Five Fallacies of Home Computing

Although microcomputers have become essential to small-business operations, they are often a disappointment at home. Many of my friends who have bought computers strictly for home use regard their machines with a mixture of pride and guilt, much the same way they might treat a health club membership, a collection of the Hundred Great Books, or their certificate from an EST seminar. It's something expensive and worthwhile that somehow they have never taken advantage of.

The reason for this is that we learn about computers from advertisements and magazines that perpetuate what I like to call the Five Fallacies of Home Computing.

• Fallacy #1:
Kids love computers.

The computer is a wonderful educational tool, but don't buy one for this purpose! Time and again I've seen concerned middle-class parents buy a computer plus excellent educational software and watched their delight as the child plinks away...for a few weeks. Then I've witnessed their disappointment as the child gradually loses interest and drifts back to TV or the playground.

Kids don't love computers—they love toys. Without game software, a computer is not a toy. And because kids don't read computer literature, they (unlike their parents) don't believe:

• Fallacy #2:
Education is fun.

Sometimes it's fun. Every child has at least a few fascinating interests and a few good teachers who help make learning fun. But in the end, education requires a certain amount of tedium, drill, and practice.

No sensible parent would lay down $1000 for a piano, sit Junior down at the keys, and walk away. Except in the case of the rare prodigy, this is not enough. To instill a love of music, the parent must spend a great deal of time entertaining the child, playing the piano for him or her, going to concerts, cajoling, nagging, exhorting, and otherwise communicating enthusiasm and encouragement.

No one would argue with this, but sticking a child in front of a computer is no different. A parent must be just as intensely involved if the child is to benefit.

• Fallacy #3:
A computer will organize your life.

Recently I reviewed a home accounting program. When you enter your income, expenses, checks, bills, credit-card purchases, interest and dividend statements, and so forth, your entire financial life becomes clear as crystal. A few keystrokes will call up the month's budget, balance your checkbook, analyze your stock portfolio, generate a list of tax-deductible expenses, and print your checks.

Because my finances are not simple, I might use this sort of program. In addition to my freelance income, I have a part-time salaried job at a university and a one-man medical business. I have three checking accounts, two savings accounts, an IRA, and sundry insurance policies, credit cards, and business expenses. But I'm well organized—I have no trouble balancing my checkbooks or paying bills on time. I doubt if I spend an hour a month keeping things straight.

My wife, who runs a small business, is another story. She often puts off balancing her checkbook to the point that it requires an afternoon of agony. Bills, receipts, and even incoming checks are thrown in a pile, and some lost forever.

At the beginning of every new year, I start in on a three-month campaign of coaxing, wheedling, pleading, abuse, and threats that lead inevitably to a frantic weekend yielding a page of tax data.

Which of us would benefit from a home accounting program? Me, of course. An obsessive type, I'd have no trouble sitting at the computer to enter the necessary updates. My wife would never do that. The moral is that if you're already organized, a home computer will make you more so. If you're not organized to begin with, a computer will no more organize your accounts than a library card will make you literate.

• Fallacy #4:
A computer will save you money.

Nonsense—it will cost you plenty. But that's OK; it's like my recent purchase of a $900 washer and dryer after five years of hoofing it to the laundromat. Even with the most liberal cost analysis, this purchase is a loser. But I love it.

• Fallacy #5:
A computer will save you drudgery.

Sometimes, but if used properly, it also requires a lot of drudgery. Take, for example, the superb database program that houses my journal articles. It's far superior to a simple file cabinet because I can classify articles under a dozen different headings and retrieve them with a couple of keystrokes. But it required about 50 hours of unrelieved, boring data entry to reach this state of grace.

So home computers are not likely to save you any appreciable time or money. A computer by itself offers no magic that will educate your children or organize your life, although it can help you accomplish these goals if you invest time and effort. But one claim about home computers is not a fallacy—the machines are lots of fun.

—Michael Oppenheim

Michael Oppenheim is a physician and freelance writer living in Los Angeles.

Illustration by John O'Leary

Multilingual Word Processing

The advantages of computerized typing and editing are now being extended to all the living languages of the world. Even a complex script such as Japanese or Arabic can be processed

by Joseph D. Becker

The personal computer has become a familiar fixture in the office and even in the home. It is useful in many ways, and yet for all its novel applications it is probably most useful to the greatest number of people when it serves the function of a typewriter. In that role it enables the user to see text displayed on a screen, so that words can be reviewed and revised before they are ever committed to paper. The kind of computer program underlying such an ability is known as word-processing software.

So far computers have largely been limited to the processing of words in the English language. That is not surprising: most computers have been developed in English-speaking countries, and English is the principal language of international commerce. Yet there is no technical reason for word processing to be confined to English. Indeed, it is possible for word-processing software to handle not only French, German, Italian, Russian, Spanish and other European languages but also more complex scripts such as Arabic, Chinese, Hebrew, Japanese and Korean. My colleagues and I at the Xerox Corporation have been developing multilingual word-processing software for a personal computer work station called Star, which is manufactured by Xerox. Our basic idea is that the computer should deal with a universal notion of "text" broad enough to include any of the world's living languages in any combination.

In effect, therefore, the fascinating diversity of mankind's written symbols must be made to coexist in the computer. At first it hardly seems possible. Arabic script, for example, flows from right to left in curlicues. Thai and other scripts, originally from ancient India, have letters that sometimes step around their neighbors and thus get out of phonetic order. Occasionally a letter even surrounds its neighbors. Korean groups its letters in syllabic clusters. (The Korean alphabet was designed from scratch by a group of scholars in 1443.) Chinese, the most ancient of living writing systems, consists of tens of thousands of ideographic characters. Each character is a miniature calligraphic composition inside its own square frame. It seems the developers of the computer and of word-processing software were coddled by the English language, which happens to have the simplest writing system of all: unadorned alphabetic letters laid out one after the other.

How can computer software originally designed to handle only English text be broadened to encompass the full diversity of the world's writing systems? The many challenges of the task can be divided into three basic realms. There must be a way for text to be represented in the memory of a computer; there must be a way for text to be typed at the keyboard of the computer; there must be a way for the computer to present text to the typist. I shall refer to these realms as encoding, typing and rendering. By rendering I mean both the display of text on the screen of a computer and the printing of text on paper.

Encoding is governed by a single, basic fact: the computer can store only numbers. Indeed, it can store only binary numbers, consisting of strings of 0's and 1's. Hence text is represented in a computer by storing a binary code number for each letter. In the case of the English language the American Standard Code for Information Interchange, abbreviated ASCII, assigns the binary code number 01000001 to the letter *A*, 01000010 to *B*, 01000011 to *C* and so on. Thus when you type an *A* on a computer keyboard, the computer is really being instructed to store the code number 01000001. When the computer comes to display or print a letter encoded as 01000001, its instructions cause it to draw a symbol you recognize as an *A*. As long as the input and output instructions are consistent, you have the illusion that the letter *A* itself was stored.

Computers generally store information in units of bytes, where each byte is a group of eight bits. It therefore seems a sensible strategy to store text as one byte per character. The trouble is, there are only 2^8, or 256, ways in which eight 0's and 1's can be combined in a byte. The living scripts of the world have far more letters than that. A two-byte coding scheme, in which each letter would be identified by two successive bytes, would yield 2^{16}, or 65,536, possible codes; a three-byte scheme would yield 2^{24}, or well over 16 million, codes. But employing two or three bytes per letter where only one byte is needed would waste space in the computer's memory. The answer is to arrange for the encoding to expand to two or three bytes per letter only when necessary. This can be done by setting aside a few bytes as signals to the computer and putting those signals into encoded text.

The first step in establishing a scheme for multilingual text encoding is to assign a binary code number to each of the alphabets of the world. The Roman alphabet is assigned 00000000, Greek 00100110, Russian 00100111, Arabic 11100000 and so on. (The particular choices, like the choices for individual letters, are based on international standards.) Next the code 11111111 is designated the shift-alphabet signal. The

MULTILINGUAL TEXT occupies much of this "screen dump," which shows the content of the display screen of a computer. At the upper right is a "virtual keyboard." It establishes that the Arabic alphabet has been assigned to the standard keyboard attached to the computer. Pressing the *g* key, for example, stores the Arabic letter *lam*. Below the virtual keyboard is a document in Arabic and English. The former reads from right to left, the latter from left to right; the computer enables the typist to embed one language in the other while preserving the directionality of each script. Below the bilingual text is a set of "icons" that represent facilities such as printers. The left is occupied by a document that shows symbols available to the typist.

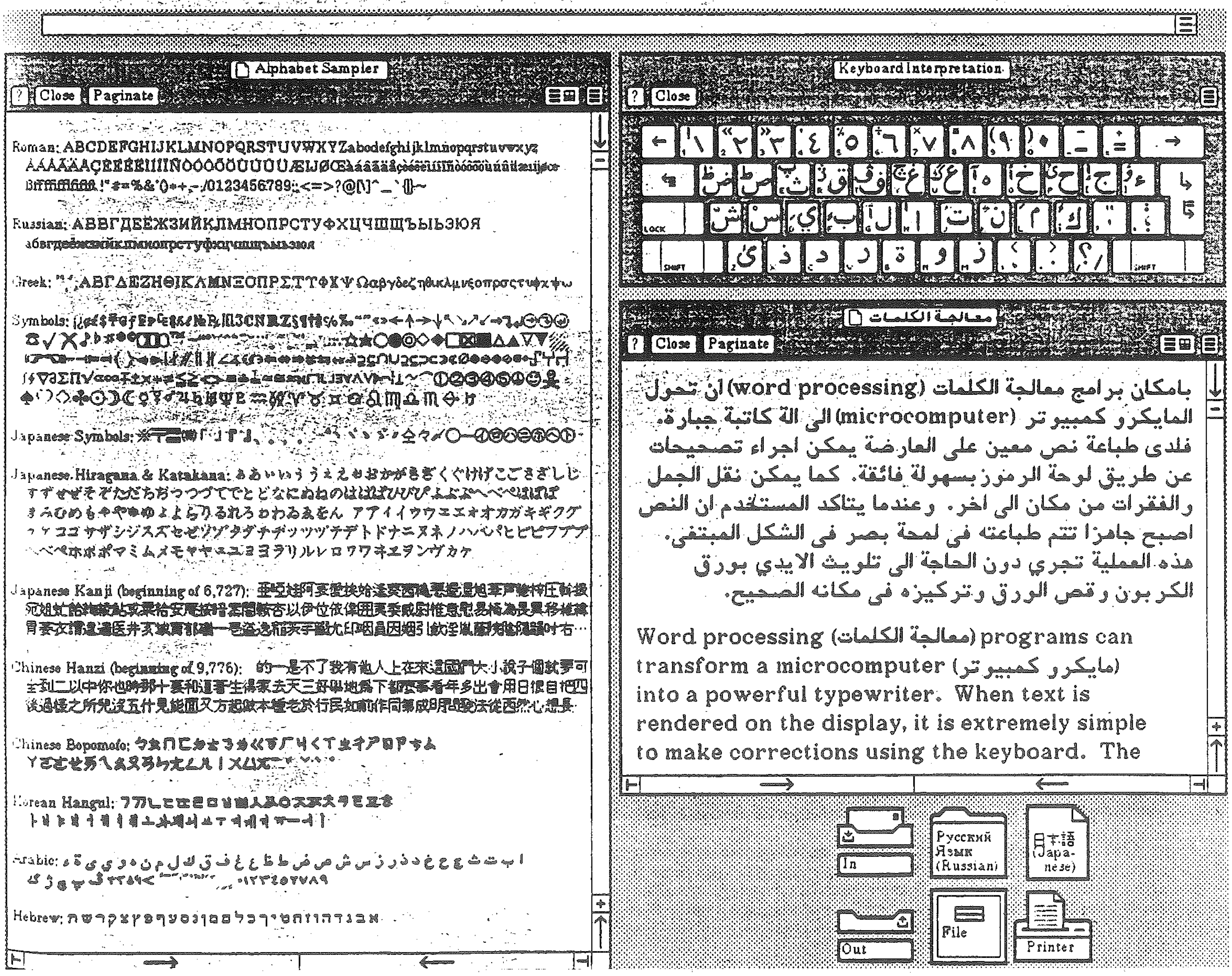
computer is to start by assuming the text is in alphabet 00000000—that is, in Roman script—but whenever it encounters 11111111, it is to interpret the next byte as the code specifying a new alphabet and the succeeding bytes as codes for letters in that alphabet [*see top illustration on page 100*].

That solves the text-encoding problem for the world's phonetic alphabets. The Chinese ideographs remain: they push the total number of letters above the 65,536 the scheme can encompass. To accommodate all the ideographs one must create another level in the encoding hierarchy. One can consider a group of 65,536 letters to be a "superalphabet" and specify two bytes of 11111111 in succession to be a shift-superalphabet signal. The signal causes the computer to interpret the following byte as the code number of a new superalphabet and succeeding pairs of bytes as codes specifying symbols in that superalphabet. The main superalphabet, designated 00000000, is all one needs except for very rare Chinese characters.

This strategy of "flexible encoding," which was devised by Gael Curry of Xerox's Office Systems Division, opens the computer to a range of more than 16 million characters, including all Chinese ideographs. At the same time it optimizes the storage of text by encoding ordinary (that is, phonetic) alphabets with a single byte per letter. It allows text in any mixture of living languages to be represented economically in the computer as a sequence of bytes.

The sequence of bytes is stored in the linear order in which the text would be spoken, and as such it is isolated from graphical complexities such as the variant forms of letters and the mixing of the directions in which multilingual text might have to be written. This accounts for a fact that may seem surprising: the internal computer processing of multilingual text is not affected by the presence of exotic scripts. Word-processing operations such as the editing of text, the search of text for particular characters or words and the electronic transmission of text depend not on the graphical form of the text but on the internal sequence of bytes that represents its information content. The only real complexities in multilingual word-processing software involve the typing and rendering of text.

