

INTERNATIONAL CULTURAL DIFFERENCES IN SOFTWARE

By Timothy G. Greenwood

ABSTRACT

Throughout the world, computer users approach a computer system with a specific set of cultural requirements. In all cultures, they expect computer systems to accommodate their needs. A major part of interaction with computers occurs through written language. Cultural requirements, particularly written languages, influence the way computer systems must operate. Cultural differences concerning national conventions for the presentation of date, time, and number and user interface design for the components of images, color, sound, and the overall layout of the screen also affect the development of computer technology. Successful computer systems must respond to the multicultural needs of users.

INTRODUCTION

Not chaos-like, together crushed and bruised,
But, as the world harmoniously confused:
Where order in variety we see,
And where, though all things differ, all agree.
-- Alexander Pope

[NOTE: The reader is advised that the occurrence of [*] in this file represents an unusual symbol, such as a rural mail box, or a character from a language that does not use the Latin alphabet. These symbols and letters are not available in ASCII format.

In the first years of the computer age, users adapted themselves to the requirements of the computer. They had to learn the language of the machine to interact with it. Now the computer is part of daily life, a tool to complete a task. Computer systems must be adapted to the needs of their users. Computer users approach a computer system with a specific set of cultural requirements. Successful systems respond to these requirements.

INTERNATIONAL ADAPTATION OF COMPUTER SYSTEMS

Each nation has developed its own culture, and some areas of the world share a cultural background. Adaptation of computer systems to different cultures uses processes known as localization and internationalization.

Localization is the process of changing products to suit users from different cultural backgrounds. Localization is achieved by taking the source code for a product developed for one country and modifying the source code and product to satisfy the needs of other countries. Often teams of developers in different countries are needed to adapt products. If the original product is not

built with a view toward being localized, this can be a very expensive and time-consuming process. There is the direct cost of multiple development teams modifying the source code of the original product. This process also produces multiple code bases, which makes future development and maintenance more complex.

Building software that can be localized with minimal software changes is called internationalization, often abbreviated to I18N (the letter I followed by 18 letters and the letter N). The basis of internationalization is to identify those cultural elements that the software must accommodate and to simplify the task of adapting the product. This paper describes a set of these cultural elements. The remainder of this issue of the Digital Technical Journal details specific aspects of building international software.

CULTURAL DIFFERENCES

Language is the most obvious cultural difference among people. Written language is an important method of communication with computers. This paper examines written languages and their representation in computer systems. It also presents cultural differences concerning national conventions for the presentation of date, time, and number and user interface design for the components of images, color, sound, and the overall layout of the screen. The base functions of a product may change in response to different needs around the world, and some examples of these differences are illustrated. Finally, with a look to the future, the paper presents deeper cultural differences that are only beginning to be represented in software.

Written Language

The written representation of spoken language requires a script and an orthography. The script is the set of symbols that represents the sound or meaning of components of the language. The orthography consists of the rules of spelling and pronunciation. Specific spelling and pronunciation rules may differ among locations or communities; for example, the American English orthography differs from the British English orthography. A script may be tied to a specific language, for example, Korean Hangul, but frequently a script can represent several languages. French and Italian both use the Latin script.

A written language may be broadly categorized into either an ideographic, a syllabic, or an alphabetic writing system. The category is determined by examining the relation between the symbols in the script and the unit of sound or meaning represented.

In writing systems based on ideograms, every symbol has a specific meaning that is not related to its pronunciation. The

ideograms imported from Chinese, and used in Chinese, Japanese, and South Korean writing provide examples in current use.[1] Thus [*] represents a man or person, even though it is pronounced ren in Chinese, zin in Japanese, and in in Korean.[2] "3" represents "three" even though it is pronounced tatu in Swahili and trwa in French. Ideographic writing systems typically contain several thousand discrete symbols with a subset of approximately 2,000 symbols in frequent use. The users of this writing system continue to develop new symbols.

