Characters, Glyphs and Beyond

Tereza Haralambous and Yannis Haralambous

Abstract

The distinction between characters and glyphs is a fundamental issue of computing. This talk aims in giving a new definition of these notions. We first review and comment the definitions given in various standards. Then we give and explain our own definitions. We consider that the Unicode character model is lacunary and formulate a proposal for adding supplementary information and obtaining thus "rich Unicode characters." We illustrate our arguments with many examples, taken from various writing systems.

Keywords: character, glyph, language, writing system

The distinction between characters and glyphs is currently a very popular issue. The complexity of this issue is, in some sense, related to the fact that computer systems have been build by engineers not very proficient in linguistics, and interested only in the English language. Exploring non-latin writing systems one realizes what has not been clear from the beginning: that modelizing written language is not a trivial task, and that it is fundamental to all exchange and processing of textual information.

Let us start the exploration of this universe by giving some definitions of the terms we are using. Let us see how the terms "character" and "glyph" are defined.

According to ISO 9541 [6] released in 1991, a "glyph" is "a recognizable abstract graphic symbol which is independent of any specific design," while a "glyph image" is "an image of a glyph, as obtained from a glyph representation diplayed on a presentation surface," where "glyph representation" is "the glyph shape and glyph metrics associated with a specific glyph in a font resource." We may argue if this distinction between "abstract glyph" and "concrete glyph" is necessary, but this is how ISO 9541 defines these.

According to W3C (quoting "A Character Model for the World Wide Web" by Martin Drst and others [2]), a character is "the smallest component of written language that has semantic values; refers to the abstract meaning and/or shape." We find this definition quite vague since everything we perceive may or may not have semantic value, depending on our culture, context and even mood... We all know that Unicode is full of inconsistencies, because of its requirement to be compatible with legacy encodings. Has this definition been made to encompass Unicode weaknesses, and is therefore voluntarily vague?

W3C uses the ISO 9541 definition of glyph, probably to be consistent with the only available standard on "Font information interchange." The definition of "glyph" in Unicode is slightly different: a glyph is "a shape that a character can have when rendered or displayed." Notice two things: first, the fact that the definition of glyph is based on the one of character, so if the first one is vague, the second one is even more vague; secondly, the fact that there is no distinction anymore between "glyph" and "glyph image," as in ISO 9541. We are now talking about shapes, and nothing else. There is an illustration in the Unicode book which clearly shows glyphs corresponding to the same character, in different fonts. This shows that Unicode's definition of a glyph is rather the one of "glyph image" in ISO 9541.

For whatever it is worth, the PDF Reference 1.4 (2001) [1], defines a character as "an abstract symbol," whereas a glyph is "a specific graphical rendering of a character." Once again we have a vague definition of character and a definition of glyph relying on it. After all, what is an "abstract symbol"? It doesn't give us a clue about why "A" is an "abstract symbol," and not "fi."

Now let us give our own definition of character and glyph [4, 5]. First of all, we believe that the best way to define these notions is going from glyph to character and not the other way around, as W3C and PDF are doing it.

For us, a glyph is "the image of a typographical sign." You may object why we use the term "typographical" in our definition. Well, typography has been a first modelization of human writing. Books are based on this modelization (even if in some cultures books are still written by hand) and books are the carriors of human culture. Computers are based on this modelization. Typographical signs are uniform, at least in the frame of a given book, or of a given page. In such a narrow frame, the differences between typographical signs are microscopic, this is not the case for hand writing. Of course if for a given writing system there has never been any typographical tradition, then we must amend our definition to something like: "a glyph is part of the image of written text, not too big and not too small, so that the given writing system can be obtained by an optimal sequence of these images, arranged in a regular way." This apparently complex definition is better explained as: "let us first try to modelize the given written system as typography would have done, and then let us take as glyphs the ones of our model." But these kinds of writing systems are quite exceptional, and they are not the main topic of our talk.

