ELECTRONIC PUBLISHING, VOL. 6(3), 999 (SEPTEMBER 1993)

The Design of a Unicode Font

CHARLES BIGELOW
Department of Computer Science
Stanford University
Stanford, California 94305, USA
email: bigelow@cs.stanford.edu

KRIS HOLMES
Bigelow & Holmes Inc.
P.O. Box 1299
Menlo Park, California 94026, USA
email: D2782@applelink.apple.com

The international scope of computing, digital information interchange, and electronic publishing has created a need for world-wide character encoding standards. Unicode is a comprehensive standard designed to meet such a need. To be readable by humans, character codes require fonts that provide visual images - glyphs - corresponding to the codes. The design of a font developed to provide a portion of the Unicode standard is described and discussed.

KEY WORDS Unicode international character standard type design Lucida

1 INTRODUCTION

Unicode is a world-wide character encoding standard developed by the Unicode Consortium, a group of companies and institutions with interests in international text-encoding and computing applications. The Unicode standard was developed to provide solutions to the most common and serious problems encountered in multi-lingual computer programs, including "overloading of the font mechanism when encoding characters, and use of multiple, inconsistent character codes caused by conflicting national character standards." [1]

The Unicode standard distinguishes between "characters" and "glyphs" in the following way: "Characters reside only in the machine, as strings in memory or on disk, in the backing store. The Unicode standard deals only with character codes. In contrast to characters, glyphs appear on the screen or paper as particular representations of one or more backing store characters. A repertoire of glyphs comprises a font." [1] Hence, in Unicode terms, the relationship between glyphs and characters is not a one-to-one mapping. Latin Capital letter 'A', for example, is encoded as Unicode character 0041 (hexadecimal), but the visual glyph representing that character in a particular instance on screen or paper could be a Times Roman 'A' or a Helvetica 'A' or a Courier 'A', or any number of other 'A' forms, depending on the type style chosen for the text. Even within a single font, more than one glyph may represent a single character. In the Arabic alphabet, for example, the written form of a letter depends on context, and the shape of the glyph that renders a given character differs according to whether the character occurs in initial, medial, final, or isolated position in a text string. Unicode 1.0 does not encode these allographic variations (but see "Revisions and Updates" below).

Despite the usefulness of Unicode's careful distinction between character and glyph, in this paper we often use the term "character" rather more loosely, and more in keeping with tradition and ordinary language, to mean an element of a script or a unit of a writing

0894–3982/93/030999–01$5.50
©1993 by John Wiley & Sons, Ltd.
Received 1 September 1993
Revised 15 December, 1993

system. By "script" or "writing system" we mean a graphical representation of language. Hence, a character in this looser sense is any significant unit of writing, e.g. a letter of the Latin alphabet, an ideograph (more properly, logograph) of the Chinese writing system, a numeral, a punctuation mark, a symbol, a unit of space. Prior to the invention of typography, characters were rendered "on-the-fly" by the hand of a writer and the kind of logical distinction that Unicode draws between character and glyph was not as sharply defined. Handwriting usually contains context sensitive variations, and in any writing system, some rules of variation are explicitly formalized, while others are unconscious. Accumulated contextual glyphic variations are what transformed capitals into lower-case, for example, and turned roman into italic. The invention of typography resulted in unvarying, prefabricated, mass-produced glyphs in the form of printing types. The separation of text composition from letterform creation, and the need for bureaucratic indexing, storage, and retrieval of typographic characters encouraged a conceptual distinction between the abstract notion of a character or letter and the tangible realization of it as a piece of type. Because the preponderance of characters composed, displayed, and printed by computer are typographic, the implicit disjunction between abstract character and concrete glyph has achieved a logical formulation in the Unicode standard.

The Unicode standard provides a uniform, fixed width, 16-bit character identification and encoding scheme that covers the majority of writing systems used in the world today. Version 1.0 of the standard encodes approximately 28,000 characters, of which some 3,000 are alphabetic letters and diacritical marks for European, Indic, and Asian languages, 1,000 are various symbols and graphic elements, and 24,000 are ideographic (logographic), syllabic, and phonetic characters used in Chinese, Japanese, and Korean scripts [1] [2].

Unicode is a major step toward achieving portability of documents across the boundaries of different nations, languages, and computer systems. The comprehensive scope of Unicode presents a challenge to the developers of operating systems, but at least three recently developed s: Microsoft "Windows NT 3.1", AT&T Bell Laboratories "Plan 9"[3], and Apple "QuickDraw GX"[4].

Character Sets Large and Small

Unicode, with its thousands of characters, presents a challenge to Western type designers unaccustomed to large characte saccustomed to large characterofets. For most of the past 500 years, Latin fonts were founded with character sets sufficient to compose text in one or a few European languages. Although some early fonts were designed with many variant characters and ligatures to emulate the rich variety of scribal handwriting, the triumph of printing in Europe was partly due to the efficiency and economy of text composition from small sets of alphabetic characters. Although the *textura* typeface used by Gutenberg in the 42-line Bible of 1455-56, the first printed book in Europe, included more than 250 characters, and the *humanistica cursiva* typeface cut by Francesco Griffo for the Aldine Virgil of 1501, the first book printed in italic, included more than 200 characters, character sets became smaller in later fonts, to reduce the costs of cutting, founding, composing, and distributing type.

Typical character sets contained capital and lower-case letters (with accents and diacritics), ligatures, numerals, punctuation, and sometimes small capitals. In the earliest European manual of printing and typefounding, *Mechanick Exercises on the Whole Art of Printing*, published in 1683, Joseph Moxon illustrates upper and lower compositor's cases

containing some 140 characters [5]. In the first volume of his *Manuel Typographique*, published in 1764, Pierre Simon Fournier lists 155 characters (of which 32 are small capitals) for a roman font [6, vol. 1]. In their classic compendium, *Typographical Printing Surfaces*, published in 1916, LeGros and Grant list 154 characters (including 29 small capitals) in the English bill of fount for a roman face [7].

The change from manual to mechanical composition technology at the end of the 19th century did not alter the average size of character sets, nor did the change from mechanical to photographic composition in the 1960s. The Monotype matrix case contained 225 to 272 characters, usually divided between two styles of one family, e.g. roman and italic, and the "duplex" matrix sets for Linotype were of equivalent size [8]. In photocomposition, the number of characters in film fonts or matrix sets varied widely from device to device, though the number of characters in a single type style ranged roughly from 88 to 128 [9]. The typewriter, a 19th century invention that has remained popular through most of the 20th century, likewise offers rather small character sets. The IBM Selectric typewriter, popular world-wide from the 1960s through the 1980s, provided 88 characters in each "type-ball" .