In the syllabic writing systems, each symbol represents a syllable. [*] in Japanese katakana denotes the ma sound. There is a wide variation in the number of discrete symbols in a syllabic system. Japanese kana uses some 47 symbols; the writing of the Yi people (a minority nationality scattered through provinces in Southwest China) uses a standardized syllabary of 819 symbols.[3]

In alphabetic systems, each symbol or letter approximately matches a phoneme (the smallest unit of speech distinguishing meaning). Thus M in Latin script, [*] in Hebrew, and [*] in Armenian denote the m sound. Most alphabets have from 30 to 50 discrete letters.[4] The match between phonemes and letters is not exact, especially in English, which has about 40 phonemes.[5] Some phonemes are represented by letter sequences, such as the th in thank.

No written language uses a pure set of either alphabetic, syllabic, or ideographic symbols; each does use one set of symbols predominantly. The Latin script is primarily alphabetic, but numerals and certain signs such as & are ideograms.[6] Other languages use a more even mix. South Korean combines the native Hangul alphabet with Hanja, the Korean name for their ideographic characters. Japanese combines the katakana and hiragana syllabaries (collectively called kana) with the ideographic characters called kanji in Japanese. Written Japanese, especially technical and advertising material, also often uses the Latin alphabet, called romaji.

Character Placement. In most European languages, basic symbols are written in a linear stream with each character placed on a baseline. In other writing systems, for example, Korean Hangul, the elements do not follow this linear layout. Rather than evolving piecemeal like most writing systems, Hangul is the result of deliberate, linguistically informed planning. It has been called "...probably the most remarkable writing system ever invented." [7] Korean uses an alphabet of 14 consonants and 10 vowels. These letters, called jamo, are blocked into syllable clusters. If the same technique were applied to English, cat might be written [*]. Figure 1 shows the Korean Hangul alphabet, and Figure 2 shows the jamo blocked into syllable clusters.

[Figure 1 (Korean Hangul Consonant and Vowel Signs) is not available in ASCII format.]

[Figure 2 (Korean Hangul Text Showing Blocking of Jamo) is not available in ASCII format.]

Thai also uses an alphabet and is written with the symbols arranged in a nonlinear fashion. Thai is a tonal language; different tones distinguish words that would otherwise be homonyms. Thai words consist of consonants, vowels, and tone marks. Each component is an atomic unit of the language. A vowel is written in front of, above, below, or behind the consonant to which it refers. A tone mark, if present, is usually placed above the consonant or above the upper vowel. Thai potentially has symbols at four levels, as shown in Figure 3. Level one is an optional tone mark. Level two is an optional vowel. Level three is either a consonant or a vowel preceding or following a consonant. Level four is another optional vowel. A consonant never has a vowel at both level two and level four.

Some level-three consonants have part of their glyph images rendered in another level. They either dip into level four or rise into level two. The last letter (yo ying) in Figure 3 is a level-three consonant, but it has a small (separate) portion written below the baseline. When this letter is written, this small portion is written at level four. When this letter is followed by a true level-four vowel, the vowel is shown instead of this portion.

[Figure 3 (Thai Script) is not available in ASCII format.]

In Figure 3, both tone marks are shown at level one to aid understanding of the script. In high-quality printing of Thai, if a cell does not have a level-two vowel, then the tone mark falls down to be positioned directly above the consonant; however, it is also currently acceptable for all tone marks to be physically positioned at level one. Thai mechanical typewriters position all tone marks at level one.

Diacritics and Vowels. In Arabic and Hebrew alphabets, vowels are indicated by placing vowel points above, below, or beside the letter. (Arabic also uses the consonant letters alif, ya, and waw to represent the long vowels a, i, and u.) Vowels are normally used only in religious text and in teaching materials for people learning to read the language; in other texts, vowels are inferred by the reader. Since vowel points are used, written Hebrew with vowels is called pointed. Figure 4 shows pointed Hebrew from a children's comic and the same text with the vowels removed.

[Figure 4 (Pointed and Unpointed Hebrew) is not available in ASCII format.]

Other Symbols. Most languages written with Latin letters have diacritical marks on some letters. In some cases, the use of a diacritic provides a stress or pronunciation guideline, as in the

English word coöperate. Removing the diacritic does not change the meaning of the word. In other languages, a mark that appears to be a diacritic is a fundamental part of the letter. The Danish letter å is a separate letter in the alphabet and is not a variant of a. In German, three vowels have umlauts and are separate letters in the alphabet. The deletion of an umlaut can change the definition of a word; for example, schwül means hot or humid, and schwul means homosexual.