So let us suppose that the writing systems we care about are those who had already a typographical tradition, be it a short one. Typographers are highly intelligent creatures and have subdivided the image of text into small pieces which are not too big (in Latin script that would be "words") and not too small (in Latin script that would be pieces of letters) but just optimal in size and quantity (in Latin script that would be letters). We have based our definition of glyph on their work.

What is then a character? Let us realize that when we see a glyph, we are *interpreting* it. If it belongs to a writing system we know, then we have some specific knowledge about it: how it is pronounced, how it gets combined with other glyphs, its numerical value, etc. If we are know proficient with the given writing system we can maybe still recognize it as belonging to that system, but no more. Sometimes we cannot do even that. In that case our interpretation of the glyph focuses on its geometrical properties: is it a triangle, a circle, does it resemble to that or that glyph we know?

Interpretation leeds to *description*. How do we describe a glyph? Take the glyph "A." Some may say it is an open triangle with a bar in the middle, other will say it is a "Latin letter A," other will say it is the mathematical "for all" operator which has been inversed. Many descriptions can be given, but only a few are interesting to computing.

Furthermore, a glyph description may fit to more than one glyphs. In fact, in most cases, it will be appropriate to an infinity of glyphs, since the images sharing a few properties can be infinitely diverse. We can say that a description is an equivalence class of glyphs: two glyphs will be equivalent if they fit to a given description.

Our definition of a character is: "a character is an equivalence class of glyphs, based on a simple, linguistic or logical description."

So if we say "`LATIN CAPITAL LETTER A`," then we describe a class of glyphs which can be interpreted as letter capital A in the Latin writing system. This description is purely linguistic.

When we say "simple," we mean that the description should be optimal in length: not too short, not too long. When we say "linguistic or logical" we refer to the fact that characters can belong to writing system for languages, but also to notation systems (as for music, industrial design, trafic signs, mathematics, etc.).

If we apply this definition very strictly, then quite a few Unicode characters are not qualified to be characters. For example "`SPACE`" is hardly a glyph, since it is an empty image. The description "`SPACE`" is even less a character since it is neither linguistic nor logical, but graphical. But "`SPACE`" could also be defined as the "word separation method" in European writing systems, which would qualify it as a character, since it is a purely linguistic description.

What about "`THIN SPACE`" (which is Unicode character 2009)? This one is more hard to defend. One could say that it is part of a notation system: the repertoire of lead types. In the frame of this notation system, it has some logic, so it would make sense to call it a character.

One way to test if a glyph equivalence class qualifies as a character is to bypass graphical representation of language and to think of what happens to these glyphs in systems like voice synthesis. "`SPACE`" is absolutely essential in voice synthesis, since without it, text would be impossible to understand. But "`THIN SPACE`" makes no sense whatsoever in voice synthesis. So there is a legitimate doubt about its character essence.

Now let us see how characters and glyphs are used in computing. As we see on the drawing, humans use keyboards to input text in computers. Keyboards refer to characters, but when we push on keystrokes what we see on the screen is already a glyph. We see glyphs on screen, but what we store in a document are characters. What we send through the Web are characters. People reading our messages or Web pages read glyphs. They interprete these glyphs, and in their minds, one could argue that glyphs become characters again, but let us not go as far as that... What happens in a human's mind is probably beyond the complexities of characters, glyphs and the like.

It is clear that going from characters to glyphs and vice-versa is a fondamental part of human-computer interaction. Still it seems that people working on human-computer interaction take it for granted and prefer to focus on higher interface elements, such as menus, dialogs, and the like.

Our argument is that the bipolarity *character/glyph* is *not* sufficient for modelizing text. We will give some examples of cases where the information needed is located somewhere inbetween those two concepts. After that we will give suggestions on ways of including this information, and examples of case where this is already possible.