In the 1980s, personal computers and laser printers, marketed world-wide, encouraged larger, 8-bit encoded sets of 256 characters suitable for multiple languages. Many of these character sets, e.g. Apple Macintosh [10], Microsoft Windows [11], and Adobe PostScript [12]. were based on the ISO Latin-1 character standard [12], with the addition of assorted Greek letters and miscellaneous symbols. Like Unicode, "character encoding" standards such as ASCII and ISO 8859 map characters to computer-readable numerical codes, in order to provide a standardized basis for information interchange. These standards also encode the so-called "control characters" like 'Carriage-Return' (ASCII decimal 13), which make no visible mark but which do signify some meaningful element in an electronic text file or stream.

The inclusion of non-alphabetic symbols and non-Latin letters in these 8-bit character sets required font developers to decide whether the assorted symbols and non-Latin letters should be style-specific or generic. Like many designers, we prefer style-specific glyphs, believing that one of the advantages to creating new fonts for electronic publishing is the opportunity to coordinate pi characters, non-Latin letters. and Latin characters. Some font vendors, however, have favored generic pi characters because they provide a degree of standardization and reduce the costs of font development.

The question of style-specific vs. style-generic symbols seems minor for the ISO Latin 8-bit character sets, where only a dozen or so characters are involved, but when we move beyond the limited domain of the Latin alphabet defined in ISO Latin-1, into the vast realm of world scripts encoded by Unicode, the question of whether or not to coordinate the designs of all the heterogeneous alphabets, signs, symbols, and characters becomes crucial. A global design decision affects the appearance of thousands of glyphs.

2 DESIGN GOALS

An International System Font

Our initial motivation in designing a Unicode font was to provide a standardized set of glyphs that could be used as a default core font for different operating systems and languages. Within one font, we wanted the different alphabets and symbols to maintain a single design theme. If the same basic font were used in different Unicode-compatible

systems, its width metrics and visual appearance would be stable and predictable, and documents of all kinds, whether brief electronic mail messages or complex, multi-lingual technical reports, would be easily transferable without typographic distortion.

Unicode is a character encoding standard, not a glyph standard. There is no requirement that a glyph mapped to the code point of a Unicode character should have a particular design, nor that glyphs in one Unicode subset share design features with those of another. Therefore, an easy way to develop a Unicode font is to arbitrarily assemble disparate fonts, say a Latin font of one design, a Greek of another, a Cyrillic of a third, math operators of a fourth, and so on, all of somewhat different styles, weights, widths, proportions, and shapes. Two different fonts assembled in this haphazard fashion could have identical Unicode character sets but very different glyphs.

A problem with this kind of haphazard development is that the typographic features of documents, programming windows, screen displays, and other text-based images will not be preserved when transported between systems using different fonts. Any typographic element could change, from type style, to text line-endings, to page breaks, to document length, to window size, and so on.

Our goal was to eliminate these kinds of problems by creating a default font that would provide stability and reliability across systems.

Harmonized Typeface Design

Our second motivation was to test the concept of harmonized design of international character sets. By "harmonization", we mean that the basic weights and alignments of disparate alphabets are regularized and tuned to work together, so that their inessential differences are minimized, but their essential, meaningful differences preserved. In this way, the "noise" of insignificant design artifacts and historical accidents is filtered out, leaving the "signal" of significant character features amplified in comparison. Within a harmonized font, when text changes from Latin to Cyrillic, or from Greek to Hebrew, or when mathematical expressions or other symbols are introduced into text, the visual size, weight, and rhythm of the characters should not appear to change, should not jar or distract the reader, but the basic character shapes should nevertheless be distinctive and immediately recognizable.

While harmonization seems like a desirable goal, there is no practical way to test the typographical effectiveness of this kind of unified design without an actual font of integrated, harmonized scripts and symbols. Formal design ideas must be seen before they can be evaluated. The Unicode standard provides codings, names, and minimal graphical depictions of characters, but offers no design guidelines for integrating the disparate sets of characters. Indeed, there are no standard guidelines anywhere for harmonized design, other than the examples of certain typeface families.

Among the few typeface design that could serve as models for design harmonization is Jan van Krimpen's intriguing Romulus family, which was originally designed with alphabets of Roman, Greek, Chancery cursive, and sans-serif, though never released in its entirety [13]. Another interesting example is Nikolai Kudryashov's Encyclopedia family, which includes Latin, Greek, and Cyrillic alphabets, in both seriffed and sans-serif styles, as well as special signs and symbols. It was originally designed for a Soviet encyclopedia, but is little known outside of Russia [14]. Stanley Morison's Times New Roman [15] [16], Max Miedinger's Helvetica [17], Adrian Frutiger's Univers [17], and Eric Gill's Gill

Sans [16] are notable examples of typefaces originally designed only in the Latin alphabet, but extended to Greek and Cyrillic alphabets as they became internationally popular. The design of a Unicode font, however, takes us even beyond the character sets of these designs.

Comparative Legibility

The design of non-Latin scripts also presents an opportunity to explore the "universals" of legibility. Most legibility research in Europe and America has focused on Latin typefaces, in which factors like weight, serifs, capitalization, and style are tested and compared [18] [19]. Although Latinate typographers and like-minded readers are often ready to offer opinions about the relative legibility of seriffed vs. sans-serif, or of roman vs. italic typefaces, such distinctions are minor in comparison to the pronounced differences in shape and detail between the Latin alphabet and alphabets like Hebrew, Arabic, Armenian, and Devanagari, scripts that nevertheless provide levels of legibility roughly equivalent to that of Latin. As designers, we would like to know how the "rules" that govern legibility in non-Latin scripts compare to the rules for Latin typefaces. Shimron and Navon, for example, report a significant difference in the spatial distribution of distinctive features in the Roman and Hebrew alphabets [20]. This is the kind of study that we would like to see available for all non-Latin scripts, but in the absence of more such cross-cultural studies of legibility, design experimentation offers an alternative way of studying cross-cultural aspects of legibility, albeit only in an intuitive and anecdotal sense.