Presentation Variants. The characters in the Arabic writing systems change form, depending on whether they are the first, last, or middle character of a word, or if they stand alone. Note that the abstract characters themselves do not change, only the glyph image. Figure 5, adapted from Nakanishi, shows the presentation variants of Arabic letters.[8]

[Figure 5 (Arabic Presentation Forms) is not available in ASCII format.]

Writing Direction

In English and many other writing systems, the letters are written from left to right, with lines progressing from top to bottom. Japanese, Chinese, and Korean may also be written in this form but are traditionally written vertically. The characters flow from the top of the page to the bottom, with lines advancing from right to left. The pages are ordered in the opposite direction to that used for English. Mongolian is also written vertically, but the columns of text advance from left to right. Consequently, pages of Mongolian text are ordered in the same direction as in English.

Figure 6 shows a portion of a newspaper printed in Taiwan. The newspaper exhibits many styles of format. Headlines may run horizontally from right to left, or left to right; the text of the article may run vertically; and advertisements and tables may run horizontally from left to right.

[Figure 6 (Taiwanese Newspaper) is not available in ASCII format.]

In Japanese writing, Latin characters (romaji) are interspersed with vertical kanji (Han script) characters. Romaji may be presented with each character in a horizontal orientation running vertically down, or they may be presented vertically, with each character rotated by 90 degrees. In addition, if one, two, or three Latin characters are mixed with vertical Han script characters, they may be presented horizontally in the vertical stream. Figure 7 shows mixed characters in a Japanese text.

[Figure 7 (Japanese Text Showing Latin Characters Mixed with Kanji) is not available in ASCII format.]

Semitic language scripts (e.g., Hebrew and Arabic) are written horizontally from right to left, with lines advancing from top to bottom, but any numbers using Arabic numerals are written left to right.[9] Any fragments of text written in the Latin script are also presented left to right. This method leads to nesting segments of reversed writing direction as shown in Figure 8. The text in this figure reads "Attention: Kalanit (1984) Tel-Aviv, ISRAEL;" where "Kalanit (1984)" is a company name. Figure 9 is another example of combining left-to-right and right-to-left text. It shows a portion of the contents page from the EL AL airline magazine.

[Figure 8 (Nested Bidirectional Text) is not available in ASCII format.]

[Figure 9 (Combined Direction Text) is not available in ASCII format.]

TEXT INPUT

The following section discusses techniques for addressing cultural differences in computer keyboard input.

Alphabetic writing systems typically have no more than 50 discrete symbols. Computer keyboards contain approximately 48 keys with symbols from the writing system inscribed. The depression of a key produces a code from the keyboard that is translated into a character coding according to some predefined coding. Input of a character not represented directly on the keyboard requires depression of several keys. For example, in terminals from Digital, the ß character is input on non-German keyboards by pressing "Compose s s". Latin keyboards typically have two possible characters available from each alphabetic key: lowercase letters are displayed by depressing the key alone, and uppercase letters are produced by depressing a shift modifier or a locking shift and the letter key.

Some keyboards have four levels, with three or four characters available from each key. Figure 10 shows the Arabic keyboard from Digital and the Khmer keyboard from Apple Computer. The user switches into the additional two groups of characters with an additional modifier or shift key. Note that the Arabic keyboard uses the additional group to support Latin characters as well as Arabic, but the Khmer keyboard uses all four groups for the Khmer characters only.[10] A four-group keyboard is now a national standard in Germany.[11]

[Figure 10 (Arabic Keyboard (above) and Khmer Keyboard) is not available in ASCII format.]

The katakana and hiragana syllabaries have approximately 50 characters each. These can be input either directly from the

keyboard or through a mapping of the syllable typed with the phonetically equivalent Latin characters. For example, the character [*] (ma) can be input either by typing the [*] key on a Japanese keyboard, or by typing m and a and using an input method to convert to [*].