Our first example is about older European writing systems, Gaelic:

An Milleónaiḋe

Tuairim is oċt mile ṡliġe siar ó Ḃaile an Loċa, baile beag aol-ḃán atá neaḋuiġṫe go deas idir réiḋ-ċnoc íseal agus faill ġéar-áird, tá séipéal lom i lár roilige ar ṡleasaiḃ an ċnuic. Ar aġaiḋ an ċnuic sin anonn atáid na tiġṫe. Aiṫneóċair tiġ beag an ṗuist ar an sanas uaine atá ós cionn na fuinneóige. Aiṫneóċair árus an té is taoiseaċ ar ṁuinntir na háite, ar a aoirde agus ar an ḃfál sigirlíni atá ina ṫimċeall. Lastall den droiċead, ag bun na faille, atá bóṫar ag gaḃáil siar i dtreó an tsléiḃe. Sa

and Gothic:

„Gewiß", schrie der Senator, dessen Gesicht und faltiger Nacken rot angelaufen waren. „Was sollte ich sonst wohl benutzen? Aber Gleichmann & Busse sind aus purer Geldgier auf den Gedanken gekommen, die Schiffsmaschinen hier am Grasbrook zu bauen. Und ich habe einige hundert Arbeiter zur halben Passage von Hull nach Hamburg gebracht."

„Scotch", sagte er zu dem herbeigeeilten Kellner, „but make it a big one. — It's to my future bride."

„Aber Herr Overholt!" sagte der Kellner, „Scotch? Am frühen Nachmittag?"

„Sie haben ein gutes Herz, Lulubelle", sagte Oskar.

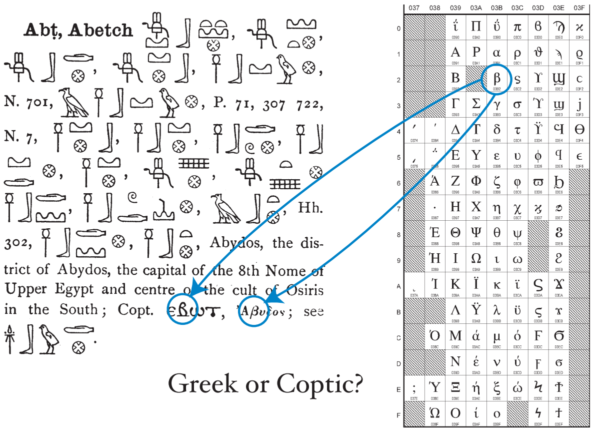
„Ich war selbst so ein Kind", sagte Lulubelle, „ich weiß, was ein Schilling für sie bedeutet."

Until about fifty years ago, Irish language was only typeset in Gaelic script. In a book combining Irish with words from Latin alphabet languages, one would switch to Latin script for these. Similarly Gothic, which is also called "broken script" was the de facto writing system for German. In Germany it is was even called "Deutsche Schrift," that is "German writing system." In a text mixing German, French, Greek and Russian, German would be written in Gothic, French in Roman, Greek in Greek, and Russian in Cyrillic script. In this context, mapping between writing system and language would be one-to-one. And hence, descriptions like "`GAELIC LETTER B`," or "`GOTHIC LETTER ES-ZET`" could be considered linguistic, since these alphabets were de facto alphabets of the languages we mentionned.

Just like Greek and Cyrillic alphabets are represented by specific Unicode characters, one would expect Gaelic and Gothic alphabets to be represented in Unicode. Well, they are not. Probably because nowadays Gothic is not used in Germany and Gaelic is not used in Ireland, besides for decorative purposes. And being "decorative" is precisely a feature of glyphs and not of characters.

So what we really need is an extra property of Latin alphabet characters saying that a given string is actually in Gothic or Gaelic script, whenever the script has a linguistic connonation.