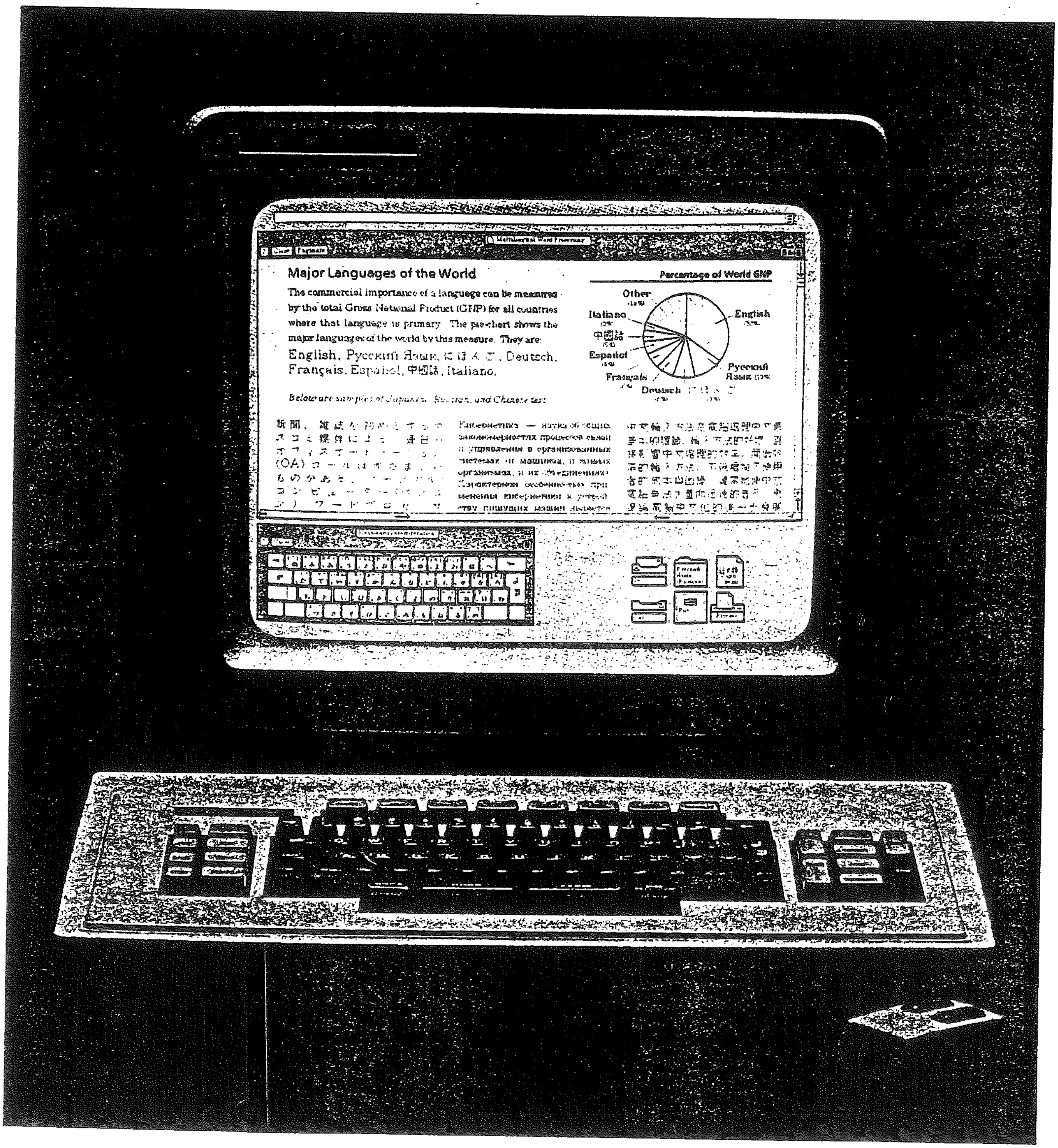
I turn next to typing. For most languages the process can be quite simple, since almost any living alphabet will fit comfortably on a standard typewriter keyboard. Indeed, in computerized typing it is easy for the software to change the computer's "interpretation" of the keys so that the typing is in another alphabet. For example, pressing the *A* key can cause the computer to store the Russian letter Ф or the Arabic letter ش, depending on the keyboard interpretation. The computer can even display on its screen a small diagram called a virtual keyboard, which reminds the typ-



ist what alphabet is currently assigned to the keyboard and which keys correspond to which letters.

For some languages the computer can simplify typing itself. Arabic script, for example, includes a special combination character that replaces the letters *lam* and *alif* whenever the two appear in sequence. On an Arabic typewriter the special character occupies its own key. The computer can automatically render all *lam-alif* combinations as the special character, making the extra key superfluous. The computer can also handle letters that change their appearance depending on their context. With word-processing software for such a script the typist need only enter a natural (that is, phonetic) sequence of characters; the computer will take over the burden of handling the complexities of the script.

There is one living script whose complexity challenges even a computer's capabilities. It is the ideographic system employed in writing Chinese, Japanese and occasionally Korean. The Chinese ideographic characters, which can conveniently be called by the Japanese term *kanji*, originated as pictographs more than 4,000 years ago and assumed their



COMPUTER WORK STATION includes the screen whose content was shown in the illustration on the preceding page. The computer itself is part of a computer system called Star manufactured by the Xerox Corporation. The space bar on its keyboard is actually four space bars. In Japanese word processing they specify three Japanese "alphabets" (*katakana, hiragana* and *kanji*) or English letters. In front of the keyboard is a "mouse" connected to the computer. The mouse is moved about on a desktop to position a pointer on the video screen.

present form some 2,000 years later. About 500 years after that the Japanese and the Koreans adopted the Chinese language for official and scholarly discourse. For writing the native language they devised phonetic letters: the *kana* (*hiragana* and *katakana*) characters in Japan and the *hangul* in Korea. The Chinese characters were unsuitable from the beginning for writing the native language; now they were also unnecessary. Nevertheless, the prestige of Chinese culture led to hybrid scripts. Today the phonetic alphabets in Japan and Korea are used mainly to write the inflectional endings of words (endings somewhat similar to English inflections such as *-ed* and *-ing*). *Kanji* are used to write word roots, that is, the basic dictionary form of words. In Japan the number of *kanji* commonly seen in published text is declining: it has been cut to about 3,500, about half the number commonly seen in China.

As a result of its curious history the Japanese language has the most complex script in the world: it remains a mixture of *kanji* ideographs and *kana* phonetic letters. Because of this complexity, the Japanese have had no reasonable way to type their own language; more than 90 percent of all documents in Japan are handwritten, or rather handcrafted. A slip of the writer's hand, and a page must be torn up. Moreover, most documents are hard to read unless the writer happens to be an accomplished calligrapher. To be sure, there does exist a *kanji* typewriter, rather like a small typesetting machine, but the device is slow and tiring to use. Professional typists are comparatively rare, and their productivity is typically about 20 characters per minute, or only 10 pages per day.

Hisao Yamada of the University of Tokyo, a scholar of the social history of typing, notes that it is not easy for a society to envision a usable typing system where none has existed. In the U.S. the possibility of high-speed typing never occurred to the inventor of the typewriter; touch-typing (with all 10 fingers potentially in contact with the keyboard) was not devised until 14 years after the typewriter was patented. In Asia the situation is much the same. The initial methods for entering *kanji* into a computer were not well conceived. Some methods employ a keyboard with several hundred keys, where each key has several characters on it. The right hand presses such a key while the left hand manipulates a bank of shift keys to specify a particular character. Other methods require the typist to analyze each *kanji* character and then enter some sequence of code symbols. This turns out to be even more tiresome than searching among hundreds of keys.

In recent years a new *kanji* typing

WORLD'S LIVING LANGUAGES raise problems for multilingual word processing. Mongolian (*a*) is in vertical columns. The specimen reads: *Erte togha tomshi ugei, nogchigsen galab-un urida anu*, or "Once upon a time, countless past ages ago. . . ." Arabic (*b*) is written from right to left in descending curlicues. The specimen reads: *najmu-l-fajri*, or "star of dawn." Hebrew (*c*) is written from right to left in letters that carry markings. The word is *ba'asher*, or "where." Greek (*d*) includes a letter, *sigma* (*colored arrows*), that has a special form at the end of a word. Here the word is *aschimos*, or "ugly." Arabic scripts (*e*) require the combining of certain letters. The word is *al-islam*, or "Islam." Hindi (*f*) has letters written out of phonetic order. The word is *hindī*; the spelling, in effect, is *ihndī*. Thai (*g*) has vowels surrounding consonants. The word is *deuan*, or "month"; the letter *eua* is in three parts. Korean (*h*) is characterized by syllabic clusters. The specimen reads: *han-gug-ŏ*, or "Korean language." Chinese (*i*) is written with thousands of ideographs. The specimens both read: *zhong-guo-hua*, or "Chinese language."

a

CODE NUMBERS	01000001	01000010	01000011	01000100
MEANING TO COMPUTER	A	B	C	D

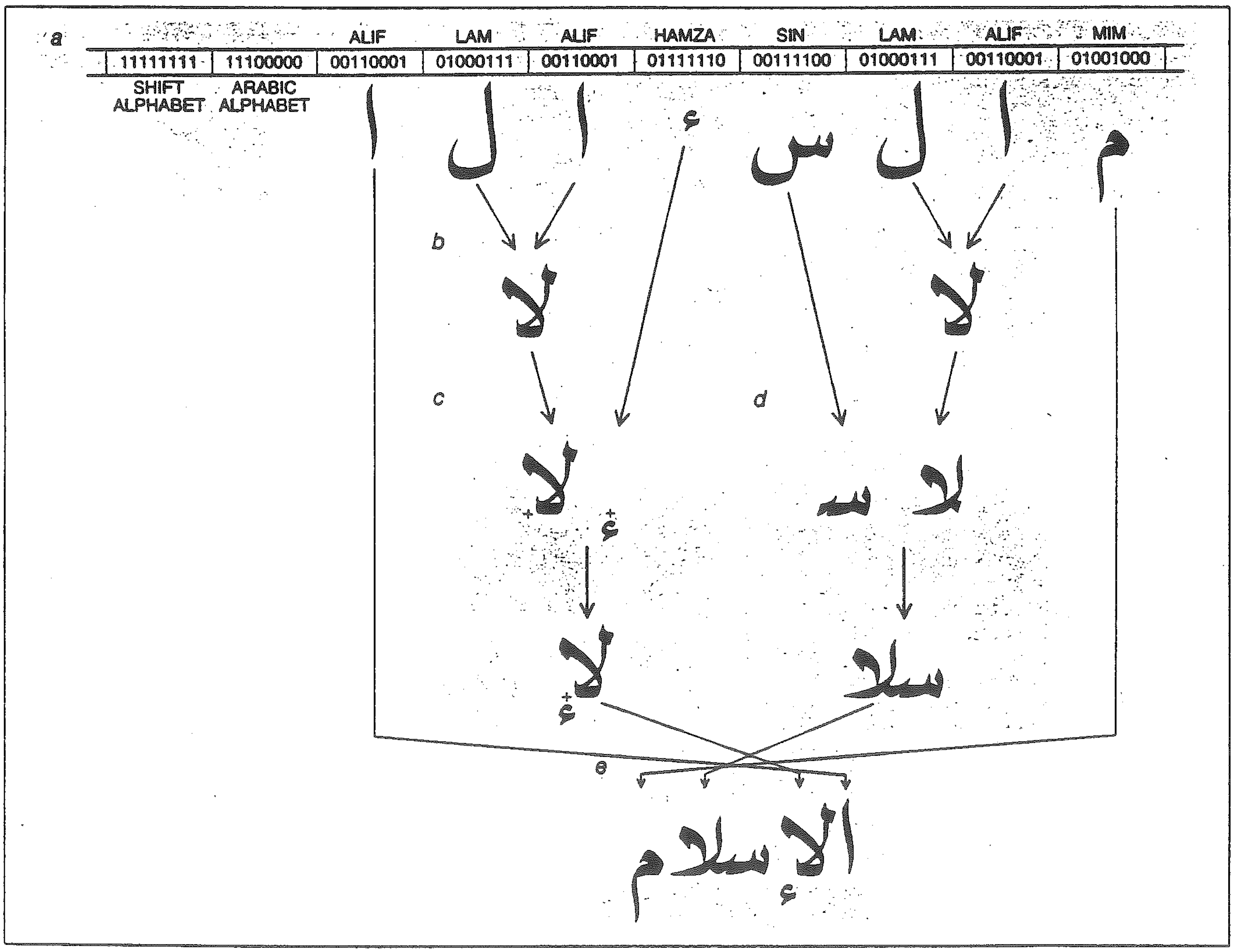
b

CODE NUMBERS	11111111	00100111	00100001	00100010	00100011	00100100
MEANING TO COMPUTER	SHIFT ALPHABET	RUSSIAN ALPHABET	А	Б	В	Г

c

CODE NUMBERS	11111111	11111111	00000000	01000110 01111100	01001011 01011100	00111000 01101100
MEANING TO COMPUTER	SHIFT ALPHABET	SHIFT SUPER-ALPHABET	MAIN SUPER-ALPHABET	日	本	語

FLEXIBLE ENCODING accommodates all the world's writing systems while minimizing the length of the string of bits, or binary digits (0's and 1's), that represents a multilingual text in the memory of a computer. For English (*a*) a standard scheme, the American Standard Code for Information Interchange, is available. It assigns one byte, or eight bits of code, to each character. For other languages, such as Russian (*b*), the byte 11111111 is designated a "shift alphabet" signal: it instructs the computer to interpret the following byte as the code specifying a new alphabet, and the bytes after that as codes specifying characters in that alphabet. For Chinese and Japanese (*c*) a still more complex scheme is required. Two successive bytes of 11111111 are a "shift superalphabet" signal, the following byte specifies a particular superalphabet and the subsequent bytes, taken in pairs, specify characters in the superalphabet (here ideographs).



RENDERING OF TEXT means the process of displaying text on a screen or printing it on a page. The rendering of the Arabic word *al-islam* suggests some of the complexities. In storage in the computer's memory the word is simply a sequence of eight letters (*a*). Their proper rendering requires some special computations. First the computer finds two instances in which *lam* and *alif* are neighbors; Arabic script requires that this combination be replaced by a single ligature that looks much like the Greek letter *gamma* (*b*). The computer "notes" this in temporary memory; the stored code for the original text is unchanged. Next the computer searches for small markings such as *hamza*, which have to be positioned above or below other letters. The small crosses in the illustration are registration marks (*c*). Third the computer searches for letters that must be joined to their neighbors. The *sin* and its neighboring *lam-alif* ligature will join, and so they are given special joining forms (*d*). Finally, the computer displays the word or prints it on paper, in proper right-to-left order (*e*).

method has emerged; it suits the average typist so well that it is fast becoming the standard for Japanese manufacturers. Indeed, it makes possible touch-typing at a rate of about 50 characters per minute for the average typist and 150 characters per minute for the best speed typists. Called phonetic conversion, it has three steps that enable the Japanese to type their language phonetically. The typist types a word in phonetic *hiragana* symbols. (This phonetic spelling is learned in childhood by the Japanese.) To change the spelling to *kanji* the typist presses a special "look-up key" and the computer finds the *kanji* spelling in a dictionary it holds on a magnetic storage disk. If several words share the same phonetic spelling or if the word has several *kanji* spellings (this happens about 40 percent of the time), the computer assigns each alternative to a key on a virtual keyboard, and the typist chooses the one wanted.