History and Culture

Finally, we were motivated by the artistic enjoyment to be derived from designing the wide variety of glyphs found in the Unicode standard. It is an opportunity to contemplate the past, and take part in the future development, of thousands of abstract graphical forms that have been developed through several millennia by generations of scribes in different civilizations. To design such a font is a way to study and appreciate, on a microcosmic scale, the manifold variety of literate culture and history. To study and reproduce the design of a Greek letter, or a Coptic letter derived from an Egyptian hieroglyph, or any one of thousands of other written forms, is to understand what a colleague saw in his or her mind's eye when fashioning that same shape thousands of years ago. It is like working through the logic of Pythagorean theorem, knowing that one is following the thoughts of a geometer of long ago. The form becomes no longer a lifeless shadow on a page, but a living idea. It just seemed too good to pass up.

3 THE PLAN OF THE FONT

The Choice of Sans-serif

We chose sans-serif as the base style of our first Unicode font, for several reasons. First, in most typographic cultures, sans-serif typefaces are more recent developments than seriffed faces, and therefore do not carry as many historico-cultural associations. Moreover, the common use of sans-serif types in signage for air travel and in international trade and communication have made them familiar around the world. Hence, the sans-serif style tends to have a neutral but familiar appearance, without connotations specific to any

particular nation, society, language, culture, or religion.

Second, our Unicode font is intended initially for use in a technical environment – computerized document preparation, exchange, and printing. Since the 1920s, sans-serif types have been promoted and promulgated for their modern, technical, industrial look, and therefore sans-serif seems appropriate across a broad range of computerized document processing.

Third, sans-serif faces represent an attempt to distill the essential graphical qualities of letterforms. Though history has shown that sans-serif types can never be truly devoid of stylistic features, they are, compared to their seriffed siblings, superficially simpler in appearance and therefore tend to exhibit the underlying forms of an alphabet without as many obscuring details. This makes it easier for designers to analyze the distinctive differences between characters, and to understand the important differences between alphabets.

Fourth, sans-serif offers a simpler initial solution to the problem of an italic companion. A "roman" typeface design (perhaps "upright" would be a less culture-bound term, since a Unicode font is likely to include Greek, Cyrillic, Hebrew, and other scripts) requires some sort of "italic" companion (again, perhaps "oblique", "slanted", "sloped", or "inclined" would be better terms). Seriffed typefaces usually have true cursive companions in which the shapes of the letters are not only inclined but different in basic form (compare the 'a' of Times Roman to the *'a'* of Times Italic), whereas sans-serif italics are usually oblique renderings of the roman.

Although many typographic purists believe that simple obliques are inferior to true cursives, Stanley Morison argued in 1926 that the ideal italic companion to roman should be an inclined version of the roman [21]. While Morison's theory has not been accepted for seriffed types (indeed, he did not follow it himself in the design of Times Italic), inclined roman has become the standard "italic" for sans-serif types.

Since an oblique face can be made by a simple shear transformation of an upright design, the initial release of Lucida Sans Unicode could be provided with a serviceable oblique companion through a simple algorithmic transformation performed by the imager of an operating system. A separate font would not initially be necessary. When the Unicode character set is implemented as a single large font, an algorithmic oblique saves memory, since a separate font does not need to be loaded into host or printer memory. Even if a separate oblique font is produced, it is much easier to develop than a true cursive italic (though careful designers "fine-tune" oblique designs and do not use simple algorithmic slanting). These are reasons based on economics, but there is another reason to prefer an oblique form as the companion face to a Unicode font: "true italic" is not a universal concept. The European mode of distinction between formal and cursive type forms is not as strong a tradition in some non-Latin scripts, e.g. Hebrew (though there may be other kinds of highly regulated graphical distinctions), so a simple oblique is a more universal, albeit minimal, graphic distinction that can apply equally to all non-Latin scripts.

The particular sans-serif typeface that is the basis for our Unicode font is Lucida Sans, part of a super-family of typefaces, including seriffed, sans-serif, fixed-pitch, script, and mathematical fonts [22] [23]. Like other "humanist" sans-serif typefaces such as Gill Sans and Hans Ed. Meier's Syntax [24], Lucida Sans is more akin to typefaces of the Renaissance, with their handwritten ductus, than to industrial sans-serifs of the 19th and 20th century. Because our study of each non-Latin script begins with handwriting and calligraphy (as did our study of Latin letterforms), we believe it is appropriate to base the Unicode font on a modern design that retains some handwritten feeling. Of course, another

important consideration is that Lucida Sans is our own design, which gives us a fundamental understanding of the "logic" of the design and how it can be extended and modified, and the legal right to make extensions and modifications. (In view of our argument in favor of an oblique "italic" for the Unicode font, we should note here that the original Lucida Sans Italic is a true cursive, based on Renaissance chancery forms, not an oblique roman, so the inclined Unicode version will be called Lucida Sans (Unicode) "Oblique" to distinguish it from the true cursive designs.)

The Design of Diacritics

When we analyzed the large numbers of accents and diacritics that appear in the Unicode standard, we concluded that, for improved legibility in international text composition, accents and diacritics should be designed somewhat differently than in the standard version of Lucida Sans. In a mono-lingual text in a standard orthography, there are relatively few different accents, and most good readers have a sense of the "word-images" likely to be found in a text, and can therefore easily disambiguate similar-looking words. Hence, although accents need to be clearly differentiated, they do not need to be emphatic, and. indeed, overly tall or heavy accents can be more distracting than helpful to readers. The situation is different in multi-lingual texts, where there can be many different accents used in several different orthographies, and where readers are not likely to be proficient in all the different languages. To aid legibility, or at least to increase decipherability, the diacritics require greater differentiation. Accordingly, we designed the lower-case diacritics of Lucida Sans Unicode to be slightly taller and a little different in modulation than those of the original Lucida Sans. Following current practice, we used the lower-case accents to composed accented capitals.

The Latin 1 and European Latin Unicode subsets contain many characters composed of letters and diacritics, but these do not limit the possible letter + diacritic combinations. One of the advantages of Unicode is that it includes a Generic Diacritical Marks set of "floating" diacritics that can be combined with arbitrary letters. These are not case-sensitive, i.e. there is only one set of floating diacritics for both capitals and lower-case. In our first version of Lucida Sans Unicode, we implemented these as lower-case diacritics and adjusted their default position to float over the center of a lower-case 'o'. Ideally, there should be at least two sets of glyphs, one for lower-case and one for upper case (determined automatically by the text line layout manager of the OS or application), along with a set of kerning tables that optimizes the visual appearance of each combination of letter + diacritic. Because of many questions and complexities concerning ideal positioning of diacritics, we postponed development of the alternate glyph set and kerning tables until a later version of the font.

