L2/02-362

Date: October 28, 2002
From: Michael Kaplan
Subject: Comments on DUTR#29 for UTC#93

In looking at DUTR#29 ("Text Boundaries"), I was very happy to see the bulk of the feedback from UTC #92 meeting applied appropriately. In looking at the DUTR and Section 5.15 of the book, I see only a few real differences:

1. The introduction and use of "grapheme cluster" as a term meant to refer to a "user character".
2. The use of regular expressions in explaining the default boundaries.
3. The addition of testing data.

I have a few minor concerns about the above three issues, and then one larger concern that I think deserves some discussion in the meeting. First the minor issues:

Item #1) I think there is too much time spent trying to define something already defined in the glossary – indeed, if every usage of the term "grapheme cluster" is going to have half a page of information then the glossary itself loses purpose, and the UTC and the Editorial Committee are then forced to maintain a definition in several places. This bulk of this text should be shortened/removed and instead the glossary should be referenced, rather than forcing a future clarity update to the definition to change yet another UTR (or in this case UAX).

Item #2) I actually think this is a good idea, certainly better than what is in the book now if an implementation will use regular expressions. If it will not, then they are no worse off. However, I do think that the text makes assumptions about the potential implementer's understanding of regular expressions PRIOR to referencing UTR #18 (Unicode Regular Expression Guidelines). This text should be changed so the UTR is referenced before anything relating to regular expressions is mentioned (the text as it stands goes on in section 1.1 at length prior to mentioning UTR#18, a very disorienting approach for people who do not understand regular expressions (yet).

Item #3) Another good idea, I think. But there is perhaps not enough emphasis on the fact that this is only a default implementation throughout the DUTR, which will lead me into my larger concern, below:

The issues that remove section 5.15 from the 4.0 book and place it in a UAX for **non-normative** information are worrisome. I wonder whether this text could not simply be included in the book, as is, with the test data on the book's CD. The fact that the book contains so much less of the actual information needed to implement the standard makes the book less useful to implementers.

Beyond that, the fact is that people perceive a UAX as a part of the standard, and therefore tend to assume normativity even when it does not, in fact, exist. I wonder whether the definition of UAXes themselves needs modification:

> ***A Unicode Standard Annex (UAX)*** forms an integral part of the Unicode Standard, but is published as a separate document. Note that conformance to a version of the Unicode Standard includes conformance to its Unicode Standard Annexes. The version number of a UAX document corresponds to the version number of the Unicode Standard at the last point that the UAX document was updated.

By emphasizing conformance in the definition, it is easy to assume that they must conform to the UAX, even if the UAX does not have a conformance requirement. This is a larger problem then this one UAX but it seems particularly worrisome here since an implementer is left to wonder whether they must support ***at least*** the default suggested here, unless they have something more sophisticated. Is this the intention? Because it's an easy conclusion for an implementer to draw from the text as it stands, in part because the flaw in the UAX definition as it applies to informative information and in part because of the way the DUTR itself handles the issue.

At a minimum, the UTR needs to explicitly say what is required. With the UAX definition as it stands, it is not enough to say its informative; if it is conformant to Unicode 4.0 to do no text boundary work whatsoever, this must be explicitly stated. Not doing so leaves implementers wondering what they are required to implement which will lead to excessive confusion and worry on what to implement.