Phonetic-conversion typing employs a standard typewriter keyboard. The *hiragana* characters fit neatly on such a keyboard; in fact, there is a standard arrangement for them. Yet many Japanese are familiar with the English typewriter keyboard and do not want to have to learn another arrangement. In consideration of their preferences many phonetic-conversion systems offer a *hiragana* typing method called *romaji* conversion. *Romaji* signifies the use of Roman letters to spell Japanese words. *Fuji, sayonara, samurai* and a host of other words are *romaji* spellings well known to Westerners. Most Japanese know them too. To type the word *sushi*, therefore, the typist can simply type *s u s h i*. The computer converts this first into *hiragana* すし and then into *kanji* 寿司. Offered a choice between typing *hiragana* on a standard Japanese keyboard and typing *romaji* on an English keyboard, nine out of 10 Japanese users prefer the latter.

The second step in phonetic-conversion typing—the computer's search in a dictionary—is greatly complicated by the inflectional endings of the Japanese language, which are somewhat similar to those of a European language. (For example, *aruku* means "walk," *aruita* means "walked" and *aruite* means "walking.") The dictionary may contain well over 100,000 word roots. The inflected forms of those words would number in the millions. Plainly the computer cannot store them all. The problem is solved by software that draws on a grammar of Japanese inflections to analyze the phonetic spellings supplied by the typist [*see illustrations on pages 102–104*]. In essence the typewriter is made to know the complete grammar of the language typed on it.

The chief advantage of phonetic-conversion typing is that the typing is analyzed by the computer word by word rather than one *kanji* at a time. Consider the nature of *kanji*. Although each *kanji* character signifies a separate concept, most words in Chinese, Japanese and Korean are compounded from two such concepts (and therefore from two *kanji*), much like English words such as *blackboard* and *railroad*. The compounding allows the creation of far more words than there are *kanji*. In addition it reduces the phonetic ambiguity of the language. For example, a Japanese dictionary lists 64 common *kanji* pronounced *tou* and 53 common *kanji* pronounced *kyou* but only one compound word pronounced *toukyou*, namely 東京, or "Eastern capital," the name of the city Tokyo. In the face of the alternatives each *kanji* in the compound word uniquely determines the other.

a

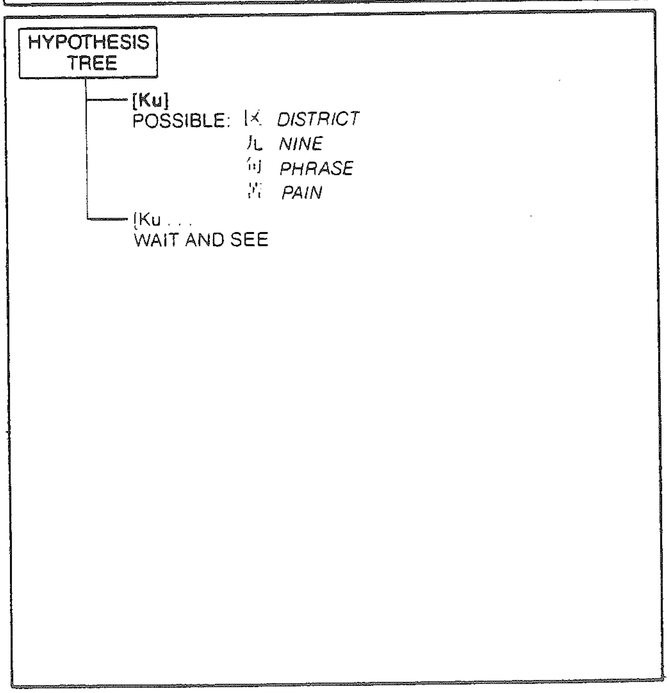
ku → く

WORD ROOT DICTIONARY

[Ku]

KANJI	PART OF SPEECH	RELATIVE FREQUENCY	*ENGLISH MEANING*
区	NOUN	8	*DISTRICT*
九	NUMBER	7	*NINE*
来	IRREGULAR VERB STEM	6	*TO COME*
句	NOUN	5	*PHRASE*
苦	NOUN	4	*PAIN*
繰	VERB STEM	1	*TO SPIN*

[Ku . . . MANY ENTRIES IN DICTIONARY



b

kuru → くる

WORD ROOT DICTIONARY

[Ku-ru]

KANJI	PART OF SPEECH	RELATIVE FREQUENCY	*ENGLISH MEANING*
狂	VERB STEM	5	*TO GO CRAZY*
包	VERB STEM	2	*TO WRAP*

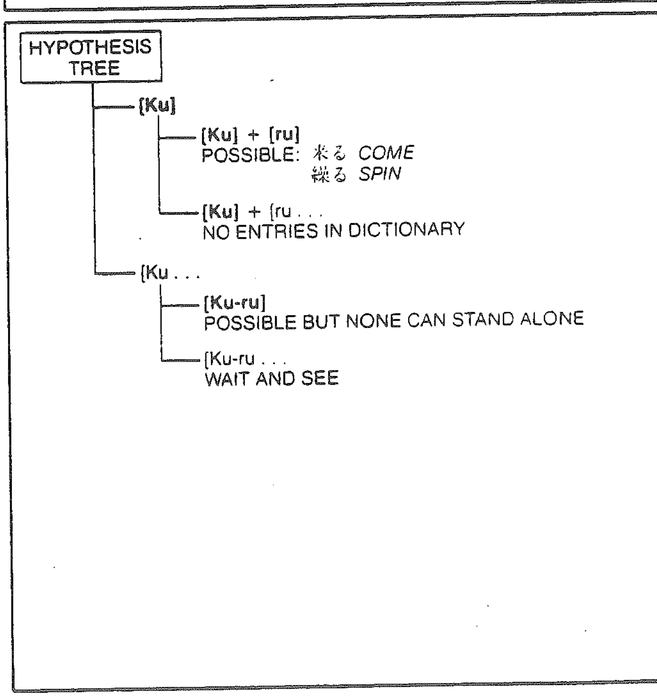
[Ku-ru . . . MANY ENTRIES IN DICTIONARY

INFLECTION DICTIONARY

[ru]

HIRAGANA	PART OF SPEECH	RELATIVE FREQUENCY
る	VERB ENDING	9
る	IRREGULAR VERB ENDING	9

[ru . . . NO ENTRIES IN DICTIONARY



JAPANESE WORD PROCESSING challenges a computer because the Japanese language is a complex mixture of thousands of *kanji* (ideographic symbols borrowed from the Chinese) and dozens of *kana* (indigenous phonetic symbols). Here a typist types *kurumade*, which means "by car." The typing is in *romaji*, that is, it employs a standard English keyboard. Thus the typist begins (*a*) by typing an English *k* and *u*. The computer translates that immediately into a single *kana* character. Meanwhile the computer consults an internal dictionary to find entries consisting entirely of *ku* (*boldface type in square brackets*) and entries beginning with *ku* (*light type with a left bracket*). From the entries it constructs a "hypothesis tree." The processing continues (*b*) as the typist enters the English letters *r* and *u*, completing the second syllable of *kurumade*. Again the computer translates the *romaji* into *kana*. The result is a string of two *kana* characters. Then the computer consults the internal dictionary. The possibilities are more varied. *Kuru* could be a complete word. *Kuru* could be the start of a word. *Ku* could be the root of a word and *ru* could be an inflection, or modification of the root, somewhat like the English *-ed*, *-ing* and so on. Finally, *ku* could be a root and *ru* could be the start of an inflection. Some of the possibilities are ruled out because the dictionary has no such entries; others remain conceivable. Accordingly the hypothesis tree grows further branches (*color*). The third step in the processing (*c*) occurs when the typist enters the English letters *m* and *a*. The computer now has a string of three *kana*, from which it constructs the corresponding additions to its hypothesis tree. Some of the additions are conceivable but are not preferred. In particular, the root *ku* plus the inflection *ru* could in principle act as an adjective meaning "coming" or "spinning" (in the sense of making thread);

Consider, then, the typing of *toukyou*. When 東京 is typed a *kanji* at a time, the typist must somehow pick the desired *tou* from among 64 alternatives and the desired *kyou* from among 53 alternatives. In contrast, phonetic conversion requires merely that the typist enter *t o u k y o u*. The computer will respond with the unique, correct *kanji* pair. The creation of the computer's dictionary bestows on the computer a store of information about *kanji*. This greatly reduces the amount of information the typist must transmit to the computer.

I have treated phonetic-conversion typing in reference to Japanese, the language in which it was first applied, but it is equally effective in Korean and Chinese. For Korean a *hangul* keyboard is likely to be popular; for Chinese the romanized spelling called *pinyin* can serve

c

kuruma ⟶ くるま

WORD ROOT DICTIONARY

[Ku-ru-ma]

KANJI	PART OF SPEECH	RELATIVE FREQUENCY	*ENGLISH MEANING*
車	NOUN	8	*CAR*
包ま	VERB STEM	2	*BE WRAPPED*

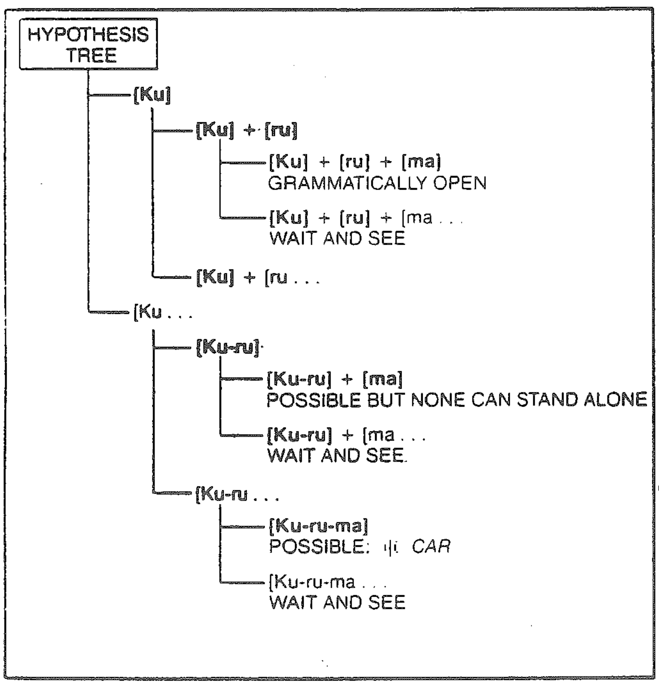
[Ku-ru-ma . . . MANY ENTRIES IN DICTIONARY

INFLECTION DICTIONARY

[ma]

HIRAGANA	PART OF SPEECH	RELATIVE FREQUENCY
ま	VERB ENDING	9
ま	POLITE VERB ENDING	9

[ma . . . MANY ENTRIES IN DICTIONARY



d

kurumade ⟶ くるまで

WORD ROOT DICTIONARY

[Ku-ru-ma-de] NO ENTRIES IN DICTIONARY

[Ku-ru-ma-de . . . NO ENTRIES IN DICTIONARY

INFLECTION DICTIONARY

[de]

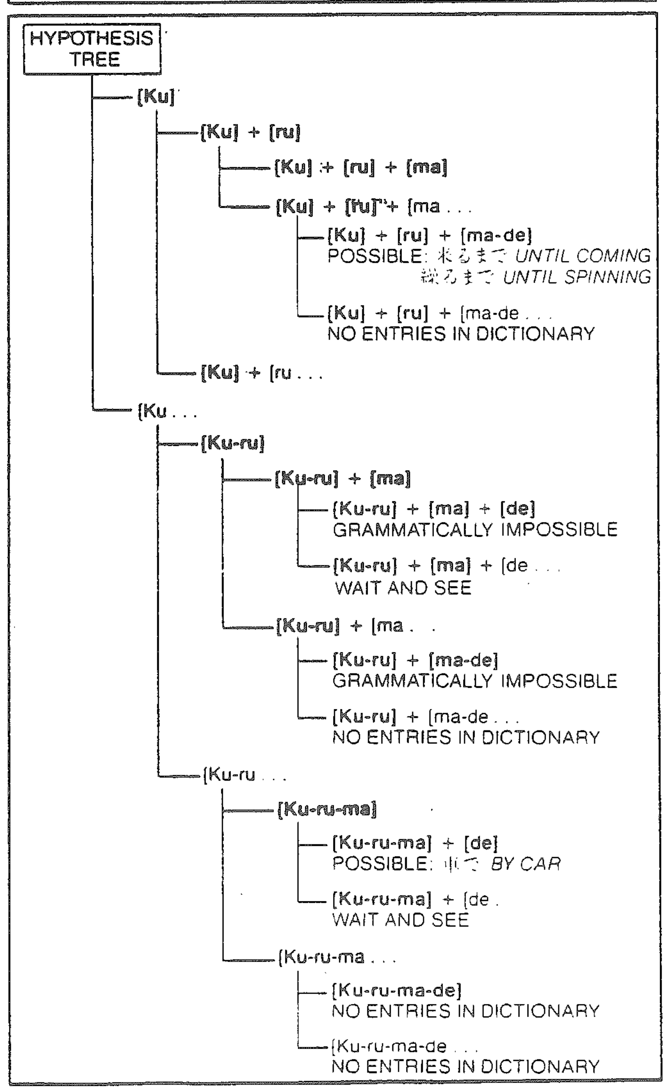
HIRAGANA	PART OF SPEECH	RELATIVE FREQUENCY
で	VERB ENDING	9
で	"BE" VERB ENDING	9
で	NOUN PARTICLE	9

[de . . . MANY ENTRIES IN DICTIONARY

[ma-de]

HIRAGANA	PART OF SPEECH	RELATIVE FREQUENCY
まで	VERB AUXILIARY	9
まで	NOUN PARTICLE	9
まで	NUMBER AUXILIARY	6

[ma-de . . . NO ENTRIES IN DICTIONARY



hence it could modify any noun. The computer will revert to this interpretation only if nothing more concrete emerges from further input. The tree is now at its most complex; the typing of further syllables will introduce additional hypotheses, but most of them will be eliminated by linguistic constraints, such as grammatical impossibility. The fourth step in the processing (*d*) occurs when the typist enters the English letters *d* and *e*, which complete the string *kurumade* and supply the computer with a sequence of four *kana*. The computer consults the internal dictionary one last time. The information retrieved from the dictionary constrains the possibilities. The root *ku* followed by the inflections *ru* and *made* is conceivable: it could mean "until coming" or "until spinning." Moreover, the root *kuruma* followed by the inflection *de* is conceivable: it means "by car." The last step in the processing of *kurumade* is shown in the illustration on the next page.

in the same role as *romaji*. For example, the name of the city Beijing can be typed by simply entering *b e i j i n g*. To be sure, each syllable in Chinese has a particular tonal pattern, and this input does not specify it. Still, that leaves only two dictionary alternatives: 背景, or "background," and 北京, or "Northern capital." It turns out to be far more efficient for a typist to choose among alternatives of this kind than it would be to enter an explicit encoding of the tone for each syllable. Chinese is further complicated by dialects that differ widely in pronunciation, but the standard Mandarin pronunciation is taught worldwide and already is native to 750 million potential typists.