Proportional and Fixed-pitch Spacing

In a proportional font (like the Times Roman before the eyes of the reader), the advance width of a character is proportional to the geometric logic of its form. A proportionally-spaced 'm', which has a spatial frequency of three cycles per letter, is wider than an 'n', which has a frequency of two cycles, which in turn is wider than an 'i', which has one. The rhythm of a word like 'minimum' is a regular alternation of positive and negative shapes, providing an even rhythm to the flow of text. To preserve the traditional look of the different alphabets and symbol sets brought together in the Unicode standard, we designed

our first Unicode font with proportional spacing.

In a fixed pitch font like Courier, all characters are of the same width, so that 'm' is cramped and 'i' extended, and 'minimum' has an irregular rhythm, since the spatial frequency of the letters is continually changing within the fixed width of the cells. Fixed-pitch fonts were standard in teletype and hence became the standard in computer terminals and printers until the mid-1980s. Despite their aesthetic shortcomings, fixed-pitch fonts provide well-behaved metrical simplicity, and hence have remained the default, even in advanced windowing systems, for TTY terminal emulators, "console" windows, programming interfaces, line printer emulators, and other software survivors of an anterior age.

Because of the undeniable utility of fixed-pitch fonts, it was not long after the release of our proportional Lucida Sans Unicode font, that we were asked to design a fixed-pitch version, to be backwards-compatible with the vast amounts of *retro* software still roaming at large in the computing world. We based this second Unicode compatible font on Lucida Sans Typewriter, which is often used to emulate terminal fonts and line printer fonts, and to distinguish program code from exposition in technical documents (for examples of latter usage, see the Proceedings of the 1993 Annual Meeting of the TeX Users Group [25]). Fixed-pitch fonts are usually neglected in discussions of the art of type design, since their modern association is with typewriting, a medium regarded as inferior to "real" typography. (But the cognoscenti know that fixed-pitch designs have an ancient history: several Greek inscriptions in the mono-spaced "stoichedon" style are from the 5th century B.C.) The design of fixed-pitch fonts is far from trivial and requires a good deal of ingenuity and artifice, even if the resulting works are not appreciated for their own sake. Among the alphabetic Unicode character sets, Cyrillic poses an interesting problem in fixed-pitch mode because it has, compared to the Latin, a greater percentage of characters with higher spatial frequency (three or more cycles per letter) in the horizontal axis. The Hebrew alphabet, on the other hand, is more easily transformed to fixed-pitch mode because it has many fewer letters of high spatial frequency.

Vertically Kerning Diacritics

The design brief for the fixed-pitch Unicode compatible font also restricted the heights of diacritics. Most TTY and console window software, in keeping with its retrograde origins, assumes that characters are disjoint – that no elements intrude into the cells of adjoining characters, whether above, below, or to either side. When text is scrolled, flowed, deleted, pasted, or manipulated, character elements intruding into the space of neighboring character cells can cause various kinds of problems – junk pixels left scattered around the screen, letter parts clipped off, and so on. While the assumption that characters are fully contained within cells was invariably true for primitive terminal and TTY fonts, it is not necessarily true of typographic fonts. In many PostScript and TrueType digital fonts, diacritics on capitals extend above the nominal upper boundary of the body of the font because, in an effort to reduce font file size, capital and lower-case accented characters are built as composites in which letters and diacritics are treated as subroutines, and only one form of the diacritics, the lower-case form, is used for both cases. The lower-case form of most diacritics is taller than the capital form, and so the diacritics on composite capitals often extend beyond the nominal top of the body.

The vertical "kerning" of accents is not usually a serious problem in printed texts, because in mixed upper- and lower-case composition, accented capitals are few and seldom

coincide with the descenders of the line above, and in all-capital setting, where accented capitals are frequent, descenders are rare, occurring only on 'Q' and sometimes 'J', themselves infrequent letters. Furthermore, if accent/descender coincidences occur too often in a printed text, more line spacing can usually be added, or the font scaled to a smaller size within a given line spacing. On screen displays, addition of line spacing is not an effective solution for terminal emulators of fixed size, because it alters the number of text rows in the window. Scaling the font to a smaller size within the body diminishes the effective resolution, reducing the detail and legibility of the font.

We addressed this problem by creating a special, Procrustean version of Lucida Sans Typewriter. This new font, dubbed Lucida Console, has special shortened capitals for Latin, Cyrillic, Greek, and other alphabets that use capital forms, and special shallow capital diacritics. The resulting design provides accented capitals that can be adequately distinguished, maintains adequate size differentiation between capitals and lower-case, and maintains a large lower-case x-height, all within a strictly limited body size, and a fixed-pitch. Because we were able to develop the design by combining Lucida Sans Unicode with Lucida Sans Typewriter, the development process was much faster and more efficient than starting a new design.

4 PUZZLES & PROBLEMS

Having described reasons in favor of creating a Unicode font, we should also discuss arguments against such an undertaking, and various problems we encountered.

Ars longa, vita brevis

The first objection could be called the 'Art is long, life is short' argument, from the well known proverb of Hippocrates. A font that would contain all of Unicode 1.0 would be of daunting size, and the standard continues to grow as the Unicode committee adds more characters to it. Even without the Chinese, Japanese, Korean set, the alphabets and symbols comprise almost 4,000 separate characters, sixteen times more than the usual 8-bit character sets. If the designs of these characters are to be harmonized, they must be originated by a single designer or by a close collaboration between a few designers. This means that any original Unicode font design will in most instances require years to complete, and hence that earlier releases of the font will inevitably be incomplete. We plan to develop the font as a series of software releases, with each new release adding "functionality" in the form of additional complete character block.