Although some early keyboards had many ranks of individual keys, input of ideographic characters from modern keyboards always requires a multiple-stroke input method, with some user interaction. Both Japanese and Korean have phonetic writing systems. Users of these languages primarily use phonetic methods to input ideographic characters. The Chinese-language user has many different input methods; these are based on phonetic input or on strokes or shapes in the character. Almost all of these methods display a list of possible candidates as a result of the string input, and the user selects the appropriate candidate. The implementation of Japanese input methods is detailed in a separate paper in this issue of the Journal.[12]

Bidirectional Text

Hebrew and Arabic user interfaces have an additional level of difficulty. As discussed earlier, the text is bidirectional; the primary writing direction progresses from right to left and includes left-to-right segments of numbers and non-Hebrew or non-Arabic text. As shown in Figure 8, these segments can nest. The order in which to read bidirectional text can be ambiguous and can depend on the semantics of the text. Figure 11 duplicates the information on a pair of signs displayed at parking lots in Tel-Aviv. Urban legend has it that at least one parking ticket was dismissed by the court on the argument that the sign indicated that parking was not allowed from 5:00 p.m. to 9:00 a.m.

[Figure 11 ("No Parking" Signs in Tel-Aviv) is not available in ASCII format.]

To some extent the correct direction can be assigned automatically. Hebrew and Arabic characters have an implicit direction of right to left, and Latin text has an implicit left-to-right direction. Thus an output method can lay out simple combinations of bidirectional text correctly. Beyond these characters, direction can be ambiguous. Punctuation marks are common to both Hebrew and Latin text. Thus a period or comma or space has no implicit direction; the software must wait for the next character to determine the direction of the segment. In more complex cases such as the nested directions shown in Figure 8, direction attributes must be explicitly assigned to the segments.[13] As discussed in the paper on Unicode in this issue, the Unicode and ISO 10646 characters sets do include a rich set of directional markers.[14]

Insertion of text should be performed in the way the user finds most convenient, which is not necessarily in accordance with the

"correct" directional order of a segment. If entering a two-digit number in the "common" direction requires too many operations, or if the user was trained on a manual typewriter, most users would use the easiest typing order, i.e., entering the least-significant digit first and the most-significant digit second. "Smart" software, which puts the digits in the supposedly correct order, is not doing this user a service.

NATIONAL CONVENTIONS

Various entities such as date, time, and numeric values can be presented differently. Such presentation differences develop both from national and from personal or company styles. These presentation differences are not only tied to different writing systems. For example, dates are presented differently in the United States and in England.

Date Formats

The ninth day of October 1990 is written 9/10/90 in Europe but 10/9/90 in the United States. The order of the day and month numerals is well defined for a particular culture, but there are no overall formats for the separator used, or indeed for the general style. The separator may be a slash, hyphen, colon, space, or another symbol, according to policy or personal preference. The style may be numeric date as shown or the name of the month may be spelled out, and the year may be two or four digits.

In Japan, dates are based on the reign of the emperor. As shown in Figure 12, 1990 was the second year of Heisei, the reign of the current emperor. (The first and last years of two eras may coincide. Showa, the previous era, ended January 7, 1989, and Heisei started on January 8, 1989.) This date format is routinely used in business in Japan. The Western date formats are also used, so a date parsing program should be able to process both formats.

[Figure 12 (Japanese Date Format) is not available in ASCII format.]

Time-of-day Formats

Similarly, time-of-day formats vary according to personal and, to some extent, national preference. Possible time formats include

9.15am 09:15 0915 09:15:36 09 15 09h15

Time-zone abbreviations also change around the world. Two or more different abbreviations may indicate the same time zone. Eastern Standard Time (EST) is a U.S.-specific time-zone indicator. This

zone is called HNE (Heure Normale de l'Est) in French-speaking Canada. Central European Time is known as HEC by the French-speaking populations and as MEZ by German speakers. The same time-zone abbreviation may stand for different time zones. AST is used for both Alaska Standard Time and Atlantic Standard Time, which are five hours apart. Time-zone abbreviations are not standardized and may change. Time zones are not all at one-hour intervals. Some countries have time zones at a 30-minute difference from a neighboring zone. Certain towns in Islamic countries use solar time and thus can have time differences of several minutes between towns within one time zone.

Number Formats

The separators used with numerals to express quantities vary as part of national and personal preferences. In the United Kingdom and the United States, the comma is a thousands separator, and the period is a decimal separator. In continental Europe, the opposite is true. Separators include

1,234.56 1.234,56 1 234.56 1234,56
1'234.56

Numbers written in Japanese or Chinese using Chinese ideograms sometimes include the unit indicator, as in the number 28 [*] ("two", "ten", "eight") and sometimes omit it [*].