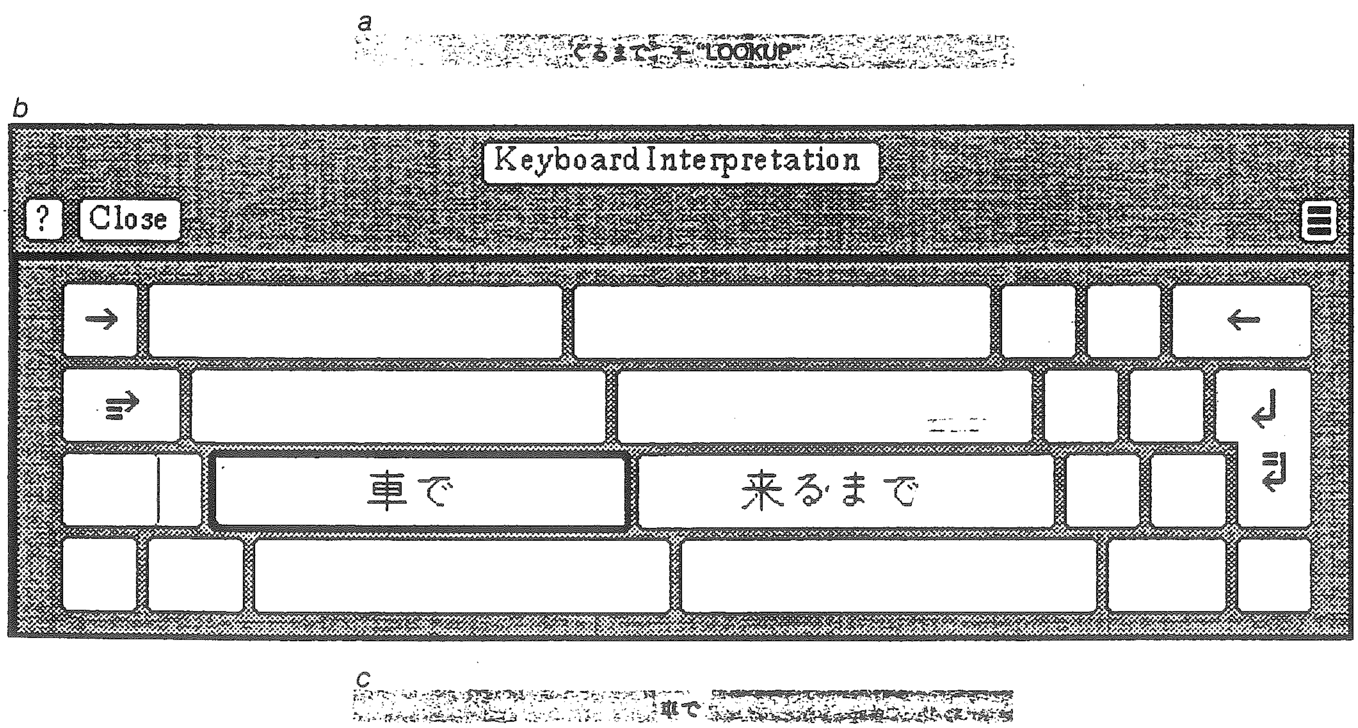
I have described how multilingual text can be typed into a computer and stored in its memory. I turn now to the third aspect of multilingual word processing: the rendering of text that is stored in the computer. For word processing in the English language there is a simple one-to-one correspondence between code numbers in computer storage and rendered characters on a display screen or a printed page. In fully multilingual software, however, that correspondence must be abandoned and replaced by a much richer scheme. When the computer processes a sequence of text codes for rendering, it must be empowered to examine any number of consecutive codes at a time, and it must be allowed to make any computation in order to choose the graphical forms and positions for the characters. In particular, the computer must be provided with variant character forms that it can use to represent the changeable letter shapes characteristic of many writing systems.

An example of the application of these ideas is provided by the handling of the Greek letter *sigma* (σ), which takes on a special shape (ς) when it appears at the end of a word. In the computer every *sigma* is stored as the code for an ordinary *sigma*, even if it comes at the end of a word. Whenever the computer is called on to render a *sigma*, however, it examines the character that follows the *sigma*. If the *sigma* is found to be at the end of a word, the computer renders the variant form instead of the ordinary form of the letter.

Arabic offers a more complex example: most Arabic letters have four forms, depending on whether the letter stands alone or is at the beginning, in the middle or at the end of a word. In addition the rules for joining Arabic letters to their neighbors must accommodate the presence of small markings placed above or below the letters. Nevertheless, the solution is much the same: the computer can be instructed to consider each letter's context before choosing its rendered shape. All the contingencies can be taken into account, given that the computer can make any computation in determining the final appearance of the text.

An important element in rendering is the ligature, in which two or more letters fuse to form a single rendered character. In most English typefaces (including the one in which this article has been typeset) there are ligatures for a few common letter combinations, chiefly *ff*, *ffi*, *fi*, *fl* and *ffl*. Software that takes each letter's context into account can instruct the computer to render the word *office* in four characters—*o*, *ffi*, *c* and *e*—while retaining in its memory the six-letter sequence *o f f i c e* required to process the word correctly.

A similar approach can be adopted for accents, or diacritical marks, such as the German umlaut (*ü*) or the French accent grave (*è*). The International Organization for Standards requires such marks to be represented in the computer as separate codes preceding the code for the letter to which the mark is applied. Hence a letter such as *ü* is represented by two bytes. The computer can render the two bytes ¨ and *u* as the single character *ü* by the same process that renders the two bytes *f* and *i* as the ligature *fi*. Some scripts require the application of multiple marks above or below a base character; the computer can handle this problem by a somewhat more complex rendering procedure.



FINAL STEP in the word processing of *kurumade* comes after the *romaji* typing of the word, when the typist presses a special "lookup key" (*a*). Inside the computer the dead branches of the hypothesis tree are ruled out. The wait-and-see branches, now superseded, are ruled out. The grammatically unpreferred branches, also superseded, are ruled out. The surviving hypotheses are ordered by the frequency of their appearance in Japanese; then the more likely ones (here the ones that mean "by car" and "until coming") are displayed on a virtual keyboard (*b*). Roots are displayed in *kanji*; inflections remain in *kana*. Extremely unlikely possibilities (here the one that means "until spinning") are not displayed unless the lookup key is pressed again. When the typist chooses an alternative, it replaces the string of four *kana* (*c*).

Even the unique syllabic clumps that characterize the Korean *hangul* script can be rendered if the computer is given appropriate software and rendering variants of the Korean letters, which it can build into square groupings. Similarly, the slanting descent to a baseline characteristic of many Arabic fonts can be rendered, freeing Arabic typography from the stricture of a flat horizontal baseline. In the Hindi language the word *hindī* itself has the first vowel (*i*) written out of phonetic order: it is placed before the initial consonant (*h*). When the rule for placing the vowel *i* is incorporated into the computer's instructions, however, the typist can enter the word in its normal phonetic order and the computer will automatically place the *i* before the *h* when it renders the word. The same approach can even rationalize the handling of scripts such as Thai, in which vowels can actually split into fragments that surround a neighboring letter.

Equipped with enough flexibility in the rendering process, the computer can handle any instance in which the letters of a script have a contextual effect on one another. The only remaining rendering problem is a broader one: How is the computer to mix scripts that run in different directions?

A computer cannot do the impossible: there is simply no sensible way to mix vertical text and horizontal text in a single paragraph. Chinese, Japanese and Korean are vertical by tradition. Asian printers, however, have developed the practice of printing these languages horizontally from left to right. Mongolian too is vertical by tradition. Mongolian printers rotate text by 90 degrees when it is necessary to combine it with horizontal text.

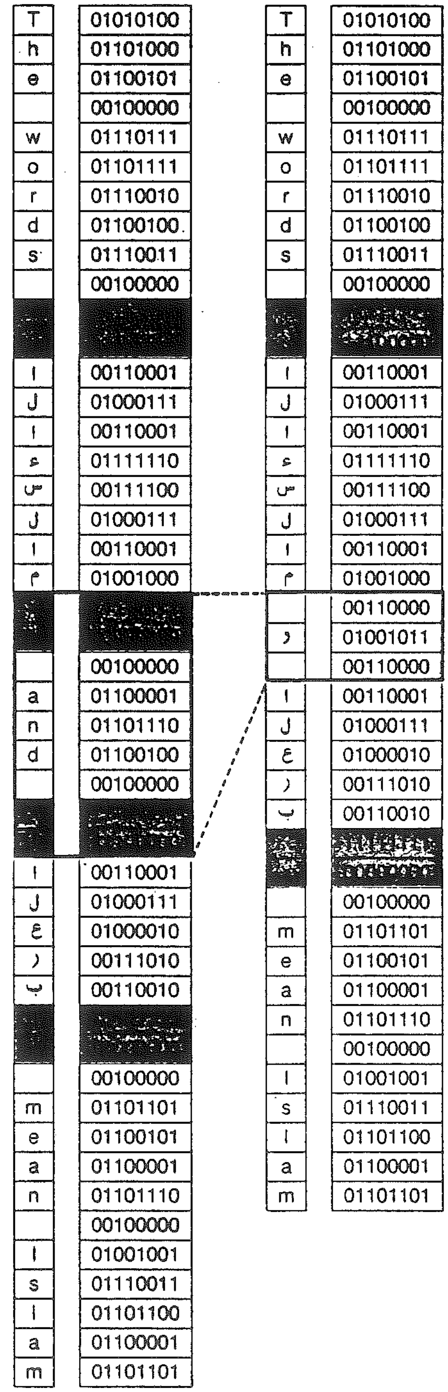
In sum, typographical usage calls for all languages to be typeset horizontally at need. The one circumstance that remains for the computer to handle is the mixing of a horizontal left-to-right script such as English with a horizontal right-to-left script such as Arabic. The typist wants to type Arabic in its proper phonetic order and have it appear in its proper right-to-left sequence, even if it is typed into the midst of English text. Conversely, the typist wants an English passage to appear in left-to-right sequence even in the midst of Arabic text.

The problem may seem perplexing, but it looks simpler from the standpoint of the computer. The computer works much faster than the typist; thus the computer spends most of its time waiting for the typist to enter the next character. Between keystrokes the text is simply displayed without change on the screen; the computer has done no more than produce a static text layout. The static arranging of mixed-direction text is handled routinely by printers of books and magazines in which Arabic or Hebrew script appears.

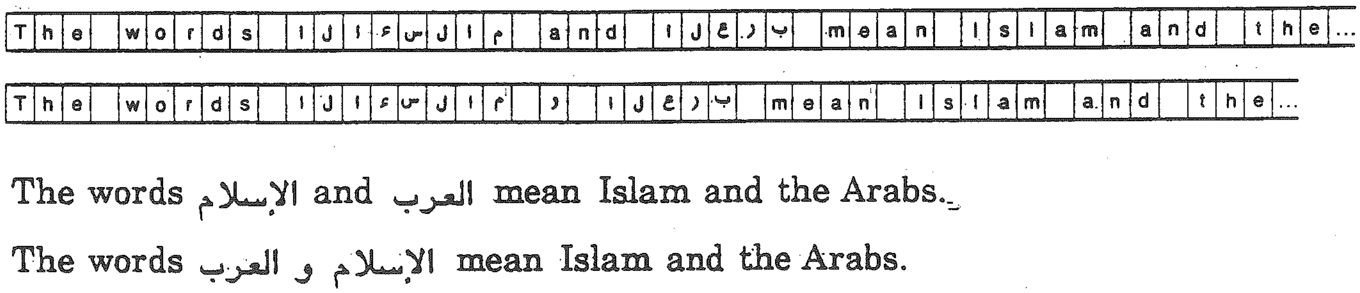
Mixed-direction text can go through remarkable transformations as it is entered into a computer. In most word processors the place on the screen where the next typed letter will appear is indicated by a blinking marker. The marker moves along as the text continues to grow. When a line fills up with type and the marker reaches a margin, the marker automatically drops to the beginning of the next line. For the typing of all-English text the marker moves from left to right. For the typing of all-Arabic text it moves from right to left. For mixed-direction text its activity is novel. Throughout the typing of English text embedded in Arabic the marker must stay put and the newly entered text must slide away from it. The stationary marker cannot, of course, reach a margin. All the same, the filling of a line means that the marker and subsequent words must drop to the next line. The typist has the strange impression that text is falling from the middle of a line. Yet the drop creates the correct layout of the mixed-direction text.

The editing of a mixed-direction text can cause changes that are even more remarkable. Consider the English-Arabic sentence "The words *al-islam* and *al-'arab* mean Islam and the Arabs," where *al-islam* and *al-'arab* are written in Arabic [*see illustration at right*]. Suppose in the course of editing the sentence the English word *and* is replaced by the Arabic equivalent; the Arabic words must then switch places, because they are now part of a phrase expressed in Arabic that should read from right to left. Yet the Arabic words themselves were not involved in the editing operation. Indeed, the text in computer storage changes only to accommodate the replacement of the word *and*. The visible permutation of the Arabic words is entirely a consequence of the rendering process.