To call an incomplete font containing Unicode subsets a "Unicode" font could be misleading, since some users could mistakenly assume that any font called "Unicode" will contain a full set of 28,000 characters. However, a font that contains multiple subsets poses a puzzle in naming. Our first release of Lucida Sans "Unicode" contains some 1,725 characters, including the Unicode blocks of ASCII, Latin 1, European Latin, Extended Latin, Standard Phonetic, Modifier Letters, Generic Diacritical Marks, Greek, Cyrillic, Hebrew, General Punctuation, Superscripts and Subscripts, Currency Symbols, Letterlike Symbols, Arrows, Mathematical Operators, Pictures for Control Codes, Form and Chart Components, Blocks, and Geometric Shapes. Other blocks, including Armenian, Arabic, Devanagari and other Indic alphabets, as well as various symbol sets, such as Number Forms, Miscellaneous Technical, and Miscellaneous Dingbats, will be added in future

releases. We have not yet been able to devise a naming scheme that can concisely denote all the subsets contained within a Unicode compatible font. Since many operating systems place severe limitations on the length of file names, this naming problem remains unsolved.

The 4,000 letters and symbols are still only a fraction of the full Unicode set; there are in addition some 24,000 "Han Ideographic" characters [2]. These are the logographs used in Chinese, Japanese, and Korean scripts. The Unicode standard "unifies" similar characters in these scripts, since the ideographs are for the most part derived from common sources, but cultural differences between the different nations mean that different graphical styles of the ideographs are preferred in the different countries. As a result, a single rendering of the Han Ideographic set would not be acceptable to readers in all three countries. Different renderings of thousands of glyphs would be required, and hence, because art is long and life short, we must admit that the magnitude of the Han Ideographic task appears beyond the scope of our small partnership, despite the siren call of the marvelous logographic characters. One practical solution would be for our "alphabetic" Unicode font to be used in conjunction with ideographic fonts of similar design. For Japanese, for example, the Lucida Sans Unicode font could be amalgamated with a "gothic" style Kanji font chosen to visually harmonize with Lucida Sans in general weight and proportion. (The Japanese term "gothic" is equivalent to "sans-serif", and is also used in English for sans-serif faces, particularly those of 19th century American design, e.g. Franklin Gothic and News Gothic.) Similar amalgams could be made for Chinese and Korean. Our role as designers would then be one of cooperation with Japanese, Chinese, and Korean colleagues.

Culture-bound Design

A second objection could be called the "culture-bound" argument, based on the notion that type designers proficient in one tradition cannot adequately render designs in a different tradition. We have already mentioned cultural differences between the Han Ideographic characters used in different Asian countries, but we can easily point to similar problems, although on a much smaller scale, in Latin scripts. A minor but instructive example can be seen in Times Roman, Stanley Morison's 1931 adaptation of a 16th century French design (with perhaps some Flemish influence). Although Morison was one of the pre-eminent typographic authorities of the 20th century, and Times Roman one of the most popular typefaces of all time, some French typographers objected to Times New Roman as excessively Anglo-Saxon in style. To satisfy the French critics and give Times greater appeal in the French market, the Monotype Corporation cut a special version of the face in accord with the dictates of Maximilien Vox, a noted French typographic authority [15] [26]. Vox's re-design of some fourteen characters brought Times slightly closer to the sophisticated style of the French Romain du Roi, cut by Philippe Grandjean *circa* 1693. If the English and French, two peoples who have engaged in literate cultural exchanges ever since the late 8th century when Charlemagne invited Alcuin of York to become the master of the Palace Schools of the Frankish Empire, can disagree on matters of typographic taste, then we can expect even greater differences between cultures more widely separated.

The French are not alone in feeling that Times requires modification. For the German typographic market, Monotype cut a version of Times with lighter capitals that are less distracting in German orthography, where every noun is capitalized [15]. Although we have not seen this version used in laser printing, the Swiss type designer Hans Ed. Meier has demonstrated a similar re-design of Times with lighter capitals, done experimentally for

the department of Informatik at ETH Zurich [24].

An instance of a greater cross-cultural gulf is seen in Eric Gill's design of a Hebrew typeface design of 1937, which Henri Friedlaender's criticizes as a "curiosity" that "is a hybrid attempt to implant the system of Roman Capitals into Merooba which is governed by entirely different laws" [27]. Friedlaender's own Hadassah design is still used in Israel, but Gill's is not.

When confronting the question of culture-bound design, we can seek guidance from social sciences such as ethnology, and linguistics, which assume that human culture is knowable and transmissible beyond the boundaries of a single society. Such faith applies equally to the design of letterforms. For a designer in the Latin tradition, a non-Latin script is an opportunity to learn new rules of design, new formal concepts, new modes of legibility. Indeed, throughout the history of European typography, many of the finest type designers embraced non-Latin forms with obvious enthusiasm and evident respect. Francesco Griffo's Aldine Greeks [28], Claude Garamond's Grecs du Roi [29], Guillaume Le Bé's Hebrews [30], and Fournier's Greeks, Hebrews, Syriac, and Arabic [6, vol. 2] testify to the skill, understanding, and application that designers have brought to foreign scripts.

If a patron saint or historical paragon of multi-cultural type design were needed to lend respectability to modern efforts, our nominee would be the 16th century French punch-cutter, Robert Granjon. An artist of remarkable skill in the cutting of stylish romans and italics, he was astonishingly versatile in the cutting of splendid fonts of non-Latin scripts, including Armenian, Arabic, Cyrillic, Syriac, and possibly Greek and Hebrew as well. His ecclesiastical patrons in Rome called him "the excellent...", "the most extraordinary...", "the best ever..." cutter of letters [31].

Homogenization

Against the harmonized design approach, one could argue that regularization of design elements is homogenization. If it erases distinctive differences between scripts, it increases the possibility of confusion. In this view, the attempt to filter out unwanted noise may instead remove significant portions of the signal and distort what remains, thus decreasing the signal and adding noise.

To avoid this potential problem, we attempted to learn the basic rules of each non-Latin alphabet on its own terms, before starting the design of the Unicode version. We studied handwriting manuals for children, calligraphy books for adults, photographs of historical manuscript hands, typefaces created by designers literate in the script, and examples of text typography and signage of different kinds. This is not a foolproof approach, but at least it provides a foundation that is based on the script itself, rather than on Latinate notions of "correct" letter design. We did not try to adapt non-Latin scripts to a strict Latin model. For example, in designing Lucida Sans Hebrew, we developed and tested three versions. In one version, the Hebrew letters aligned with the Latin capitals; in another they aligned with the Latin lower-case; in a third, they were midway between. The third one provided the best visual equivalence when Hebrew text is used adjacent to Latin text of mixed capital and lower-case. Similarly, we followed a traditional Hebrew thick/thin modulation, in which horizontal elements are thicker than vertical – the opposite of the Latin convention – but weighted the Hebrew characters to have visual "presence" equivalent to that of the Latin.