Positive and negative indicators differ. The plus and minus signs may be used before or after the number. In accounting, negative numbers are usually enclosed in parentheses.

Currency Formats

In currency formats, the currency symbol may be one or several characters and may be placed before or at the end of the number, or used instead of the decimal point. Some examples are: ÖS 2,50 (Austria); 2,50 \$ (French-speaking Canada); 2\$50 (Portugal); and \$2.50 (United States).

USER INTERFACE

As the point of contact between the user and the machine, the user interface is an obvious area for potential clashes of culture between the designer and the recipient. The interface must be localized to fit the cultural expectations of the end user.[15] The interface designer must be aware of issues of geometry management, images, symbols, color, and sound.

Geometry Management

Graphical interfaces in English use menu bars aligned at the left, with cascading menus falling from left to right. Menus in Hebrew and Arabic cascade from right to left. Figure 13 shows a menu from the Hebrew version of DECwindows XUI.

[Figure 13 (Hebrew DECwindows XUI) is not available in ASCII format.]

Although Japanese and Chinese are traditionally read from top to bottom with columns advancing from right to left, most technical material is presented with the same flow as English has. Consequently, user interfaces have the same left-to-right flow as English. This may be considered an aspect of new technology setting new cultural norms. Japanese and Chinese do present some geometry management challenges. A word processor for English uses the right scroll bar to advance from page to page. The analogy is from writing on a long scroll of paper, which is cut into pages. For a Japanese word processor, which enables the user to type in the traditional top-to-bottom orientation, does the bottom scroll bar control page advance by sliding the selection to the left? There is no one correct answer. A designer can keep consistent with the traditional horizontal scroll or with word processors for Latin-based writing systems.

Images

Some designers may consider that using images instead of text creates an international, culturally neutral product that requires no localization. This is only the case if the image is entirely abstract and chosen to be equally foreign to all cultures. This may meet the requirements of internationalization, but at the expense of good user interface design.

Most images are chosen to provide a cultural mnemonic to the action. This link may have little meaning in another culture. The rural mailbox image [*] chosen for certain electronic mail systems is a good example. This image is unknown outside the United States, and some American city dwellers are unfamiliar with it as well. The convention of raising the flag on the mailbox to indicate that new mail has arrived is not common throughout American rural communities. It can instead indicate the presence of outgoing mail.

In addition, a graphic may be a play on words that will not translate. One personal computer product uses a musical note to indicate that a written note is associated with an item in its database.

Symbols

Symbols commonly used in one culture may be misinterpreted by

someone from another culture. For example, the cross [*] is often used to indicate prohibition. However, in Egypt it does not have this connotation.[16] Designers should allow for the replacement of selection symbols such as ticks (checkmarks) and crosses found in many user interfaces.

[*] Italic

[*] Bold

Color

The significance of color varies greatly across cultures. Table 1, taken from Russo and Boor, gives the ideas associated with colors in six cultures.[17] For example, red means danger in the United States, but it has the connotation of life and creativity in India. Garland found that using a red "X" as a prohibitive symbol in Egyptian pictures was not effective because the color red is not associated with forbiddance, and the "X" is not understood as prohibitive.[16]

Table 1 Significance of Color across Cultures

	Red	Blue	Green	Yellow	White
U.S.	Danger	Masculinity	Safety	Cowardice	Purity
France	Aristocracy	Freedom Peace	Criminality	Temporary	Neutrality
Egypt	Death	Virtue Faith Truth	Fertility Strength	Happiness Prosperity	Joy
India	Life Creativity		Prosperity Fertility	Success	Death Purity
Japan	Anger Danger	Villainy	Future Youth Energy	Grace Nobility	Death
China	Happiness	Heavens Clouds	Ming Dynasty Heavens Clouds	Birth Wealth Power	Death Purity

Sound

In the book *Global Software*, Dave Taylor relates that when Lotus localized its 1-2-3 spreadsheet for use in Japan, the developers had to remove all beeps from the program.[18] Japanese users,

typically seated much closer together than their Western counterparts, did not appreciate the computer broadcasting to their colleagues every time they made an error. Since beeps can be irritating in open offices in all cultures, well-designed systems allow users to eliminate them or modify the volume.