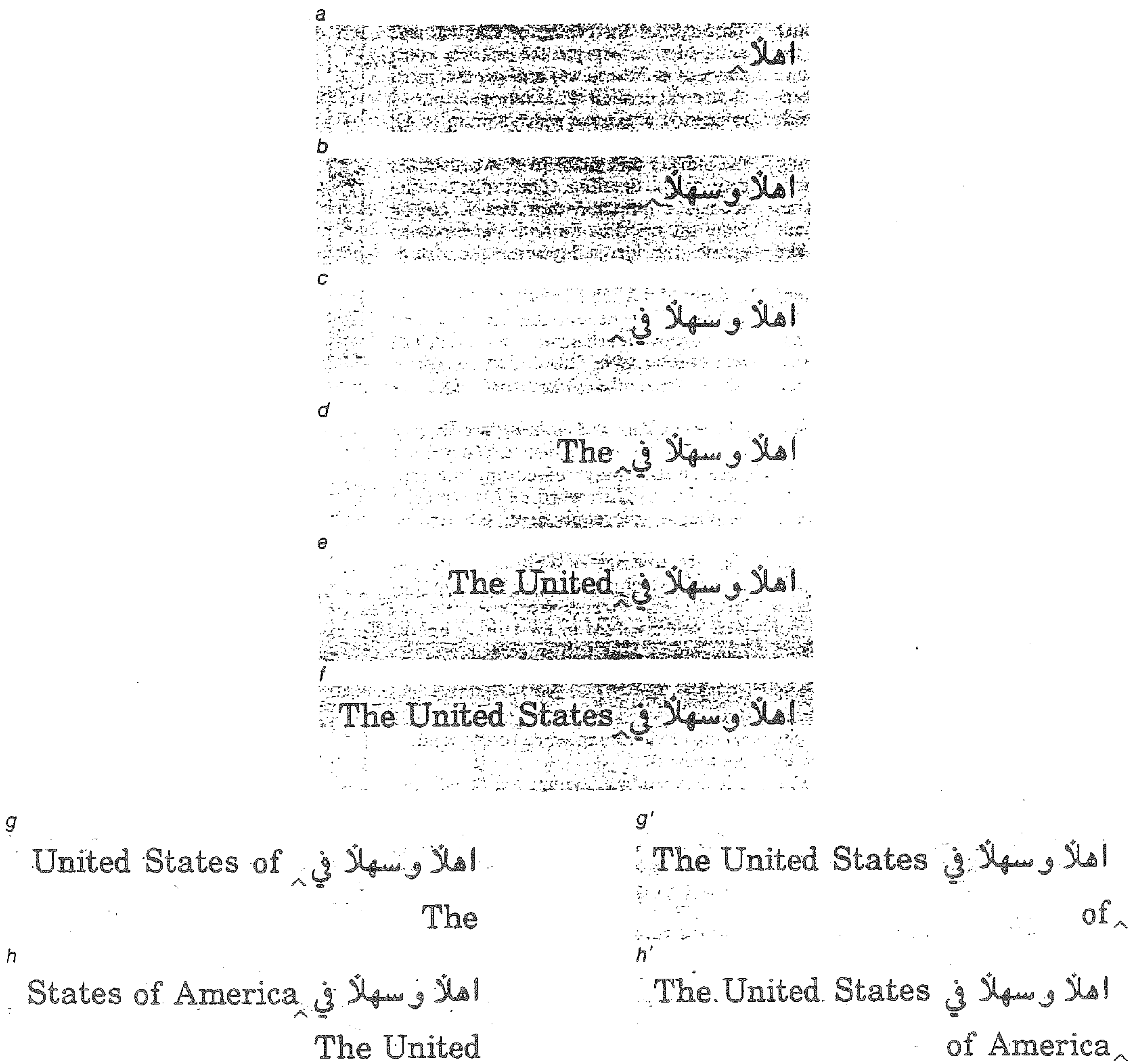
The encoding, typing and rendering of text are the basic elements of multilingual word processing, but they do not exhaust the challenges that must be faced in designing a multilingual word-processing system. Suppose an oil company's Texas offices are made part of a worldwide network of word-processing computer work stations. An Arabic document arrives electronically from the Middle East, but the computers in Texas do not have the software needed to render Arabic text. The Texas computers must nonetheless handle the text as best they can without "crashing" and without mistaking the Arabic codes for English ones. They must at least render the English portions of a multilingual document. Furthermore, the computers in Texas should be capable of accepting an



BILINGUAL TEXT in storage presents no special problems for multilingual word processing. Here the text includes the sentence "The words *al-islam* and *al-'arab* mean Islam and the Arabs," where Arabic words are expressed in Arabic. The sentence is stored in phonetic order (*left*), even though Arabic characters are rendered from right to left whereas English ones are rendered from left to right. Some bytes (*color*) are special codes that shift alphabet and specify a new alphabet. When the sentence is edited (*right*), so that the English word *and* is replaced by its equivalent in Arabic, the only change to the stored text is the removal of the bytes specifying *a*, *n* and *d*, along with some special shift codes, and the insertion of the code for the Arabic letter *wa*.



RENDERING OF THE STORED TEXT shown in the illustration on the preceding page yields a remarkable consequence. The change made in the text involves only the English word *and*, as shown at the top of this illustration. Yet in the rendering of the text the words *al-islam* and *al-'arab* must exchange places, because the entire phrase is now rendered in Arabic script, which must be read from right to left.



"WORD WRAP" is the ability of a computer to move a word to the next line of the display screen if the word will not fit on the line being typed. If the text combines a language that reads from left to right with one that reads from right to left, a special problem arises. Here *ahlan wa sahlan fi* is typed in Arabic (*a*). It means "Welcome to. . . ." The Arabic characters appear on the screen in right-to-left order; a marker (a caret) has moved to the left to show where the next new character will be entered (*b, c*). The typist then switches to English to type "The United States" (*d, e, f*). Now the caret is stationary; the newly typed characters slide to the left to make room for input. That input ("of America") will require a wrap, but if the computer continues to push text to the left, the words "The United" will be forced down a line (*g, h*), reversing the halves of the English. A more appropriate wrap (*g', h'*) drops "of America" and the caret onto the next line.

Arabic-software module, which would add Arabic rendering instructions to the repertory they already have.

The oil company's situation suggests the broad-scale design goals for a multilingual word-processing system. They are compatibility (all the computers in the system must be able to exchange documents in any combination of scripts); open-endedness (each computer must be able to deal sensibly with scripts unknown to it), and modularity (it must be possible to add the capacity for new languages one by one). These goals are difficult, but they can be achieved. The solutions derive in large part from the treatment of text in ways that are broad enough to include any mixture of the world's living languages.

The initial applications of multilingual word processing surely lie in the creation of multilingual documents. After all, as worldwide commerce and politics grow, all kinds of multilingual documents become essential. In addition software that can manipulate multilingual text is certain to bring benefits to language teachers and translators. The automatic translation of documents from one language into another is a far more distant goal. Indeed, at the moment there is little cause to imagine that high-quality machine translation can be achieved. The faithful translation of a passage requires that the translator understand the passage both in its explicit content and in its implications. In a quarter century of intensive research there has been no significant progress in supplying a computer with such an ability. Low-quality translation of texts with circumscribed meaning (such as instruction manuals) is already a reality. Even there, however, the success of machine translation depends heavily on editing by a human proofreader.

The questionable prospects for high-quality machine translation are balanced, I think, by the brilliant outlook for electronic mail: the near-instantaneous transmission of messages or documents by way of private electronic networks or public communication lines. International electronic-mail systems are already in everyday use. They are similar to telephone and postal systems in that their usefulness increases with the extent of the network. It seems inevitable, therefore, that electronic mail services will eventually expand and join to form a single worldwide electronic-mail utility. If this is to come about, one prerequisite is inescapable. The telephone does not require its users to speak only English, nor does a postal system require its users to write only English. Electronic mail will not succeed as a global communication medium unless the text it carries is fully multilingual. To my mind that is the ultimate application for multilingual word processing.

THREE MAJOR APPROACHES TO DEVELOPING COMPUTER-ASSISTED LANGUAGE LEARNING MATERIALS FOR MICROCOMPUTERS

David H. Wyatt

ABSTRACT

Whenever a new project to produce computer-assisted learning materials is begun, one of the most fundamental decisions concerns the selection of an approach to program development. There is a choice between at least three major approaches: using a general purpose programming language, an educational programming language, or an educational authoring system. There are considerable differences in factors such as programming expertise and development time required by the three approaches. Perhaps the most dramatic example of this is the fact that the third approach demands no programming expertise whatever. It is therefore important to be aware of the alternatives in order to make an informed choice of the most suitable approach for each materials development project.

Questions concerning the 'best' approach to developing courseware have often been the subject of intense debate (Holmes 1983; Wyatt 1983a). Some have argued that the current strong interest in producing one's own materials is just a short-term phenomenon and that in the future CALL materials will largely be purchased in ready-to-use form. In this view, considerations regarding the optimum method of developing courseware will soon be relevant only to a small number of professionals. Others maintain that the role of language teachers (and other subject specialists) in software development should be limited to specifying the instructional content and its style of presentation on the screen; all questions of how to realize the material in programming terms are to be left to programmers (Bork, 1981).

Options

In spite of these views, the fact remains that an increasing number of language teachers and curriculum developers are becoming interested in producing their own courseware. This article is addressed to them in an attempt to provide a guide to the various options which should be considered. Three major vehicles for creating courseware will be examined and compared in some detail: **general-purpose programming languages, educational programming languages**, and **educational authoring systems.** The first two of these require the user to acquire programming skills, while the third type demands only the simplest level of computer literacy. Perhaps the most important point to bear in mind before we begin to focus on the different approaches is that *all* of them can be successfully used in different situations. They all have weaknesses as well as strengths, and the choice between the approaches depends very much on your individual circumstances and requirements.

To facilitate the comparison of development methods, we will consider CALL software as essentially being composed of three elements: the language content, the driver, and the management system. This is an oversimplification, and is not equally appropriate for all types of educational software, but it will serve to underline some fundamental points. The *content* may be separate from the program, residing in data files, or may be incorporated directly into the program

David H. Wyatt is currently director of an educational consulting firm in computer-assisted language learning. Educated at Cambridge University and the American University, he is a consultant to the Center for Applied Linguistics and the National Clearinghouse on Bilingual Education. His former positions include Director of CAI at a large system of ESL schools.

itself. The *driver* is the computer program which takes the content and manipulates it so as to present examples and questions, accept answers, and provide appropriate responses. In general, a different type of CALL exercise or activity requires a different (and usually a separate) driver program. The *management* system is the element which determines how students can enter and use the courseware, what route they follow through the materials, and what score and progress information is recorded. Management systems usually involve some separate programs, but management elements are generally also built into each driver program.

General-Purpose Programming Languages

The first main approach to the development of courseware involves the use of general-purpose programming languages such as BASIC and Pascal. In practice, BASIC has been the overwhelming choice of producers of commercial educational software for microcomputers. It has been estimated that 80 per cent of the software to date has been written in BASIC and the remainder in machine language with no significant contribution from any of the other programming languages.

There are excellent reasons for the popularity of BASIC in the creation of software for microcomputers. In theoretical terms, programmers working in general-purpose languages are in much more direct contact with the computer's microprocessor and memory than with the other two approaches. It is possible to exert much more control, and more flexible control, over each step of the operation of the computer. Programmers may design and create every step of the main drivers for their exercises. Similarly, they have complete control over the management system elements in their courseware. For example, the other two approaches frequently offer either rudimentary management systems or demand the presence of a second disk drive if score recording and other management capabilities are to be used. For these reasons, one project directed by the author (Wyatt 1983b) made use of BASIC to create a powerful management component which required only one disk drive for full utilization.

Practicality also dictates strong arguments in favor of general-purpose languages. One important factor is the lack of dependence of the educational programmer on other factors. In one well-known recent case, for instance, a new version of an educational programming language was released with the promise of a management component soon to follow. However, the management component was delayed and became available much later than expected, which must have caused difficulties for those to whom the management capabilities were important. A second practical point concerns the position when new microcomputers are released. In virtually every case, a version of BASIC is immediately available for new microcomputers, whereas a period of months or even years may pass before effective educational programming languages or authoring systems are produced for the computer.

Since general-purpose languages were intended for a wide range of different applications, these languages have few if any commands oriented towards the needs of the educator.

However, the reader has probably already perceived the roots of a problem during this discussion of general-purpose languages—control over all the individual steps of an exercise program, desirable as it may be, implies the need for time-consuming specification of all of those steps. In general, the time required to create a given student activity or exercise in a general-purpose language is significatly greater than with the other two approaches. To be more specific, it is the programming of the new driver (and, to a lesser extent, inclusion of the management system elements) which demands the greater investment of time. A second disadvantage is the considerable learning time required for the novice to become sufficiently expert in a general-purpose language in order to produce moderately sophisticated educational programs.

One of the main reasons for the time-consuming nature of producing educational programs in general-purpose languages is their lack of educational 'power.' Since general-purpose languages were intended for a wide range of different applications, these languages have few if any commands oriented towards the needs of the educator. After only a little experience with general-purpose languages, even novices in educational programming begin to recognize the lack of convenient commands. At a lower level, desirable features of a programming language would include simple commands to cause the program to wait a specified number of seconds and also accept student input in a fully controlled manner (filtering out any unwanted or troublesome keypresses). At a higher level, we would look for some ability to perform natural language answer processing—to search students' answers for key morphemes, words, or phrases (Pusack 1983). Ideally, we would prefer intelligent answer processing, in which students' wrong answers are compared with the expected response, 'marked up' to indicate problem letters or words, and then turned back over to students in an 'edit' mode so that only the mistakes need be retyped at the second attempt. Even the lower-level educational commands described above are entirely absent from BASIC, although they can be duplicated by the programmer with varying degrees of difficulty and effort. These added capabilities are generally developed by the program in the form of subroutines which can be reused whenever they are needed in this or future driver programs. General-purpose language programmers tend to build up large libraries of such subroutines to facilitate their developmental work. Producing the higher-level answer processing features is a major undertaking, however, and BASIC is in any case not suitable for the final form of such routines because of technical reasons (for one thing, it runs too slowly).

Serious as the deficiencies of a general-purpose language may appear, there are effective remedies for many of them. In some cases, it is possible to buy a ready-made set of educational subroutines which will provide the low-level capabilities described above. As one example, the Minnesota Educational Computing Corporation has produced a well-documented collection of subroutines in BASIC for the Apple II (MECC 1980). These ready-made routines can immediately be used in creating courseware and provide the beginning programmer with a head start in general-purpose language work. In at least one instance, an enhancement to BASIC is available which will provide an excellent high-level answer processing capability (Tenczar *et al* 1983). When the problems are solved in this way, the advantages of intimate contact with the computer and complete control over program operation become very compelling.