Character Standard vs. Glyph Standard

As noted above, Unicode is a character encoding standard, not a glyph standard. It does not recognize alternate versions of characters if those versions are merely "allographic", i.e. if the differences are not significant in an orthographic sense, if they do not change the meaning of a word. For example, Unicode treats Latin capital 'B' as one character and lower-case 'b' as another because these are significant differences in Latin orthography. A "Bill" is a man whose first name is "William", whereas a "bill" could be a bird beak, a cap brim, or an invoice. Unicode does not treat italic 'b' or a bold 'b' as separate from 'b', because those letters are merely "allographs", as the linguists would say, the graphic differences not being orthographically significant.

Characters and glyphs may not always exist conveniently in an isomorphic, one-to-one mapping in a font. For many uses, fonts should contain more glyphs than characters. Sophisticated Latin typography often requires alternate forms of letters or symbols that are different in a graphical sense but not in a character sense. In the higher class of books and journals, for example, small capitals and old-style figures are often used. Ornamental swash variations of letters and fancy ligatures are popular in "fine-printing". Unicode does not encode such distinctions, though designers may well wish to include them in a font. Among certain non-Latin scripts, the differences between characters and glyphs become even greater and more fundamental. Arabic scripts are notable for using context-sensitive allographs of most letters. Initial, medial, final, and isolated forms are obligatory for many letters, but these are graphical variations, not semantic differences, and are not encoded by Unicode. Hence, one character can map to several glyphs, depending upon its context.

A somewhat different problem can occur between languages that use slightly different variants of the same characters. S-cedilla and T-cedilla are used in both Turkish and Romanian, but the cedilla may be rendered as either the French form of cedilla or as a comma-like accent below the letter. Different versions of the cedilla are preferred by different groups of users, but Unicode does not provide separate characters for the variants.

Our first release of Lucida Sans Unicode simplistically treats the character encoding as a glyph encoding, and offers few alternate glyphs, except in an *ad hoc* fashion (encoded with "user-defined" instead of standard codes). Despite its over-simplification of a potentially subtle and complex issue, a one-to-one mapping of characters to glyphs does make the font simpler to use and more portable, since any operating system can access all the glyphs directly through the character encoding. We see the character vs. glyph problem mainly as a temporary limitation of technology, not as a philosophical conundrum for designers. Font handling by operating systems is becoming more powerful in the handling of alternate glyphs. For example, Macintosh QuickDraw GX provides a powerful mechanism for accessing alternate glyphs in a font [4]. Eventually Unicode fonts will be able to include as many alternate glyphs as necessary to achieve the level of quality and breadth of application that a designer or user may require.

One Big Font vs. Many Little Fonts

Once digital glyphs have been designed and developed for Unicode subsets, e.g. the ASCII, Latin 1, and Cyrillic subsets, they can be implemented in either of two main methods: as a set of small fonts, each corresponding to one Unicode subset, or as one big font into which all the subsets are assembled. The choice of method is of course important to operating

system builders, but does not greatly impact font designers. Pike and Thompson [3] discuss the advantage of economical memory management and greater font loading speed when a complete Unicode character set is implemented as many small subfonts from which characters can be accessed independently, and this is the method used in the Plan 9 operating system, both for bitmap fonts and the Lucida Sans Unicode outline fonts. The other method is used in Microsoft Windows NT 3.1, in which the first version of Lucida Sans Unicode font is implemented as a single TrueType font of 1,740 glyphs. In the current version of Windows NT, this allows a simpler font-handling mechanism, makes the automatic "hinting" of the font easier, since all characters can be analyzed by the hinting software in one pass, and preserves the default design coordination of the subsets, if the font is based on a harmonized set of designs. As designers, we were able to provide our Lucida Sans Unicode designs in the form of many small "fontlets" for use in Plan 9, and as one big TrueType font for use in Windows NT 3.1.

Revisions and Updates

Standards, even though they are supposed to be standard, are subject to change. We began the design of the first version of our Unicode font with reference to the Unicode Standard Version 1.0, but before we finished, we found it necessary to make a few changes to bring the font into conformance with Unicode Version 1.01. In 1992-93, Unicode was aligned with the proposed ISO/IEC 10646 standard, resulting in a new Unicode Version 1.1, which incorporates more than 5,000 additional characters, including ligatures for contextually sensitive glyph forms in Latin, Armenian, Hebrew, and Arabic. [31] The increase in size and complexity of fonts with very large character sets, and the need to update such fonts in accordance with changing standards and with the addition of further character sets means that future font naming conventions will need to include version numbers and indicators of the scope of the character set.

5 CONCLUSION

In a set of 1,700 character designs, there are many details that can be discussed, but we believe we have touched upon several of the more interesting and general issues. For brevity, we conclude by showing a specimen of the first version of Lucida Sans Unicode, arranged according to the layouts of the Unicode subsets.

ACKNOWLEDGEMENTS

We are grateful to Steve Shaiman and George Moore at Microsoft Corporation for commissioning the first version of the Lucida Sans Unicode font, and to David McBride, Dave Ohara, Asmus Freytag, Michel Suignard, and others at Microsoft for helping test and debug the font on the way to its eventual release with Microsoft Windows NT 3.1. We thank Rob Pike, John Hobby, and others at AT&T Bell Laboratories for getting a version of Lucida Sans Unicode up and running in Plan 9. Glenn Adams at the Unicode Consortium alerted us to several intriguing problems concerning the practicality and advisability of attempting the design of a Unicode font. Pierre MacKay, William Bright, Michael Sheridan, Rick Cusick, and Kathy Schinhofen gave us helpful alphabet specimens and/or advice, which they may have forgotten but which we remember with thanks.