FUNCTIONAL DIFFERENCES IN SOFTWARE

Much of this paper has covered areas where the form of the information must change for different cultures. The software may also require functional changes for different cultures. Applications that manipulate text provide a set of operations linked to the nature of both the writing system and the code set. We have seen that typing Japanese and Chinese requires an indirect input method. Applications using the Latin script provide a user interface to an operation to change the case of a character. This operation is not applicable to Japanese, but a Japanese word processor has an operation to convert from katakana to hiragana.

A delete operation on a Latin letter deletes both the letter and the rectangular cell, a piece of the screen real estate, causing the adjacent text to close up. With the cursor to the right of a Korean syllable cluster or Thai consonant/vowel/tone combination, the user presses the delete key. What should be deleted? Thai and Korean do not have the union between a letter and its linear space that the Latin alphabet has. Two separate operations with different user interfaces may be required, whereas one suffices in English. The code set used also plays a part in determining the nature of the operation. The Thai code set independently codes every letter and tone, so deleting a single letter or tone is practical. The national Korean code set codes syllable clusters.[19] Deleting one letter from a cluster may produce a combination with no code. In Digital's Thai and Korean products, the action of the delete operation is as suggested by the code set. Thai deletes one letter or tone mark; Korean deletes the syllable cluster.

In unidirectional writing systems, the right arrow key navigates the cursor over the logical reading order of the text as it moves smoothly over the screen. The operation of logical movement and geometrical movement across the screen is identical within one line. This is not the case with bidirectional text. The following fragment is from a Hebrew application [*]. Pressing the left arrow key moves the cursor to the left of the word "one" if the action means to follow reading order, or to the left of the "o" in "two" if the action is one of navigating screen real estate.

Functional differences may come from regulatory constraints. The United States has export prohibitions on certain encryption techniques. Non-U.S. versions of products may need to remove them or use different techniques. Standards and regulations for

connection to external devices such as modems vary around the world.

Product features may also need to vary based on less tangible aspects of a culture. LYRE is a hypertext product developed in France. The product allows students to analyze a poem from various viewpoints selected by the teacher. Students are not allowed to add their own viewpoints. This is acceptable in France but not in Scandinavian countries where independent discovery is highly valued.[15]

Correct and Incorrect Actions

Learning the rules concerning cultural sensitivity does not guarantee that a software developer from outside, or even inside, that culture will not make errors. Two examples illustrate this.

When Lotus localized its 1-2-3 product into Japanese, the developers were aware that the Japanese date counts the year from the ascension of the emperor to the throne. In their initial test of the product under development in Japan, they included the ability to reset the counter and to modify the field naming the reign. This appears to be admirable planning, sensitive to the needs of the local date format; however, the Japanese users strenuously requested that this feature be removed since it anticipated the demise of the emperor.[20]

In Arabic and Hebrew bidirectional text, deletion of one segment of text can cause the surrounding segments to be rearranged under certain circumstances. This follows from a logical analysis of ordering of the segments and was implemented in an early version of Hebrew DECwrite. Studies with users revealed that they found this rearrangement of text disconcerting and preferred to manually rearrange segments. The program was changed in a subsequent version. Note that this resolution is dependent on the specific product. One should not conclude that automatic reordering of text is always incorrect. Other bidirectional text systems perform this reordering.

RESPONDING TO AND SETTING CULTURE

New technology in computer applications must reflect the prevalent existing culture, but it also plays a part in creating new cultural norms. An earlier section described how users of a Hebrew word processor might enter digits into a stream of Hebrew by reversing the order of the digits. This cultural behavior was introduced during the days of manual typewriters or older computer systems, which required additional keystrokes to change writing direction. An older technology introduced a cultural expectation. As users in Israel grow more accustomed to word processors that enter the correct order automatically, and as the base of users exposed to older technology shrinks, we can

anticipate that the standard expectation of the order in which to enter digits will change.

The Arabic and Khmer writing systems modify the shape of the written glyph based on surrounding characters. The Khmer keyboard (Figure 10) shows separate glyphs for each variation (implying separate codes). This design follows the lead of earlier typewriters and is familiar to users trained on such typewriters. It adds complexity to the keyboard and requires the user to manually enter the correct glyph. The Arabic keyboard is from a system that codes each character independently of glyph; the renderer selects the correct glyph to display based on context. This system may require a longer transition for users trained on manual typewriters, but it is the preferred use of a more advanced technology.