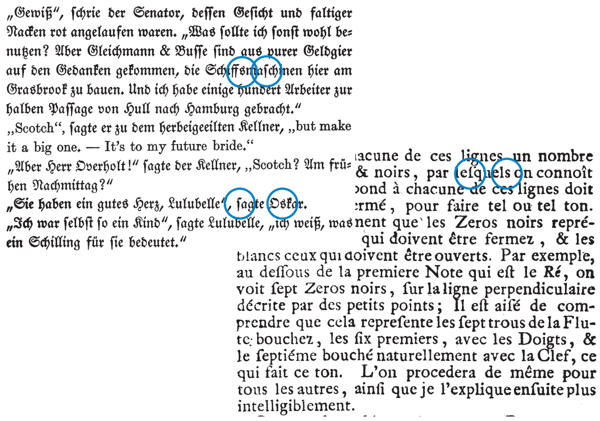
Another example, the Coptic script:



Greek or Coptic?

This one is provided in Unicode, but it uses the same character positions as Greek. Nevertheless Greek and Coptic are quite different, and each one is used to denote a single language. Well understood, both can appear in the same text. Once again one needs an extra propertu saying that a character is Coptic or in Greek script.

Now let's go into more subtle examples. On the figure below, one can see a French and a German text using the long and round s letters:

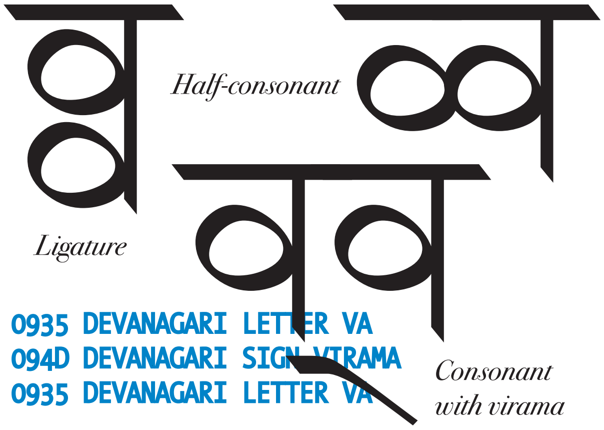


Long s is indeed a Unicode character. This makes sense in the German example, since words like "Schiffsmaschinen" on line 4 use the round s to show that there are composite (in this case: "Schiff" = boat and "Maschine" = machine). But in the French sample, the letter s follows a very strict rule: when it is initial and medial, it is long, when it is final, it is round. This behaviour is totally predictible and hence can be obtained simply by giving a contextual rule, without any special characters. This also means that we could switch from old French to modern French and back without any loss fo information, which is not the case for German. It would be interesting to attach the property of being "mandatory" to the long s Unicode character, and use this property for German but not for French.

Similarly, Latin alphabet ligatures like "fi," "ffi," "ffl," etc. are used without any further consideration in the French language, are never used in Turkish language, and are used selectively in the German language, where their absence implies that the word is composite:

In this case, it is the absence of a ligature which plays a linguistic role. In Unicode, this is obtained by using the ZWNJ ("zero width non joiner") character. For the German language it would be interesting to systematically use ZWNJ between word components: this would not only break ligatures, but also optimize hyphenation. A word like "transformation" would use a ZWNJ character in German language, but not in French or English. In other words we would need a ZWNJ character specialized to a given set of languages, and not active in other ones.

Talking about ligatures, more interesting examples can be found in Indic scripts. There are three ways of writing "vva" in Devanagari:



as a vertical ligature, as a half-consonant followed by a complete consonant, as a consonant with virama followed by a consonant. To obtain these three representations of the same grammatical sentence one would use the same Unicode character sequence, namely a `VA` followed by a `VIRAMA` followed by another `VA` with implicit `A` vowel. But these three ways of writing the same sequence fo characters are semantically different: the first one is rather used in Sanskrit texts, the second is used in normal typography, and the third is a trade-off whenever the second is not feasible. It is the level of sophistication that changes, and this level is often related to the cultural level of the text and the quality level of the printer. Hence we need a way to complement the character information by a property giving the ligature level which is expected, or maybe a preferential order of possible ligature levels.