Educational Programming Languages

The second major type of approach to developing courseware involves the use of programming languages specifically designed to meet the needs of educators. The best known of these educational programming languages in the context of microcomputers is PILOT (Burke 1983), although a recent development known as EnBASIC is also of considerable interest (Tenczar 1983). Because of their specific orientation, these languages incorporate a range of very convenient commands which provide both trivial and powerful educational facilities for the programmer

We will illustrate the range of new commands by reference to the low- and higher-level facilities discussed in the previous section, using the SuperPILOT version of PILOT for the Apple II as our example. At the lower level, PILOT provides convenient commands to cause the exercise to wait a specified number of seconds before proceeding and to 'bombproof' the keyboard so that only desirable, meaningful keypresses have any effect on the screen. In both these cases, simple one-line PILOT commands can permit the program to ignore unwanted keypresses. At a higher level easy-to-use facilities also provide powerful answer processing. One example is the PILOT 'key search' capability. Using this, students may be permitted to enter their answer in a relatively free manner. Once entered, their input can be searched for the significant part of the answer, which might be an affix, single word, phrase, etc. Less important parts of their input can be disregarded, so that spelling mistakes can be tolerated where appropriate instead of causing an otherwise acceptable answer to be judged incorrect. However, PILOT stops short of providing highly intelligent answer processing. EnBASIC is the only current programming language for microcomputers in which this is available. Using EnBASIC, students' answers can be compared against the predicted responses, and then automatically 'marked up' with a simplified set of proofreader's symbols to indicate problem letters and words. Students are then put into a special 'second chance' mode in which they edit their original answer, changing only the incorrect portions. (EnBASIC also provides most of the facilities of PILOT mentioned above.)

In reality, time-saving is probably no longer the most important advantage of educational programming languages.

SuperPILOT includes a number of other features in a convenient package to take advantage of specific capabilities of the Apple II microcomputer. These include utilities to permit creation of special characters and diacritics (such as those needed in foreign language courseware), simple music and sound effects, color graphics, and control over external videotape and videodisc players. At least one other version of PILOT for the Apple, PILOT Plus, provides very similar capabilities including provision for touch-sensitive screen input, and other versions of PILOT now exist for an increasing number of microcomputers. It should be stressed again that programmers in a general-purpose language can supply themselves with very similar capabilities. Features such as special character fonts, sound and musical effects, color graphics, and control of video devices can either be developed by the programmer as reusable subroutines or purchased commercially as 'utilities' in ready-to-use form. This piecemeal solution lacks the convenience of the package provided by an educational programming language, but this deficiency must be balanced against the greater control and flexibility involved in being able to select particular utilities and subroutines rather than being limited to the single packaged facility.

The powerful, convenient features described above make it potentially quicker and easier to program a new exercise driver in an educational programming language. A further saving of time is posible if a management system is also available as part of the package, as is the case with the PILOT Log component of SuperPILOT. The time-saving aspect of educational programming languages should not be overstated. Programming time will be less than in general-purpose languages, but will nevertheless be of the same order of magnitude. It is generally agreed that the time taken for a beginner to become proficient will also be significantly shorter in an educational programming language, but is is again important not to overestimate the gain. Claims of a few hours' learning time for educational programming languages refer only to the use of the simplest commands. In order to use the capabilities of the language at a moderately sophisticated level, far more time is required. In its original form, PILOT was a greatly simplified language, but the extended versions produced for microcomputers have added more complex commands and capabilities. Some have argued that these extended PILOT languages actually require the same learning time as general-purpose languages (Hardy and Elfner 1982). In reality, time-saving is probably no longer the most important advantage of educational programming languages. Perhaps their chief benefit lies in high-level capabilities such as natural language answer processing which enable programmers to develop courseware which is significantly more open-ended and sophisticated than would otherwise be possible.

In this generally positive situation, however, there is the potential for some serious problems. Educational programming languages represent an attempt by their designers to predict what types of commands and capabilities educators will need. Since computer memory capacities are limited, programming languages tend to represent compromises—not all the desirable commands and features can be incorporated, and inclusion of a range of new educational commands may imply the loss of some of the facilities available in general-purpose languages. Indeed, with the original version of PILOT, the range of commands was deliberately kept to a minimum to permit rapid learning of its capabilities (Merrill 1982). In any case, the tendency to reduce the scope of commands usually available in general-purpose languages may lead to problems if the designer of the educational language has not fully anticipated the needs of the courseware developer.

We will illustrate the potential for difficulties with reference again to different versions of PILOT for microcomputers. In one early version, there was relatively little memory space allotted for the textual content of the exercise, so that repeated accessing of the diskette was necessary in order to load and display the necessary screen contents. This caused frequent short delays in the running of many language-oriented exercises. Apparently the designers of this PILOT version had not considered the delays to be significant, and some educators also found them acceptable. A sizable number of teachers, however, felt that the delays were irritating enough to students to seriously jeopardize the effectiveness of their courseware. This problem has apparently been much reduced or entirely eliminated in recent versions of PILOT for microcomputers.

A second specific instance of a problem not anticipated by the educational language designer concerns the character set editor available with PILOT. In general, this is a powerful utility which permits the creation of new symbols and letters needed for foreign language instruction. Unfortunately, however, in PILOT it is necessary to designate a specific key on the keyboard whose symbol will be replaced by the new character. Thus, the additional forms of the vowels in French (acute, grave, etc.) cannot be represented in any very logical fashion; ê, â, and ô must be represented by quite different keys. Using a general-purpose language, however, it can be arranged for the circumflex form to be generated simply by pressing the original vowel followed by a single 'circumflex key'—as it happens, there is an appropriate symbol on the keyboard. Similarly, all acute vowels could be generated by pressing the base letter followed by the slant on the keyboard, whereas PILOT would require them to be represented by different symbols which could not be related to the base vowels. This is a complex but apt illustration of the difficulty of trying to predict, as the designer of an educational programming language, what the requirements of the users are likely to be. Finally, a more general observation which has been made regarding educational programming languages such as PILOT is that they are structured so as to be well suited only to tutorial and drill-and-practice types of CALL, lacking the flexibility for use in developing more innovative and communicative activities.

The authoring system ensures that the screen presentation and question flow are handled very professionally.

Most of these drawbacks are theoretically avoidable, however, and as the relatively young field of microcomputer-assisted learning matures it can be seen that educational programming language designers are improving on their early efforts. Certainly one of the newer languages, EnBASIC, does not seem to suffer from any of these problems. It functions as an enhancement to Applesoft BASIC, so that virtually none of the capabilities of that general-purpose language are lost. At the same time, it supplies some very powerful commands and features for educational applications. The only major sacrifice is a sizable part of random access memory—approximately 16,000 bytes are occupied by the language itself.

Educational Authoring Systems

The third general method of developing courseware is through the use of an educational authoring system. Unfortunately, there is often considerable confusion in terminology between educational programming languages and educational authoring systems. The criterion used for the classification in this paper is whether the courseware developer is required to perform any type of programming while using the method; if so, it is at least partly composed of an educational programming language. If no actual programming is required, then the method is a pure authoring system.

This last point embodies the outstanding advantage of authoring systems: teachers do not need to learn any programming language in order to use them successfully. In fact, all that is absolutely required in terms of computer knowledge is familiarity with the keyboard, disk drives, and diskettes—in other words, an elementary level of computer literacy. This means that the time involved in learning to use an authoring system is essentially required for study of the features of the system itself rather than those of a computer or programming language. A dramatic reduction in learning time is thus possible. The simpler authoring systems can actually be mastered in a period of a few hours or less. More powerful authoring systems have features and techniques which may take considerably longer to master, but they still offer a tremendously shorter learning period than the approaches which employ programming languages.

Authoring systems can also greatly accelerate the speed of development of courseware. Once the language content of the lesson has been specified in the instructional design phase—required in any approach to CALL—the authoring system will dramatically reduce the time taken to translate the plans into the form of software. As an example, let us briefly examine the operation of a moderately sophisticated authoring system, the Assisted Instructional Development System, or AIDS (Wolfe 1982). AIDS is largely a menu-driven authoring system. When first activated by inserting the diskette and turning the computer on, the system offers a menu of actions from which to choose: creating a new exercise, editing existing exercises, transferring files from one student diskette to another, deleting unwanted exercises, etc. If 'create a new exercise' is selected, the user is then prompted for the information required at virtually every step along the way. First, a student diskette must be inserted on which the new exercise is to be recorded. For each question of the exercise, the developer is asked to type in information such as the text to be read (if any), the question to be asked (if separate from the text), the correct answer or alternatives, comments to the student if correct, and to predicted wrong answers, with specific feedback messages on each of the individual problems involved. As each question is completed, the system automatically stores it on the student diskette previously inserted.

When all the questions for the exercise have been typed in, the system then asks if the developer wishes to review and edit the new exercise immediately. Because the process is guided and structured at every point by the authoring system, the creation of new exercises proceeds rapidly and smoothly. When the resulting material is used by students, the authoring system ensures that the screen presentation and question flow are handled very professionally. This is rather gratifying to the novice user, as even the first attempts appear quite polished when used by students. The system also makes it very difficult to create material which will 'crash' because of programming defects, so that the debugging process is greatly shortened and is essentially limited to correcting errors in the language content.

It can be seen that authoring systems supply all the necessary program drivers in ready-made form. Some systems also provide built-in management capabilities. With the AIDS system, for example, there is an extensive score recording and exercise routing capacity. The system also provides a simple keyword answer processing feature.

Although AIDS is an excellent example of a medium-power authoring system, it is not specifically designed for foreign language applications, as is shown by the absence of ready-made foreign alphabet character sets. An example of an authoring system specifically for foreign language application is the DASHER (Pusack 1982) authoring system, which offers foreign characters and diacritics as well as a number of other interesting features. It incorporates an intelligent answer processing capability which is similar in power to that of EnBASIC. Mistakes in answers are indicated by simple 'markup' symbols, and students are then put into an editing mode which enables them to attempt to correct their answers by changing only the incorrect portions. The power and convenience of this edit mode makes this feature one of the best of its kind among current microcomputer-based systems and languages.

DASHER and AIDS are examples of medium-power authoring systems. At a higher level, a number of systems exist which will provide even more sophisticated capabilities. The PASS system from Bell and Howell, for example, will permit the control of videotape and videodisc players from within the courseware. At a lower level, there are a host of inexpensive authoring systems available. Many of these offer a range of relatively rigid exercise formats—typically including true/false, multiple choice, and matching—designed for general educational purposes. Recently, however, there has been a tendency to offer some language-oriented authoring systems which will generate a very limited but indefinitely reusable type of activity. For example, Clozemaster (Jones 1982) will generate a wide variety of different cloze exercises from any paragraphs which are typed into it.

The disadvantages of the authoring system approach are as obvious as its advantages: such systems are relatively quick and easy to use because they embody a built-in educational methodology and program logic. Almost invariably they are strongly *instructional* in nature, and it would be difficult or impossible to use them to create more open-ended or communicative activities of a *collaborative* type (Wyatt 1983c). In general they are not even suitable for instructional programs of the tutorial type, since their branching capabilities are very limited or non-existent. In the great majority of cases, they are suitable only for the creation of drill-and-practice exercises and quizzes. However, if it is precisely this type of courseware that is desired they offer a highly cost-effective option that deserves very serious consideration.

Summary

In summary, there is no single 'best' method of developing courseware. This paper has presented a wide spectrum of possible approaches to the creation of materials for computer-assisted language learning, all of which can be appropriate in different circumstances and for different prposes. It is important that prospective developers of computer-based curricula be aware of the alternatives open to them since different approaches may offer very significant variations in time and energy required for completion of a given project. It is to be expected that the situation will continue to change as new or improved versions of these approaches become available.

REFERENCES

Bork, Alfred. 1981. *Learning with Computers.* Bedford, MA: Digital Press.

Burke, Robert. 1983. *CAI with PILOT.* Englewood Cliffs, NJ: Prentice-Hall.

Hardy, R. Reed, and Eliot S. Elfner. 1982. Writing courseware on the TRS-80. *Creative Computing.* October 82: 34-35.

Holmes, Glyn. 1983. Creating CAL courseware: some possibilities. *System.* 11 (1).

Jones, Christopher. 1982. *Clozemaster* (booklet and diskette) London: WIDA Software.

Merrill, Paul F. 1982. The case against PILOT *Creative Computing* July 82, 70-77.

Minnesota Educational Computing Consortium. 1980. *Programmer's Aid Volume I* (Booklet and diskette) St. Paul, MN: MECC.

Pusack, James P. 1983. Answer-processing and error correction in foreign language CAI. *System* 11 (1).

__________. 1982. *DASHER: A Natural Language Answer Processor.* Iowa City, IA: Conduit.

Tenczar, Paul, Stanley Smith, and Allen Avner. 1983. *A Manual of EnBASIC Usage.* Wentworth, NH: COMPress.

Wolfe, George. 1982. Assisted instruction development system. *BYTE*, August 82, 408-414.

Wyatt, David H. 1983a. Computer-assisted language instruction: present state and future prospects. *System*, 11 (1): 3-11.

__________. 1983b. *Teachers' Management Program* (handbook and diskette). New York, NY: Regents ALA.

__________. (forthcoming 1983c) *Computer-Assisted Learning in ESL.* Washington, DC: Center for Applied Linguistics.

Foreign Language Teaching and the Computer

Edited by Robert S. Hart and Nina Garrett

With this issue of *Foreign Language Annals* we offer a new regular feature presenting reviews of foreign language software and discussions of problems and perspectives in foreign language computer-assisted instruction (CAI). We hope to engage the interest of those who are skeptical or anxious about using the computer, not just to preach to the converted. We are eager to receive materials for review, and to hear from *Foreign Language Annals* readers experienced in foreign language CAI who would be interested in submitting reviews to this column. We also invite suggestions of problems or topics on which future discussions might focus. Send items of interest to:

Robert S. Hart and Nina Garrett
Language Learning Laboratory
University of Illinois
G 70 Foreign Languages Building
707 South Mathews
Urbana, IL 61801

Robert S. Hart begins with an overview of the state of the art in microcomputer courseware. Future columns will be more specific in reviewing kinds of software available.

Microcomputer Courseware for Foreign Languages

CAI has been with us since the 1960's, but it is only with the advent of microcomputers that it has become a matter of immediate concern to teachers. Over 50% of all public schools now have at least one microcomputer, and by 1990 the figure is likely to approach 100%. While teachers and administrators interested in using this new resource face a range of troublesome questions concerning educational policy, curriculum content, and instructional value, the most urgent questions for classroom teachers in every field are: what kinds of courseware are available? which materials best suit my needs?

Researching these questions can be an enormous task for the individual teacher, because the tools available to help with choosing textbooks (catalogues, library collections, reviews in professional publications) are not yet well organized for electronic publishing. In some areas there are district consortia or individual schools designated to provide resources and expertise to help with identifying and evaluating software, but it is difficult to obtain preview copies of lessons directly from distributors because publishers are painfully aware that diskettes can be mass-copied easily and cheaply. Many teachers, therefore, have to depend on the random coverage they can obtain from conferences or word of mouth. This column is designed to provide additional information for teachers to make a judgment.

What is available?