As described previously, written Thai and Korean both use syllable clusters, but the delete operation on each script differs due to the different methods by which the code set represents the script. Which is the correct action? The question does not have an easy answer. From a formal analysis of the language, one might argue that deleting the individual letter is correct; but as we have seen, formal analysis need not yield an appropriate answer. Ultimately the correct answer is a delicate balance between users' expectations based on the past and the requirements of innovation. The users' expectations are set by previous implementations, which were derived from limitations in the technology of the time. We have a cycle of computers adapting to people adapting to computers.

DEEPER CULTURAL DIFFERENCES

Some of the cultural differences discussed in this paper such as the presentation of dates and currency are obvious even on a superficial examination of the culture. Others such as the cultural reaction to color are learned from deeper study.

We can expect the future development of software to consider as yet unexplored cultural differences. New features in user interfaces, the use of sound, voice, pen-based computers, and animation, will tie into aspects of cultural behavior that are currently little explored by researchers. Higher-resolution screens and the prevalence of color bring the ability to design applications that relate more directly to the user's sense of beauty.

The personal computer revolutionized personal productivity. Applications such as spreadsheets succeeded because they modeled individual user's existing work practices and extended their capability. A current trend is toward applications for the work group or collaborative computing. This style attempts to revolutionize the way groups work. Jeffrey Hsu reports that "Collaborative systems can meet stubborn resistance when they are

introduced in a company, because they challenge the organizational culture with a new means of communication." [21] The difference in the business decision-making process between Japan and the United States is well documented, with Japanese groups valuing group decision and harmony or wa highly. We can expect the emerging "groupware" applications both to model existing styles of group work and to change those styles.

The future will also bring software agents. [22] This software will act as a collaborator with the user to process information in much the same way as a human personal assistant. As with a human assistant, we can anticipate that software agents will adapt to the specific requirements and habits of the user, a culture of one. We can imagine an agent recommending circulation lists for memos and aiding in correctly phrasing the mail. The forms of address will vary not only across national boundaries, but across companies. As the set of cultural differences to be addressed goes deeper, the circles of people sharing those cultures will shrink.

Techniques exist to build products with a high level of internationalization. These are described in other papers in this issue. These techniques will continue to develop and improve, but internationalization will never be a fully resolved consideration. The term may fall from use as the cultural differences being addressed have a decreasing relationship to national boundaries. Internationalization is simply making software easy to localize, and the essence of localization is meeting the individual needs of the customers. As computer systems become more powerful and software more sophisticated, adaptation to the individual will continue. Techniques to adjust software to fit personal preferences will continue to develop.

ACKNOWLEDGMENTS

This paper is adapted from an earlier unpublished work, circulated within Digital Equipment Corporation. The author would like to thank Gayn Winters for initiating and driving the paper and for many valuable comments on early drafts. This paper took shape as reviews from colleagues around the world corrected errors and contributed many examples, some of which are seen in this paper. The author would like to acknowledge the input from John McConnell and Michael Yau in the United States; Jürgen Bettels in Switzerland; K.H. Chan and Fred Li in Hong Kong; Trin Tantsetthi in Thailand; Mike Feldman, Moti Huberman, and Moshe Loterman in Israel; Hirotaka Yoshioka in Japan; and Nai-peng Kuang in Taiwan.

REFERENCES AND NOTES

1. South Korean writing uses two scripts. Hangul is an alphabetic system. Hanja is the set of ideograms imported

from China and used as the sole script until the invention of Hangul. There exists a widespread misconception that Hangul is ideographic. The author wishes to stress that only the Hanja script uses ideograms.