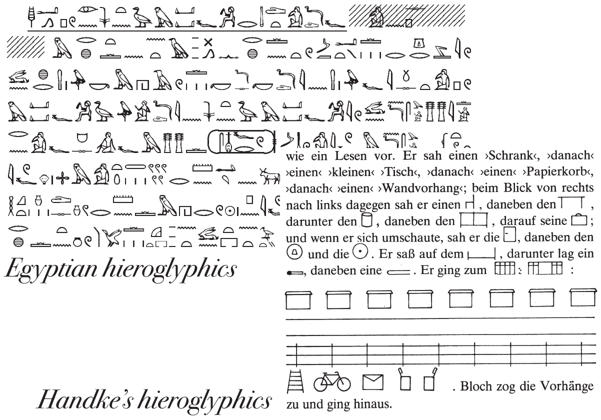
In Arabic scripts we also have a big number of ligatures, but these are more font and calligraphic style-dependent than the Indic ones. But there is a different interesting phenomenon encountered in the Quran: we have a glyph, nowadays called "swash kaf" which was known for many centuries under the name "the kaf of impiety":

This happened because this letter was mainly used in the word "al-kufr," which means "the impious." This is not accidental: the big size of this letter has served the calligrapher to warn the reader about one of the biggest sins in Islam: impiety. By using a different glyph, the calligrapher was able to add meta-information to the text, and we are talking about a text the corruption of which could be punished by death.

Later on the kaf of impiety became a letter of the Sindhi version of the Arabic alphabet, and thru this came into Unicode. But still, the Sindhi `ARABIC LETTER SWASH KAF` (Unicode `06AA`) is usually much shorter than the real kaf of impiety. It is located somewhere inbetween the regular kaf and the kaf of impiety.

The Japanese reader of this paper is probably very familiar with another problem of Unicode related to characters and glyphs: kanji variants, or, more generally, consequences of the unification of Chinese, Japanese, Korean and Vietnamese ideographs. We all known how controversial the decision of Unicode Consortium has been, to unify, in some cases, ideographs which looked alike but were not the same, or not to unify ideographs, which were considered the same, in some other cases. Not only we need to attach to a Unicode ideographic character the language of the context, which may be a series of languages in a preferential order, for example when we write Chinese poetry which can be read in Chinese or in Japanese. But we also need to attach graphical variants of ideographs which may be needed for personal names (as in the case of our dear Takahashi-san), or other special uses. It is a pity to see Unicode having taken so drastic and unpopular decisions just to stay in the 16-bit range, and then, a few years later, finally give up the 16-bit range...

Even more difficult to normalize are Egyptian hieroglyphics. Hieroglyphs are not connected, so that we can easily divide them into glyphs, and hence into characters. They are quite well standardized, at least for the main period of existence of the Egyptian Empire. But they can come in several sizes and be combined in ways much more flexible than Korean Hangul. And despite standardization, one could expect, or at least assume, similar problems to those of ideographs. And hieroglyphs were not only used in Ancient Egypt, they are even used today, as in the book of the famous German author Peter Handke "The Goalkeeper's Fear of the Penalty Kick":



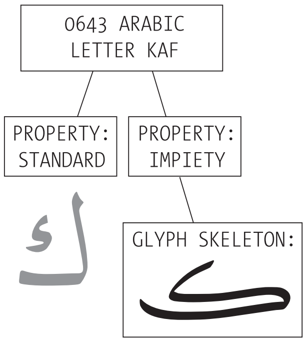
Egyptian hieroglyphics

Handke's hieroglyphics

Of course there are no limits to the imagination of the author, and at the same time these are clearly characters and should be encoded. Will Unicode unify them with the ancient Egyptian hieroglyphics? And what about other pictograms?

Some of the examples we have given can be handled by simpling adding a property to the character: choosing between Coptic and Greek, Gaelic or Gothic and Roman, etc. In the case of the long s or of Indic ligatures a simple classification of these would be sufficient to obtain a list of properties.

