent from the internal tables of a specific font, and
independent from all language-specific or context-specific things.
(Something which is addressed by variation sequences.)

>> ... that is not difficult to implement

Difficulty is in the eye of the beholder. There are people who
consider anything which smells like programming as beyond their
grasp, or who simple hate it.
Some of these may be excellent font designers.

>> 7. It is harder for a font to support a number of cultures
>> [each of which may involve multiple distinctive glyph forms] than
>> it is to support a mechanism such as variation sequences that
>> allows alternate glyphs to be specified on an individual basis
>> (e.g. if there were, say 50 variation sequences to be supported).

>> ... it assumes that only the variation sequence mechanism can support
>> that glyph-by-glyph approach, implying that OpenType Layout
>> mechanisms cannot. This is untrue, however.

Sorry, but this is true (as long as "only" refers to the two
alternatives "variation sequences" and "OpenType Layout").

The main misunderstanding is:

OpenType provides an *abstract mechanism* to select between several
glyphs which are contained in a font for arbitrary characters.

Variation sequences provide a *concrete list* of variants
(with their own representative glyphs which constitute a guideline
for the glyphs provided by the font for a specific variant, like
the representative glyph of a character gives a guideline for the
font designer).

Having said this: "Variation sequences" and "OpenType mechanisms"
are no alternatives which possibly exclude each other.
They are orthogonal to each other. They can coexist or even complement
each other (by introducing OpenType mechanisms which make use of
variation sequences).

>> ... In fact, a set of OpenType features with precisely this functionality
>> are defined: see
>> http://www.microsoft.com/typography/otspec/features_ae.htm#cv01-cv99

Wrong:
1.) I do not see a *globally defined* feature to get e.g. a two-storey a
for the character "a",
In fact, I see not a single concrete entry like "cv12 selects the
small-form-derived form for the Latin capital letter eng, if
available in the font, and is under no circumstances used for
something else."
2.) The entry states under "function": "A font may have stylistic-variant glyphs ..."
This is exactly what (in my view) variation selectors are *not* for.

>> Granted, the association of individual cvXX features to
>> particular glyph forms is not conventionalized and may vary from
>> one font to another.

This *exactly* is the problem I address with variation sequences.
(Please note I do not refer to cvxx, but to the complete available set
of OpenType features.)

The main achievements of variation sequences are:

- The variants of a character are well documented in a standard,
  easily found e.g. by font designers who have a guideline what kind
  of glyphs they are recommended to provide in an universal font.
  (This alone can be achieved of course without referring to any Unicode
  mechanism, or to any OpenType mechanism, if such one tied to a concrete
  list of variants would be introduced.)

- The variants are accessible in plain text, by a mechanism which
  already exists in Unicode and which is established for ideographic
  characters.
  (Without extending the existing Unicode mechanism, this could be
  achieved only by disunification of the affected characters, which
  would cause other problems in most cases.)
  The use in plain text is appropriate where the text is not linked to
  any language or locale (and therefore cannot be algorithmically handled
  by font mechanisms linked to a language or locale), as e.g.:
  - multilingual text
  - plain text entries in databases, when the language/locale of the
    retrieving environment is not determined
  Also, variation selectors are adequate to override the variant selected
  by language/locale for single characters (this can be regarded as a
  special case of multilingual text).

Also, it is to be noted that variation sequences does in no way interfere
with existing mechanisms to select glyphs within fonts (e.g. OpenType
mechanisms).

>> *8. Locale data could contain applicable variation sequences.*
>>
>> *This entails ... that text processing applications will start requesting the*
>> *relevant locale data at authoring or display time. ...*
>> *Note also that fonts must still be updated to support this.*

This is admitted. Doing such (in my proposal, I called it the "implicit application" of variation sequences) in fact is a new mechanism (which I did not request; I only hinted that such mechanisms will possibly emerge). The fact that any new thing has to be supported before being usable is trivial (this applies e.g. also to any newly encoded character).