2. The example shown contains no phonetic element. Many, more complex characters do have phonetic components. Some scholars disparage the use of the term ideograph to describe Japanese and Chinese writing, asserting that the phonetic element is primary. (See references 3 and 5.) This paper uses the term ideograph since it is in common use.
3. J. DeFrancis, *The Chinese Language Fact and Fantasy*, Second Paperback Edition (Honolulu: University of Hawaii Press, 1989): 91.
4. M. Stubbs, *Language and Literacy. The Sociolinguistics of Reading* (London, Boston, and Henley: Routledge and Kegan Paul, 1980): 48.
5. J. DeFrancis, *Visible Speech: The Diverse Oneness of Writing Systems* (Honolulu: University of Hawaii Press, 1989).
6. & is an interesting character. It was originally formed as a ligature of e and t and is now used as an ideogram in many European written languages.
7. F. Coulmas, *The Writing Systems of the World* (Oxford: Basil Blackwell, 1989): 118.
8. A. Nakanishi, *Writing Systems of the World*, third printing (Rutland, VT and Tokyo: Charles E. Tuttle Company, 1988): 112.
9. The numerals 1, 2, 3, etc. are generally known as, and referred to, as Arabic numerals; however, by one of those quirks of language, the Arabic script uses a different set of symbols for numerals, sometimes called Indic numerals.
10. Note that the Khmer keyboard has four registers because it is based on a glyph encoding of Khmer rather than a character encoding, which would use two registers at most. Also, the subscript Khmer glyphs on the keycaps, which are used in conjuncts, are not necessary if more sophisticated display software is used.
11. DIN 2137, *German keyboard for typewriters, Allocation of Characters to Keys, Parts 1, 2, 6, and 11* (Deutsch Institut für Normung, 1988).
12. T. Honma, H. Baba, and K. Takizawa, "Japanese Input Method Independent of Applications," *Digital Technical Journal*, vol. 5, no. 3 (Summer 1993): 97-107.

13. There is some dispute in the industry on the need for explicit direction markers. Under certain circumstances, correct rendition of nested direction text can be computed. For example, a renderer could show the structure of the text in Figure 8 correctly without directional attributes. The Unicode and ISO 10646 character sets do include a rich set of directional markers.
14. J. Bettels and F. Bishop, "Unicode: A Universal Character Code," *Digital Technical Journal*, vol. 5, no. 3 (Summer 1993): 21-31.
15. J. Nielsen, "Usability Testing of International Interfaces," in *Designing User Interfaces for International Use*, edited by J. Nielsen (New York: Elsevier, 1990).
16. K. Garland, "The Use of Short Term Feedback in the Preparation of Technical and Instructional Illustration," in *Research in Illustration: Conference Proceedings Part II* (1982).
17. P. Russo and S. Boor, "How Fluent Is Your Interface? Designing for International Users," paper presented at INTERCHI, Amsterdam, April 1993.
18. D. Taylor, *Global Software: Developing Applications for the International Market* (New York, Berlin, Heidelberg, London, Paris, Tokyo, Hong Kong, Barcelona, Budapest: Springer-Verlag, 1992): 54.
19. The design of the Korean code set reflects compromises made among cultural, economic, and technological requirements. The structure of the writing system leads to independent coding of each jamo, with the display device rendering them into syllable clusters. Coding as syllable clusters greatly simplified the prevalent technology of the time and reduced the cost of the display device.
20. F. Hapgood, "A Journey East--The Making of 1-2-3 Release 2J," *Lotus: Computing for Managers and Professionals* (Cambridge, MA: Lotus Development Corporation, 1987).
21. J. Hsu and T. Lockwood, "Collaborative Computing," *BYTE Magazine* (March 1993): 120.
22. L. Tesler, "Networked Computing in the 1990s," *Scientific American* (September 1991).

TRADEMARKS

DECwindows, DECwrite, and Digital are trademarks of Digital Equipment Corporation.

Apple is a registered trademark of Apple Computer, Inc.

Lotus 1-2-3 is a registered trademark of Lotus Development Corporation.

BIOGRAPHY

Timothy G. Greenwood Since 1981, Tim Greenwood has held various positions related to internationalization at Digital. He was the architect for the Japanese and Chinese versions of DECwindows. This software introduced the compound string technology that was incorporated into Motif. Tim conceived of, managed, and wrote much of the software section of the internal version of the handbook on Producing International Products. He also participated in the design of international support on the X Window System. Tim is currently responsible for guiding the introduction of Unicode into Digital.

=====
Copyright 1993 Digital Equipment Corporation. Forwarding and copying of this article is permitted for personal and educational purposes without fee provided that Digital Equipment Corporation's copyright is retained with the article and that the content is not modified. This article is not to be distributed for commercial advantage. Abstracting with credit of Digital Equipment Corporation's authorship is permitted. All rights reserved.
=====