But in other cases the additional information we need may be more complex. For example, in the case of the kaf of impiety it would be a glyph, or at least a glyph skeleton, which would be necessary:

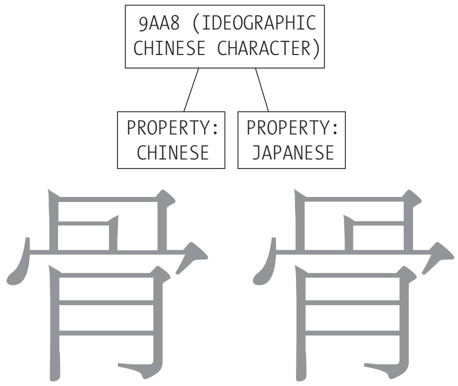


In the case of Chinese ideograph variants, one may use a glyph description appropriate for Chinese character synthesis. The same could be applied to Egyptian hieroglyphs, we would need a Hieroglyph Synthesis Engine, but maybe the market for such a program would be more reduced than the one for Chinese Character Synthesis.

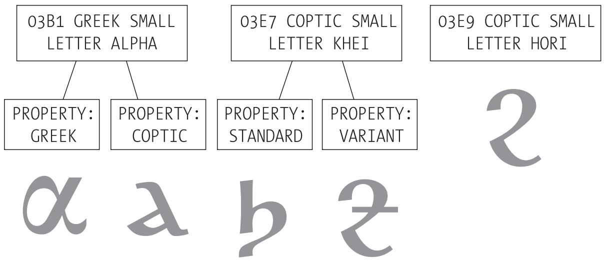
Adding properties, sub-properties, etc. to a character would be best done in a hierarchical structure. It could result in a tree, the leaves of which could be glyphs, depending on various contexts. Depending on its particular needs, software could ignore that tree and use only the character information, stay at higher levels of the tree and use only simple properties (like this is done in OpenType and AAT fonts), or go down until the leaves and fetch a glyph, or a glyph skeleton, or a glyph description.

Examples:

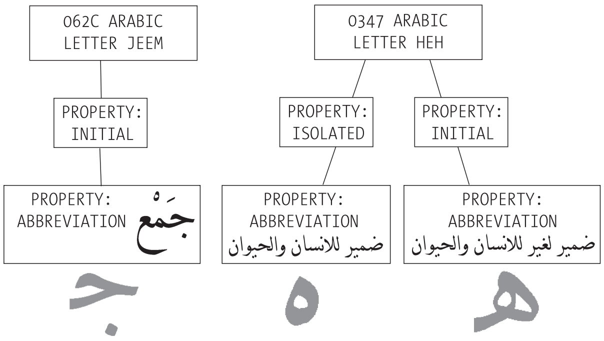
1. An ideographic character with different glyphs in Chinese and Japanese:



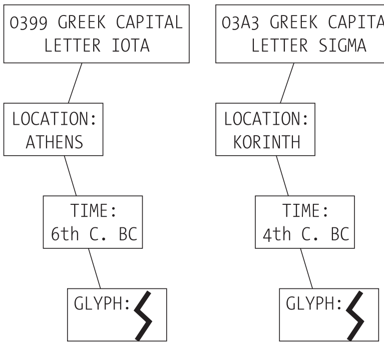
3. A Unicode character that can be a Greek or a Coptic letter, next to a definitely Coptic letter which can be standard or variant (the variant is to distinguish the letter *khei* from the *hori*) :



3. An Arabic letter whose contextual form carries extra information:

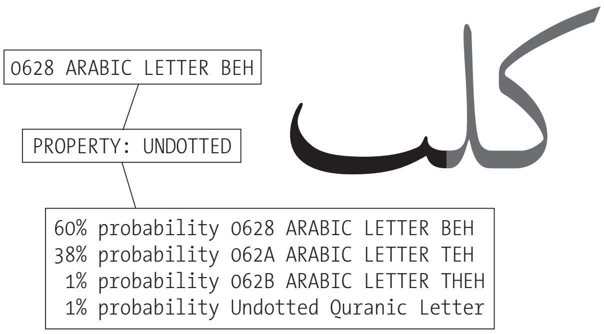


4. A Greek character, whose glyph depends on location and time, or, inversely, a glyph whose character depends on location and time:



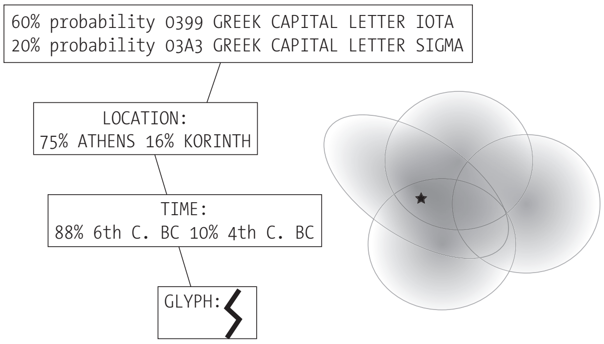
Such an approach has already been implemented more-or-less successfully into SVG [3]: one has glyph elements which can carry several layers of information, including glyph descriptions, and which have Unicode correspondances. In such way that

There are cases where the interpretation of a glyph as a character can be probabilistic. For example, in the first form of Arabic writing there were no dots to disambiguate letters *beh*, *teh*, *theh*, *yeh*, *nun*. When a manuscript is read we have to deduce from the context to which character belongs a given glyph. But this interpretation can have multiple solutions which we can ponder with probabilities:

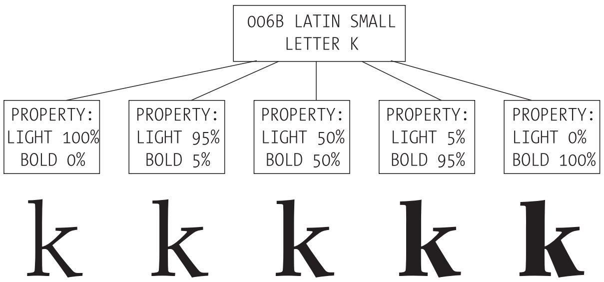


This gives us a *probabilistic "rich" Unicode character*.

We can re-consider the ancient Greek glyph above in a probabilistic perspective:



But we are talking about characters, as if they were isolated. We said at the beginning of this paper that characters could be considered as the *interpretation of glyphs*. But interpretation is always subjective and depends highly on context. On the figures we can see cases where the interpretation is not clear or may vary from one context to another:



On the figure above we are progressively boldening a glyph. When does it starts to be called "bold"? We know that typographers have many intermediate steps between light and bold, but in typesetting bold is often used for emphasizing, and hence must be clearly identifiable. In the PostScript type 1 fonts there is a flag called "ForceBold," which shows the necessity to clearly identify boldness. But how do we quantify a subjective criterion on a continuous property of glyphs? Compare the two samples below:

Probability of interpretation depends on context.
Is this bold?

Probability of interpretation depends on context.
Is this bold?

We can use gradients of boldness: depending on our typesetting configuration we can establish that the bold counterpart of a given light font must have boldness in a given range, weither this is mesured by Panose or some other measuring method.

The following line of text looks Hebrew at first sight:

it's o pity i won't

but if we see it in its context:

WHAT YOU HAVE
JUST READ WAS:
it's o pity i won't
BUT ACTUALLY
I MEANT TO SAY:
it's a pity i won't

then we realize that it is actually a Latin alphabet Hebrew simulation font, and that the line could be interpreted as: it's o pity i won't.

On the figure below we see fonts intermediate between Gothic and Roman[1]:

Une écriture bien française

And what about this typeface?

Das ist echte Deutsche Schrift

A German text typeset in such a font could be clearly considered as Gothic, a French text could be considered as Roman, maybe weird Roman but Roman however. But if German is to be mixed with French, how will these fonts be interpreted by readers? Here we need to quantify "gothicity," or "gaelicity."