Teachers who expect to purchase one neatly integrated package to fill all their courseware needs are likely to be disappointed. Most micro-based materials are self-contained, providing one or two games or drills on a single diskette, although some are organized into series, for example "vocabulary building" series, or a collection of loosely linked "grammar modules." Although foreign language courseware developers universally assume that computer-based materials will supplement the conventional textbook rather than serve as the sole source of instruction, almost no lessons are designed to accompany a particular text. This is due in part to copyright considerations which prohibit independent courseware developers from shadowing published texts, but the major factor is economic: producing a CAI curriculum requires an amazing investment of time and money—far more than is needed to create an equivalent amount of textbook material—and publishers have judged the potential market attached to a single text to be simply too small to repay that investment. Teachers must thus piece together a computer-based supplement from diverse sources, and the result may be an unsatisfactory patchwork.

The instructional design of existent courseware varies widely in quality and sophistication. The drill is the favorite instructional format. Vocabulary drill is especially popular, not because it represents the best pedagogical use of the computer but because it is technically easy and allows the use of attractive graphic illustrations as cues. Drills are often presented as games by awarding points for a correct response.

Since many teachers find a steady diet of form drills rather monotonous, lesson authors have developed electronic versions of word games such as crossword puzzles and "hangman." More sophisticated attempts to introduce meaningful use of language often utilize some combination of simulations and game paradigms: TERRORISTE (1), for example, simulates a contest between French police and terrorists, and LE DEMANAGEMENT (Krane et al., 6) instructs the student to move furniture around a simulated room. Both of these are reading comprehension activities which require the student to use French words and phrases to respond meaningfully to information presented in French. LES BATISSEURS D'EMPIRE (1), an arcade-type fantasy adventure game, requires the student to use French words and phrases to specify his moves. Such programs do motivate the use of the written foreign language, but the grammar and vocabulary are determined by the game context and may thus be pedagogically quite arbitrary.

Recently some authors have tried to provide more communicatively oriented activities by using what might be described as a modified artificial intelligence approach suitable to microcomputers. One of these, FINDER (Phillips, 7) requires students to solve simple problems by formulating relevant questions in the target language, requiring extensive use of interrogative and adjective constructions. In instructional adaptations of ELIZA (Weizenbaum, 9), a well-known artificial intelligence program which simulates conversation between a patient and a non-directive therapist, the student types in a response in the target language; the program then attempts to analyze the input using the method of key word search and to reply in a quasi-meaningful way. Key word search is not capable, however, of analyzing in detail the grammatical structure of responses, and serious errors may go undetected. FINDER and a wide variety of similar simulation and game paradigms are discussed in Higgins and Johns (5).

Some teachers feel that computer-based materials need to be closely integrated with classroom activities. In response some software manufactureres have developed programs which allow teachers to insert their own materials into a lesson format. DASHER (Pusack, 8), a well-known example of this kind of utility, allows teachers to create simple drills and even performs some error markup on the student's answer. Unfortunately, one is restricted by these programs to an inflexible and very simple exercise format: one can produce "electronic workbook" exercises but the more sophisticated potential of CAI, such as error analysis and intelligent branching, will not be available.

Technical standards have become much higher in recent courseware. Materials programmed for the low-range micros popular in schools several years ago paid little attention to the display of diacritic marks and non-Roman alphabets, sometimes even requiring the student to enter (and see displayed) numerals in place of accents. These problems have been largely solved on the newer machines, and better graphics have been developed for the older ones.

The better graphics packages are fairly sophisticated, allowing one to define complex objects, fill areas with color or texture, and animate the results in complex ways, and this has encouraged authors to dress up their lessons with lavish graphics. Most authors believe that graphics increase student interest, and marketing divisions believe that graphics increase salability. Graphics *can* be a powerful lesson feature, but in many cases they are merely added on and do not improve the content of the lesson or the student's interaction with it.

Almost all current courseware falls short in providing diagnosis and feedback in response to errors. The usual response to a wrong answer is simply "no," perhaps accompanied by a display of the right answer, regardless of the nature of the error. Failure to provide anything more helpful vitiates one of the supposed major advantages of CAI, immediate individualized feedback. Ideally the student should receive something like the markup a teacher would give on a written exercise, with spelling, typing, and accent errors differentiated from grammar errors, and, when appropriate, some explanation about the latter. In some cases the program should also accumulate information on different error types in order to provide a picture of the student's strengths and weaknesses. Why the simpler forms of error analysis have been so neglected is something of a mystery, since even a language like BASIC allows analysis of erroneous adjective or verb inflections.

Future Possibilities

Although this brief discussion suggests many of the difficulties and limitations now confronting foreign language CAI, there is reason to be optimistic. The increasing number of machines in the schools and the consequently expanding user base justify increased support for resource centers where teachers can get help in selecting and using courseware. Many textbook publishers will soon support at least an "electronic workbook" to accompany language texts. Not only will more material be commercially available, but it is likely to appear in more comprehensive packages.

A fundamental shortcoming of early foreign language microcomputer courseware was its restriction to the written modes. Recent develoments in hardware, including videodiscs, random access audio, digitalized speech, and speech synthesis, promise to end this limitation. Such hardware can be put under the control of a micro-based language lesson, which can then select and play prerecorded pieces of video and audio with virtually no delay, and, in the case of audio, can permit traditional listening laboratory activities (including recording of the student's own voice) to be embedded directly in interactive computer lessons. Availability of video allows courseware to introduce oral language in a rich visual context. A few micro-video lessons are already available: MONTEVIDISCO (Gale, 3) is an interactive tour of a Mexican town. As micro-video and micro-audio technologies mature and hardware prices fall to an affordable level, courseware will increasingly incorporate these features. At the same time, the growing power of the micro itself will allow some of the more sophisticated design features of mainframe systems like PLATO, TICCIT, and the Stanford CAI projects to find their way into micro materials.

Looking ahead somewhat further, we can expect to see artificial intelligence techniques applied to foreign language courseware. "Expert systems," programs which mimic the performance of a human expert in some limited domain, are already used in specialized areas such as medical diagnosis and tax law. Prototype work has already established the feasibility of constructing comparable systems to diagnose written grammar errors, which would greatly increase the value of drill activities.

In a somewhat different direction, foreign language courseware could draw on the techniques of natural language processing in order to allow systematic analysis of the meaning of student input. This would allow the student to carry on a meaningful foreign language dialogue with the program, within a limited simulated situation. Some of the communicative activities described in Higgins and Johns (5) represent tentative steps in this direction, but serious work will have to wait for microcomputers capable of accommodating a fairly powerful artificial intelligence system.

This does not mean that language teachers should sit back and await a courseware millenium. On the contrary, they must exert as much influence as possible over the evolution of courseware. We can do this by analyzing our own needs and providing authors and publishers with carefully reasoned evaluations of their products. In addition to informing language teachers about the existence and quality of computer materials, reviews by experienced users can also focus attention on many of the theoretical and practical issues involved in the design and use of foreign language courseware.

Review Criteria

Many of the points to be considered when reviewing foreign language courseware will be familiar to textbook reviewers: intended audience, objectives, prerequisites, pedagogical approach, selection and presentation of actual language content, format, and so on. There are other points, however, which relate specifically to the computer, and teachers' awareness of these features as crucial to quality courseware will contribute to a raising of design standards:

1. *Interactivity.* One obvious feature differentiating textbooks from computers is the fact that computers can initiate interaction. Lessons vary greatly, however, in how they apportion the privilege of initiation. In a drill format, the computer initiates and the student merely responds. In inquiry programs it is the student who asks questions while the computer takes the role of passive respondent. Lessons are generally most successful when they strike a balance, perhaps with some bias in favor of the student's initiative, thus allowing for the back-and-forth flow typical of person-to-person interaction. In any case the nature and origin of interaction in a given lesson should be made clear to teachers, who can then decide whether it fits their needs.

Rate of interaction is important too. Up to a point a higher rate leads to more involvement with the task. Some lessons have a "viscous" feel because of delays in generating elaborate screen displays and accessing information on disc; this retarding of interaction may offset the benefits of the more sophisticated features. And while a high rate of interaction may be satisfying to the user, it is not necessarily related directly to learning efficiency; some forms of high interaction, such as moving the cursor around in a video game context, may make no direct contribution to language learning.

2. *Intelligence.* One kind of lesson intelligence is represented by the specificity of its responses to the student, and this depends on the program's capacity for analysis. Responding to an error with "no" is less intelligent than "This noun is feminine; the adjective should agree. Would you like to see the feminine forms?" Another kind of intelligence depends on the program's ability to store information about the student's performance—to note, for example, that a student has made no ending errors on his last 50 verb productions. It could then skip the next verb-ending drill, taking the student to something more profitable. A less clever program would either force the student to do the drill or leave the decision to student or teacher without providing the performance analysis as a basis for it.

Whether programs should be clever about making their own instructional strategy decisions, make recommendations, or leave all such decisions open, is another important question that cannot be decided across the board: teachers must analyze their student's needs. The computer can efficiently save performance information, but the teacher and the student have access to other information (e.g. motivation level, other priorities) which the machine lacks. At present most micro materials leave such branching decisions up to the teacher. Many teachers are more comfortable with this approach, and it makes for easier programming, but the whole matter deserves systematic exploration, and lesson reviews should be explicit about the program's control of branching.

3. *Human factoring.* This term encompasses all those features which ease the student's interaction with the lesson content. Poor factoring increases the effort the student must devote to the mechanics of operating the lesson, thus decreasing his attention to the language task. More important, it can generate enough frustration and irritation to destroy the lesson's effectiveness.

The visual displays should minimize the time needed to find information on the screen by focusing attention on central rather than peripheral information. Careful design of the mechanism by which the user controls the program is also important: the user should never be uncertain about the results of his actions, and the program must behave as he expects. For example, when the student is given the option of selecting the section of the program he wants to enter next, the set of possible moves and the effect of each should be stated clearly, and he should know whether, and how, he can come back to his point of departure. The overall scheme for moving around should be simple and consistent so that the student can assimilate a mental "map" of the lesson.

Careful critique of human factoring is often omitted from reviews of courseware because it seems like nit-picking over small details, but success in programming results precisely from attention to such details. Reviewers of business software have become conscious that human factoring has a major impact on user productivity and satisfaction, and considerable experimental and theoretical work has provided scientific criteria for it (Card, Moran and Newell, 2). We should demand that foreign language courseware be at least as carefully factored as business software.

4. *Documentation.* To use a micro lesson successfully, both teacher and student must understand how the lesson behaves. The teacher has the further task of understanding enough about the pedagogical approach underlying the lesson design to evaluate its suitability. Inferring how a program operates by experimenting with it is a tedious chore; with complex designs it may be impossible even for an experienced teacher to determine how the program is making decisions. Unfortunately the explicit documentation which would provide this information is not usually supplied with lessons. Courseware authors should provide written operating instructions in addition to and more detailed than those given on-line. The major structural units of the program and the sequencing options for connecting them should be described, as should "hidden" operations which save data, make branching decisions, do error analysis, or generate items. Whenever possible, a complete listing of lesson content (for example, vocabulary used) should be given. By reviewing documentation as rigorously as the courseware itself we can encourage publishers to provide us with a more usable product.

The design and production of computer-based foreign language courseware is still in its infancy. Systematic investigation of interactivity, intelligence, human factoring, documentation, and many other important design features by a well-informed user public can provide the input needed to guide courseware development and smooth its path to maturity.

NOTES

1. Authors not given: TERRORISTE and LES BATISSEURS D'EMPIRE are distributed by The Professor; 959 N.W. 53rd Street; Ft. Lauderdale, FL.
2. Card, Stuart K., Thomas P. Moran, and Allen Newell. *The Psychology of Human-Computer Interaction.* Hillsdale, NJ: Lawrence Erlbaum Associates, 1983.
3. Gale, L. "Montevidisco: An Anecdotal History of an Interactive Videodisc." *CALICO Journal* 1, i (1983): 42-46.
4. Hertz, R. "A Software Evaluation Guide for the Language Arts." *CALICO Journal* 1, v (1984): 21-24.
5. Higgins, V. and T. Johns. *Computers in Language Learning.* Reading, MA: Addison-Wesley, 1984.
6. Krane, A., P. Orenstein, L. Copulsky, and R. Wickenden. "Le Français par Ordinateur: Le Déménagement." Boston: D.C. Heath, 1984.
7. Phillips, M.K. "Approaches to the Design of Language Teaching Software." [Paper presented at the Symposium on Man and Media, Frankfurt, Germany, June 1984.]
8. Pusack, J.P. "Dasher: A Natural Language Answer Processor." *Conduit* 7, i (Spring 1982): 64-66.
9. Weizenbaum, J. "Eliza—A Computer Program for the Study of Natural Language Communication Between Man and Machine." *Communications of the Association for Computing Machinery* 9 (1966): 36-45.

ease in storytell-
pecific about not

n the classroom,
ich preparation.
on that these are
the teacher still
more often than
r. Although it is
s material for be-
els, most of it is
s and only occa-
bout the level(s)
is appropriate.
ok (ten sections
, etc.) leads one
and techniques
into meaningful
rily the case. The
and at times not
ssification (e.g.,
udents' stories).
ed by a more sys-
nique category.
just one activity
on a collection of
room activities.
to groups (e.g.,
writing) with the
e) used for each:
bution); options
e); materials re-
markers, index
her paper); pro-
nd special con-
t the index or at
ent information
ity and the level
e is also a gram-
cusing on a spe-
ons and explana-
ow. Although not
all can be useful
teachers who will
ariations on the
new ones of their

all or part of one
r more than one
es several weeks.
any cultures and
ses and probably
foreign language

classes. For those teachers intent on "covering the textbook" these activities may appear to be time consuming, but their use most likely will be time well spent, as good teacher-directed but truly student-centered activities are more interesting, fun, relevant, confidence-building, and rewarding in the long run than mere dependence on a textbook. The only drawback to some of the activities is the possibility that in a large class there would be insufficient room for students to work well in pairs or groups and to be able to post or otherwise display materials.

Both books offer avenues by which teachers can break away from the routine of set texts and exercises into more open and real exchanges in which students play a major role. Such an approach leads to greater student motivation and cooperation, better retention of what is learned (acquisition), and a more positive attitude toward the language learning experience in general. There is no question that both books are a positive contribution to ESL teaching and to language teaching in small classes. *Action Plans* should be an asset to new and experienced foreign language teachers in general. *Once Upon a Time* seems best for experienced ESL teachers and is perhaps more suitable for teaching in England than in the United States.

ELISE ANDRÉ
Berea College

KENNING, M. J. & M.-M. KENNING. *An Introduction to Computer Assisted Language Teaching.* Oxford: Oxford Univ. Press, 1983. Pp. xii, 195. $10.95, paper.