REFERENCES

1. The Unicode Consortium, *The Unicode Standard: Worldwide Character Encoding,* Version 1.0, volume 1, Addison-Wesley, Reading, Massachusetts, 1991.
2. The Unicode Consortium, *The Unicode Standard: Worldwide Character Encoding,* Version 1.0, volume 2, Addison-Wesley, Reading, Massachusetts, 1992.
3. Rob Pike and Ken Thompson, 'Hello world....', in *Proceedings of the Winter 1993 USENIX Conference*, pp. 43–50, San Diego, (1993).
4. Apple Computer, *Inside Macintosh: QuickDraw GX Typography*, Addison-Wesley, Reading, Massachusetts, 1994.
5. Joseph Moxon, *Mechanick Exercises on the Whole Art of Printing*, Herbert Davis and Harry Carter, eds., Oxford University Press, Oxford, 1962.
6. Pierre-Simon Fournier, *Manuel Typographique*, volume 1, chez l'Auteur, rue des Postes, chez Barbou, rue St. Jacques, Paris, 1764. volume 2, *idem*, 1766.
7. Lucien LeGros and John Grant, *Typographical Printing-Surfaces*, Longmans, Green, London, 1916. (reprinted, Garland Publishing, New York, 1980).
8. Arthur H. Phillips, *Computer Peripherals & Typesetting*, HMSO, London, 1968.
9. Arthur H. Phillips, *Handbook of Computer-Aided Composition*, Marcel Dekker, Inc., New York, 1980.
10. Apple Computer, *Inside Macintosh*, volume 3, Addison-Wesley, Reading, Massachusetts, 1989.
11. Microsoft Corporation, Redmond, Washington, *TrueType 1.0 Font Files: Technical Specification*, Revision 1.02, May 1993.
12. Adobe Systems, *PostScript Language Reference Manual*, Addison-Wesley, Reading, Massachusetts, 1985.
13. John Dreyfus, *The Work of Jan van Krimpen*, Sylvan Press, London, 1952.
14. Evgenia Butorina, *The lettering art: Works by Moscow book designers*, Kniga, Moscow, 1977.
15. John Dreyfus, *The Evolution of Times New Roman*, Penrose Annual, London, 1972.
16. Monotype Typography, Salfords, Redhill, *Library of Non-Latin Typefaces*, 1992.
17. Linotype Corporation, Eschborn, *LinoType Collection*, 1988.
18. Miles A. Tinker, *Legibility of Print*, Iowa State University Press, Ames, 1963.
19. Bror Zachrisson, *Studies in the Legibility of Printed Text*, Almqvist & Wiksell, Stockholm, 1965.
20. Joseph Shimron and David Navon, 'The distribution of visual information in the vertical dimension of roman and hebrew letters', *Visible Language*, **XIV**(1), (1980).
21. Stanley Morison, 'Towards an ideal italic', *The Fleuron*, **V**, (1926).
22. Charles Bigelow and Kris Holmes, 'The design of Lucida: an integrated family of types for electronic literacy', in *Text processing and document manipulation*, ed., J. C. Van Vliet, Cambridge, (1986). Cambridge University Press.
23. Microsoft Corporation, Redmond, Washington, *Microsoft TrueType Font Pack for Windows User's Guide*, 1992.
24. Hans Ed. Meier, 'Schriftgestaltung mit Hilfe des Computers – Typographische Grundregeln mit Gestaltungbeispielen', Technical report, Department Informatik, Institut für Computer Systeme, ETH, Zürich, (1990).
25. 'TEX Users Group annual meeting proceedings', *TUGboat*, **14**(3), (1993).
26. John Dreyfus. personal communication.
27. Henri Friedlaender, 'Modern hebrew typefaces', *Typographica*, **16**, (1963?).
28. Nicolas Barker, *Aldus Manutius and the Development of Greek Script and Type in the Fifteenth Century*, Chiswick Book Shop, Sandy Hook, Connecticut, 1985.
29. *Cabinet des poinçons de l'Imprimerie nationale de France*, eds., Raymond Blanchot and Georges Arnoult, Imprimerie nationale, Paris, 1948.
30. Hendrik D. L. Vervliet and Harry Carter, *Type Specimen Facsimiles II*, University of Toronto Press, Toronto, 1972.
31. The Unicode Consortium, 'The Unicode Standard, version 1.1', Unicode Technical Report 4, Unicode, Inc., Menlo Park, California, (1993).