However, such a new mechanism has several advantages, as outlined in my proposal, e.g.:

>> *This entails that locales must be updated rather than fonts*

I see this in fact as a big advantage. Locales are defined in a central place, maintained by experts.
This means releasing font designers from the burden to understand locales to its full extent, to observe and follow all changes in these locales, and to find out which of the some 50 variants are to be applied in each of the several hundred locales which exist, and to repeat this over and over in the same way for several fonts.

>> 10. Expecting users to select culture-specific fonts is no
>> different than an era of custom-encoded fonts.

I did say "something like", not "no different than".

>> it [another variant than intended] is legible to all users.

Legible, but still wrong. Also, they may be users which only know
"their" form (imagine persons who only hardly have learnt to read);
in this case it is not even legible.

>> General comments:
>>
>> I am given the impression that the author of this proposal
>> started with a solution and then explored ways to argue in favour
>> of that solution. But in forming his arguments he has not taken a
>> comprehensive view of the overall issues. ...

Admittedly, I do myself hard to comment on this. It is not directly ad hominem. but too near to it.

However, this proposal has a history.
I had two problems to solve:

- The decision of the UTC #123 not to disunify the Florin sign from the Latin small letter f with hook, made me think whether there is another appropriate solution to address that problem.

- When researching historical Latin alphabets from the 19th and 20th century, I came over a lot of deviating uppercase forms for existing lowercase letters, where the uppercase forms alone would justify the encoding of new letters, which then would have no lowercase counterpart. As the Unicode stability policies do not allow to encode lowercase counterparts when found later (which cannot be excluded as long as not all such alphabets are uncovered), I tried to avoid this.
  Searching on possibilities, I found the mechanism of variation sequences.
  I considered them a solution (see L2/10-229, section 3.1), because:
  - equivalent uppercase deviations already exists (as was documented in L2/08-034R), which then could be handled in the same way.
  - Unicode 5.2 states in section 4.16:
    "The existence of a variation sequence does not preclude the
    later encoding of a new character with distinct semantics and a
    similar or overlapping range of glyphs";
    Thus, a later found case pair which shares the uppercase glyph
    is not precluded from encoding retaining the case mapping.

>> ... that the proposed solution could be used with existing
>> software and fonts

which is true.

The existing Unicode standard (5.2, section 16.4) says:
- "... if the variation sequence is not supported, the
  variation selector should be invisible and ignored."
- "The standardization or support of a particular variation sequence

does not limit the set of glyphs that can be used to represent the base
character alone".

According to the first citing, you can clutter every text with variation selectors even now,
and if your rendering system shows any glyphs for the selector alone,
it is in error even now.
This is far more than you can expect for newly encoded characters,
as their use in existing text is erroneous any usually leads to the
display of "non-availability glyphs" or similar.

According to the second citing, even for supported variation sequences
it is legal to ignore them; a font or rendering system which will not
support it shows another glyph of the character, as it would show a
default glyph for a character it does not support at all.
Like in any case, you need a font that supports your character.
Support of variation sequence only ensures that you get the correct
variant if/when it is available at all.

>> ... that it would require comparable updates to software ...

In fact, I doubt this: If software can support the existing variation
sequences for mathematical characters, there shall be no principal
issues to support additional variants up to the same level.

Implementing things like the "implicit application" in fact imply
comparable updates. But note that I have neither required this mechanism,
nor stated its existence as a precondition. I only have described it as
a possible mechanism which would make variation sequences even more useful.

>> ... that his solution would also
>> require changes to existing content in order to realize any
>> enhancement in rendering that content ...
>> ... it is effective only when content is
>> authored to include a variation sequence in each and every
>> instance of a relevant character for a given culture. ...

I have *explicitly* stated in the proposal that this is *not* the case.

(Having stated this, the concerns about input methods are void.
For multilingual texts where variation selectors in fact are to be
entered, there is nothing new in principle over the input of entering
the correct variants for mathematical characters. However, such texts
usually do require special input methods anyway.)

- Karl Pentzlin