More generally, on the figures below we see two extreme ways of mixing writing systems. One can try to keep an homogeneous image (lower text) or one may try to keep a traditional touch for each one of them, so that they are more clearly identified (upper text):

Διαφοροποιητικὴ μέθοδος ἐναρμόνισης. Λίγα γαλλικά: un peu de français dans une police Didot, μετὰ λίγα ρωσσικά: Все шрифты классифицируют по группам, καὶ ἀπὸ τὶς μὴ εὐρωπαϊκὲς γλῶσσες λίγα ἀρμενικά: **Գոհութիւն Աստուծոյ**, λίγα γεωργιανά: მხედრული ასოები, καὶ οὕτω καθ᾽ ἑξῆς...

Ἑνοποιητικὴ μέθοδος ἐναρμόνισης. Λίγα γαλλικά: un peu de français dans la même police Times, μετὰ λίγα ρωσσικά: Все шрифты классифицируют по группам, καὶ ἀπὸ τὶς μὴ εὐρωπαϊκὲς γλῶσσες λίγα ἀρμενικά: Գոհութիւն Աստուծոյ, λίγα γεωργιανά: მხედრული ასოები, καὶ οὕτω καθ᾽ ἑξῆς...

The choice between the two approaches carries strong semantics, and suggests a property of characters of being "separatist" or "unifying," "sensitive to local typographical traditions" or "globalizing."

[1] The first line is in a French script font, the third line is in authentic German Fraktur, but the second line is in an intermediate between Gothic and Roman. This font is called *Alsace Lorraine*, and has been designed by Harold Lohner.

The result of rendering characters with such properties will depend on the context: the more the writing systems are, and the closer they are one to each other, the more one must rely stronger on the traditional style of each script, to keep them apart.

Let me summarize the contents of this paper:

What we propose is to add optional additional information to Unicode characters in a text. We consider this additional information to be indispensable for textual information exchange. It may be implemented as higher level mark up (SVG is an example of such an attempt) or as binary data following Unicode characters and using escape sequences, we have not discussed implementation issues. It should be hierarchical, extensible, as general as simple properties, or as complex as Chinese ideograph descriptions, glyph descriptions, ranges of glyph instances in Multiple Master fonts, Metatype code, or some new type of dynamic glyph skeleton which would generalize character synthesis.

We are convinced that the character and glyph bipolarity is not sufficient for optimal textual information exchange and storage, and we are suggesting a way to go beyond that. A research team at ENST Bretagne, including two Ph.D. students are working on these issues and we are hoping to have exciting new results in the forthcoming years.

Bibliography

[1] Adobe Systems. *PDF Reference: Version 1.4.* Addison-Wesley, 3rd edition, December 2001 `http://partners.adobe.com/asn/acrobat/docs/File_Format_Specifications/P%DFReference.zip`.

[2] Martin Drst, Franois Yergeau, Richard Ishida, Misha Wolf, Tex Texin, *Character Model for the World Wide Web*, W3C Working Draft, August 2003, `http://www.w3.org/TR/charmod/`.

[3] Jon Ferraiolo, Jun Fujisawa, and Dean Jackson (eds.). *Scalable Vector Graphics (SVG) 1.1 Specification.* W3C, January 2003 `http://www.w3.org/TR/SVG11/`.

[4] Yannis Haralambous. "Unicode et typographie: un amour impossible." *Document numérique*, vol. 6, number 3-4/2002 "Unicode, criture du monde?," pp. 105-137, 2002 `http://omega.enstb.org/yannis/pdf/docnum.pdf`.

[5] Yannis Haralambous. *Fontes et codages.* O'Reilly, Paris, 2004.

[6] ISO/IEC. *Information Technology — Font Information Interchange, Glyph Shape Representation ISO/IEC 9541-3:1994(E)*, May 1994 `http://www.iso.ch/iso/fr/CatalogueDetailPage.CatalogueDetail?CSNUMBER=1%7280`.