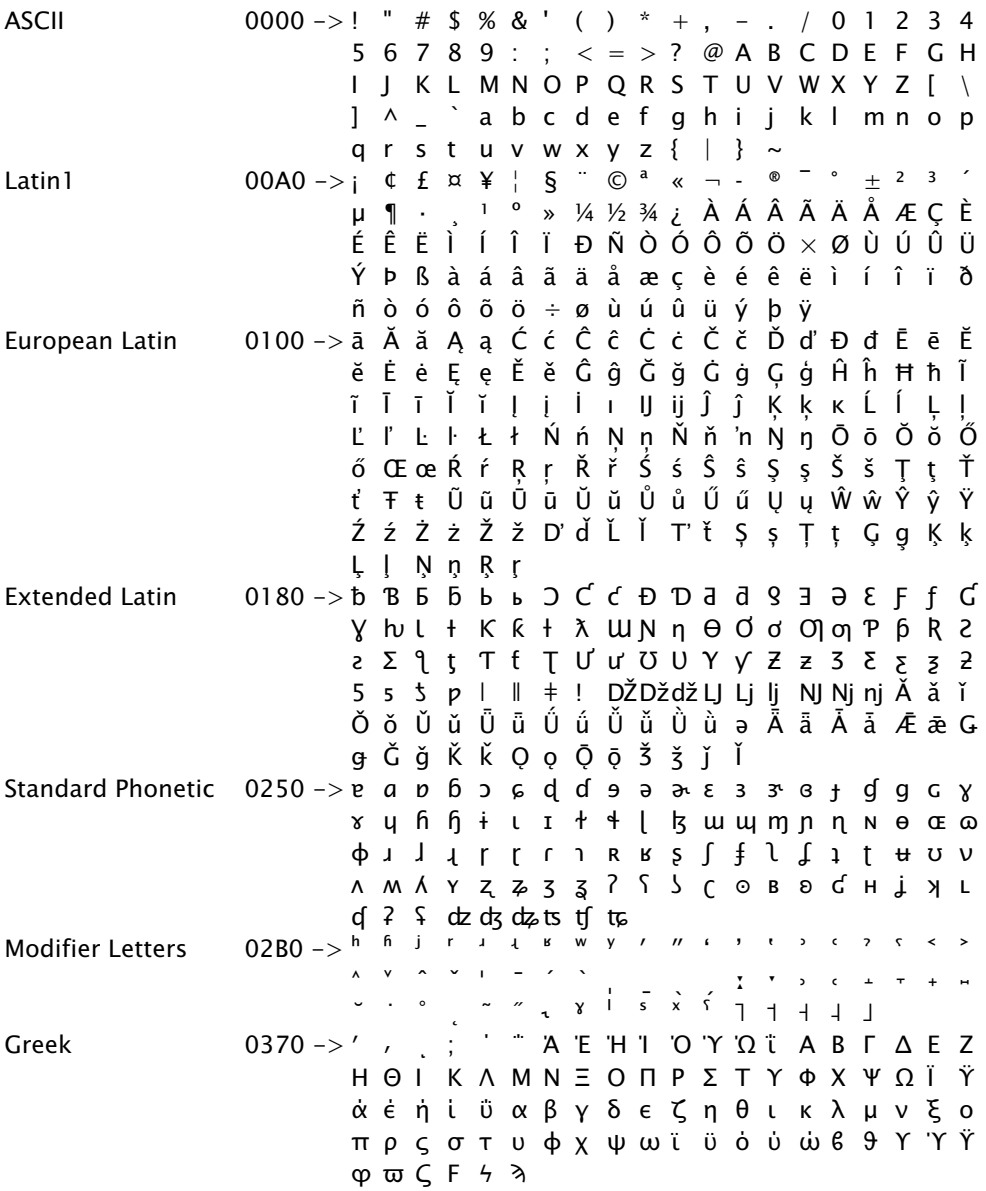

Figure 1. Palette of LucidaSansUnicode (part 1 of 3)

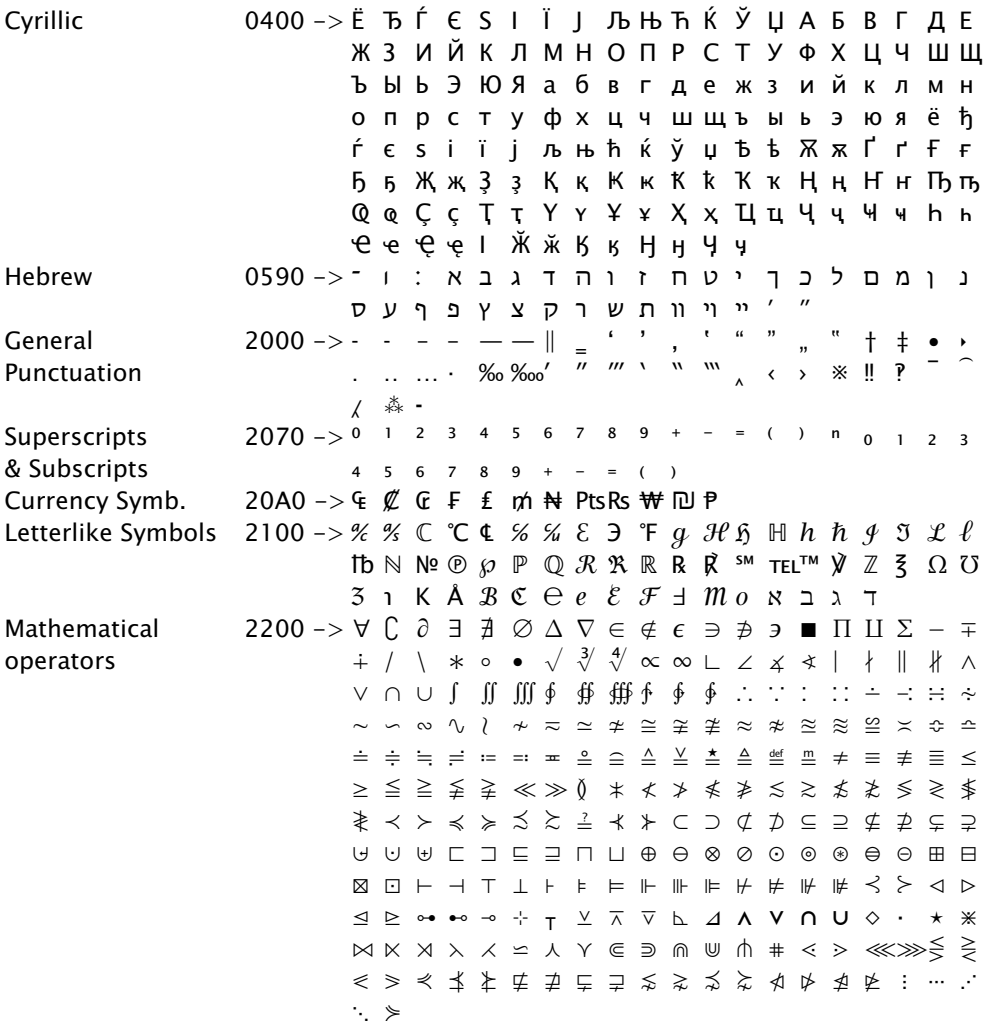

Figure 2. Palette of LucidaSansUnicode (part 2 of 3)

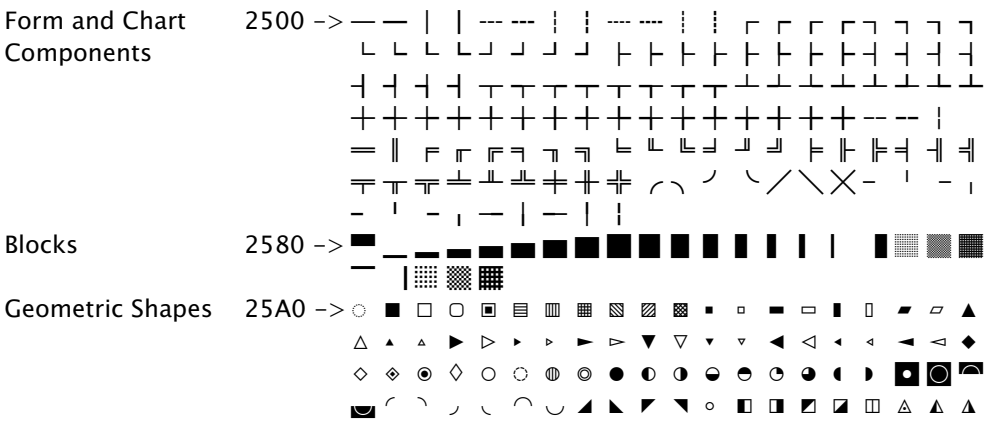

Figure 3. Palette of LucidaSansUnicode (part 3 of 3)