This practical handbook for the interested language teacher is a welcome complement to the growing number of publications pertaining to computer-aided language learning. It supplements, for example, *Computers in Language Learning* by John Higgins and Tim Johns and offers an extremely clear and practical introduction to programming in BASIC. It shows the reader step by step how to write a dialogue program that performs the standard operations needed to run an interactive grammar or vocabulary drill, a reading comprehension exercise, or any other programmed learning activity. It teaches the novice programmer some elementary tech-

niques such as input statements, strings, matching, and branching. It also shows how to write loops and how to get out of them; how to have the computer scan learner responses for exact matchings or key word matchings; how to write data-independent subroutines for various drills and exercises; how to randomize computer output, such as commiseration and congratulation statements, for more realistic effect and increased user friendliness; how to program HELP files that are initiated by the learner and offer guidance or feedback.

The beginning chapters contain sample programming exercises with which the readers can test their understanding of the material. Although language teachers do not need to learn BASIC in order to use CALT material, the authors recommend getting acquainted with programming techniques for two reasons: 1) it demystifies for the language teacher the power of the computer and shows its potentialities and its limitations as a language learning tool; 2) "writing computerized lessons fosters good habits of clarity and coherence in course design or curriculum development and is a healthy exercise for any practising teacher or student of education" (p. 132). The hints and warnings given in chapter nine for the creation of CALT material are based on sound educational principles and could be used for the development of any teaching material.

The last chapter, which takes a close look at past and current trends in CALT, shows that much of the software produced up to now is of the programmed learning type. And yet, "the association between CALT and Programmed Learning is largely an historical accident" (p. 150). The computer can be programmed to offer the user a far more creative and autonomous learning experience than would be expected from seeing much of the currently available materials.

At the end, the authors give a few examples of recent advances in CALT that "hand over control to the learner." These are so essential to the prevailing liberal approach to foreign language teaching that the reader only regrets not having been introduced to them earlier in the book as alternatives to the more programmed learning routines presented here.

CLAIRE KRAMSCH
Massachusetts Institute of Technology

HOLES, CLIVE.
Saudi Arabia.
Paul. 1984. P

This text is a
duction to the
and Saudi Ar
new point of l
by detailed exp
literation. Use
tern drills, ar
opportunity fo
lists at the en
reference. An
of the book to
work indepen
Colloquial Ar
who needs to
the education
spoken in an
through Kuv
Arabia, Qatar
Oman" (p. 1).
ficiency, the A
duced.
Certainly th
After an exter
system of Ara
a wide variety
items along wi
as the focal po
one dialogue g
cussing person
section deals w
of fasting (p.
The autho
grammar-tran
odology. Less
cal order of d
exercises, pat
drills and pro
The format is
and ESL mate
formation, m
analysis provi
language teach
table is given
translate the w
tice the patte
drawn from h
the drills. U

Computer-Assisted Teaching and Testing of Reading and Listening

David H. Wyatt
Specialized Curriculum Design

The computer has the potential to play an important role in the teaching and testing of the receptive skills. In the area of teaching, this role need not be limited to traditional tutorial and drill-and-practice activities, and a wide range of other possibilities should be considered. The area of testing also holds promise for computer-adaptive innovations in addition to the computerization of familiar types of test.

Introduction

The potential value of the computer in language learning and testing has been the subject of debate and experimentation for over fifteen years. During this period, however, the computer medium itself has been evolving rapidly and has undergone some fundamental changes. Slow, noisy, text-oriented teletype terminals have been largely replaced by fast, flexible video display units with graphics as well as text capabilities (Holmes and Kidd, 15). In the public schools, the new inexpensive microcomputers already outnumber terminals based on larger computer systems by a ratio of five to one (U.S. Department of Education, 37), and it appears that this process is now beginning to be duplicated in higher educaton. Over the same period, there have been equally fundamental changes in our views on what it means to learn a language or to test that learning. In light of such developments, it seems wise to be unusually cautious in making generalizations from individual instances of computer-assisted language learning or testing.

In terms of the theory and practice of language learning, two of the most significant changes of the last fifteen years have been the attention to communicative approaches (Widdowson, 40) and the fostering of language acquisition (Krashen, 20). Historically, computer-assisted language learning has been closely associated with behavioristic and cognitive styles of teaching as exemplified in drill-and-practice and tutorial programs. However, with the far-reaching changes in our beliefs about the nature of language learning, it is important to recognize that the use of computers as *instructors* is by no means the only possibility. Two other types of computer involvement are important: the roles of *collaborator* and *facilitator* (Wyatt, 42).

In instructional computer programs, students work in a highly structured fashion toward carefully planned learning objectives. They essentially follow directions and act as responders rather than initiators, a process that closely parallels the traditional activities found in classrooms and textbooks. In collaborative computer programs, however, the initiative is turned over to the student. Activities of this type may be planned for use with groups of students rather than individuals. The outcomes of collaborative activities may be predetermined but they may also be unpredictable. Even where the final outcome is defined, the path to it depends on the initiatives taken by individual students or groups of students. One example of this would be an adventure reading program. The third category, facilitative software, involves the use of the computer as a tool for learning. In themselves, facilitative programs are essentially empty of any teaching content—they simply make the learning process easier. One of the best known examples of this is the use of word processing programs in the writing class. Further examples of all three types of courseware will be provided in the sections on reading and listening. Similar distinctions between general types of learning experiences in a computerized environment have been made by other workers (Kemmis *et al.*, 19).

Another major consideration in computer-assisted learning is the state of flux in computer hardware. Until 1979, computer-assisted learning systems were almost exclusively based on terminals attached to large mini- or mainframe computers. Even today, the great majority of software for language learning is available only on large systems such as PLATO,[1] TICCIT,[2] and CCC,[3] despite the growing predominance of the microcomputer at all levels of education. The problem is the almost complete lack of transportability of courseware from one computer to a different model—a problem often compounded when the transfer is attempted from a large computer system to a microcomputer. Organizations such as CONDUIT[4] are working hard to alleviate this difficulty. At the moment, however, we are faced with a situation in which much of the available language courseware will not run on the most affordable and widely used computers.

For these and other reasons, this paper will not attempt to provide a systematic catalogue of existing CALL software, although individual programs will be cited in the discussion. Rather, the aim of the paper will be to describe and assess what can be achieved in the learning and testing of the receptive language skills with computer hardware now available. Some of the programs described have already been produced, and some are in progress, but many represent new directions. In this way, it is hoped to provide some guidelines and suggestions for the development of language learning and testing software.

Reading and Vocabulary

In dealing with teaching and testing the receptive skills, a broad division has been made into CALL activities related to standard, familiar classroom practices and some newer, more innovative computer-based possibilities. Often the more familiar computer-assisted activities are instructional in nature, whereas the innovative ones are collaborative.

Overview of Reading and Vocabulary

Reading and vocabulary represent two of the areas of the language curriculum where computer-assisted learning theoretically holds the greatest promise (Wyatt, 43). By its very nature, reading is a highly individual and idiosyncratic process. Even in a class where the proficiency levels of the students are unusually similar, reading speeds and comprehension abilities tend to vary quite widely. With many class activities, it is necessary for the teacher to choose an average speed of presentation. Experienced teachers realize that this compromise is particularly unsatisfactory for the reading class. Whatever the average speed selected, it will be too slow for the abler students, not challenging and developing their skills as it should, while being frustratingly rapid for the less able students. The reading class almost demands a more individualized, student-centered approach.

This need was the motivating force behind the development of 'reading laboratory' materials such as those of Science Research Associates (Parker, 29). However, a much higher level of individualization, together with all the other advantages of CALL, can be achieved through a computerized presentation. In this way, a wide range of traditional and innovative activities can be provided in the area of reading skills.

Before outlining these activities, however, some important limitations of most current computer systems must be emphasized. One of the most irksome problems is the limited amount of text displayable on the video screen at any one time. Without the use of special programming techniques, one of the most popular microcomputers in our schools will display only 24 lines of 40 characters, as compared to a typical typed page of 56 lines with 70 characters per line. Some of the new microcomputers, and specially programmed older models, can display 70 characters on a line. However, the desirability of this is debatable, since the legibility of the text is much reduced. It may well be better to work within the confines of a 40-character screen, which despite its other limitations does have excellent readability. This problem can largely be overcome by providing any reading material of more than a page in length in traditional book or loose-leaf form, reserving the video screen for what it does best—presenting information or activities to the student in a dynamic, responsive manner.

A second limitation which some teachers feel is important is the absence of a spoken presentation of the reading materials, particularly at the beginning level. In many respects, this only appears to be a problem because teachers may be thinking in terms of the computer-based activities as constituting the entire lesson. There is no reason that the lesson could not begin with the teacher reading the material aloud before moving on to the computer-based phase. If individualized access to a spoken version is desired, the least expensive method is to supply an audio cassette and cassette recorder under the student's manual control. Alternatively, the cassette recorder could be controlled directly by the computer through the addition of a simple and inexpensive interface. If more power and sophistication is desired, a random access audio device can be utilized (Wyatt, 41).

A third limitation is found in the computer's inability to handle relatively open-ended or higher-level interaction with students. Even with less ambitious CALL materials, unplanned or unprogrammed responses will arise from time to time while students are working at the computer, bringing the teacher into play as a resource person and advisor. At the more communicative level, the 'problem' of the computer's limitations is resolved when the CALL activities are seen as only one component of a lesson, to be followed by higher-level activities such as summary, discussion, and debate in a traditional manner.

CALL and Standard Reading Activities

The main focus of many modern reading courses in second language learning is the development of a body of reading abilities, skills, and techniques which have been increasingly well defined in recent years (Hosenfeld *et al.*, 16). Courses such as *Reader's Choice* (Baudoin *et al.*, 1) and *Skillful Reading* (Sonka, 35) have been based on syllabuses which include *practical reading strategies* such as skimming and context guessing; *general discourse features,* such as anaphoric reference and paragraph main idea; and *specific language features,* such as the notions of cause and effect or functions like classification and definition. Other aspects of reading which have received attention include *reading improvement* and readiness techniques and *vocabulary expansion.* This section will present some examples of the ways in which computerized materials can contribute to such curricula.

PRACTICAL READING STRATEGIES

One of the skills needed for reading successfully in a second language is the ability to *deal with new or unfamiliar expressions.* In many cases students should be capable of deducing the meaning from contextual or morphological clues. The PLATO system has software designed to encourage this process in German reading, and the program has been effective (Weibel, 38). On the other hand, there will be cases in which the clues are insufficient and the exact word meaning is crucial to understanding, thus requiring the use of a dictionary.

In many reading courses, the skills of using contextual and morphological clues to deduce word meaning are introduced and practiced at an early point. For example, the presence of an antonymous phrase in the same sentence or the *-er* suffix to indicate agentive meaning may all help to elucidate an unfamiliar expression. This practice can be individualized through CALL along the following lines:

> The computer displays a sentence or paragraph containing an expression that the student will probably not recognize. The student is asked to try to deduce the meaning of the expression. First, however, the student must indicate whether *any* of the standard clues to meaning are to be found in the sentence. The computer will thus guide students to adopt the correct *approach* in handling this type of difficulty. Through a multiple-choice format, the student is then asked to deduce the meaning of the unfamiliar expression. The computer will have a large bank of items, classified by the type of clue that students should be able to recognize and use. As the exercise progresses, the computer will automatically begin to focus the practice on the types of clues with which the student has most difficulty.

This entire introductory series of exercises could be presented in game format, with the students gaining points or losing them depending on the accuracy of their work. With the potential for attention to individual problems together with the motivating power of the game element, the computer presentation would represent a considerable advance over a traditional classroom presentation and constitute a valuable instructional type of activity.

One of the main difficulties with these word handling skills is achieving effective transfer from the initial presentations to everyday student use. The computer can provide a powerful solution to this problem, and we will return to this point in a moment. First, let us deal with the related question of dictionary use.

A serious problem for readers in a second language is excessive use of the dictionary. Any dictionary use tends to slow down readers very considerably, distracting them from the main task at hand, and the problem is exacerbated by overfrequent use. At the simplest level, the computer can relieve this problem by acting as a rapid reference tool—a facilitative function which has considerable promise and has been employed to good effect on the PLATO system (Jamieson and Chapelle, 17). However, with very little extra programming, the computer can also provide day-to-day assistance in transferring the strategies of using context and morphology clues into the students' active repertoire.

This type of *interactive dictionary* works as follows:

> The student, reading a passage on the video screen or in a textbook, encounters an unfamiliar word. To obtain dictionary help, the word is typed into the computer. Instead of giving direct assistance, the computer gives a multiple choice of several possible meanings and invites the student to use context and morphology clues, if present, to deduce the word meaning. Each time students resort to the computerized dictionary they are asked to develop their skills in handling new expressions. The whole process can take less time than locating the word in a standard dictionary.

Other reading strategies such as *reading according to purpose* lend themselves to instructional CALL activities. For example, *skimming* a reading passage to get a grasp of general meaning or *scanning* rapidly through when all that is required is one or two specific items of information are both techniques that can be practiced in an individualized manner with CALL. As explained previously, relatively long passages would probably best be provided in booklet form, with only the interactive questions and answers being displayed on the computer screen. Shorter introductory items could be presented entirely on the video display.

One reading strategy which appears to be an important component of fluent reading is *prediction*—the continuous formation of hypotheses about what is coming next, based upon the materials already encountered. Activities to encourage and practice this have been attempted in textbook form, but the computer is clearly a much more suitable medium for this. In one format, students read a certain number of sentences and are then given a choice of several possible examples of a 'next sentence.' If correct, they continue the exercise in similar steps. If their choice was not the best answer, however, the computer can draw attention to the features of the preceding sentences which mitigate against that choice. Hints designed to stimulate their thinking and promote the predictive approach can be presented, followed by a return to a multiple choice of the remaining possibilities. One strength of screen-based exercises is the ability to continue presenting the same multiple-choice items to students. Because their answer is corrected without revealing the right solution, this constitutes a highly economical use of material. In the same way, extensive hints and suggestions can be provided for the slower students without impeding the progress of the abler ones.

GENERAL DISCOURSE FEATURES

Many modern reading courses cover a wide range of discourse features such as anaphoric reference, logical connectives, and paragraph structure and main idea in a very general way. (Some of these features may be covered in a much more specific, detailed manner, as outlined in the next section.) For example, students' attention may be drawn to the range of logical connectives like *therefore, nichtsdestoweniger,* or *pourtant.* Rather than presenting the connectives in a systematic, sequenced manner, reading courses which operate at the general discourse level usually limit themselves to pointing out the importance of their function and giving practice with a limited number of examples.

In working with this type of general discourse feature, traditional textbooks suffer from two significant disadvantages. First, when new features are introduced and practiced, there tends to be far too little practice material for any but the most able students. In terms of the economics of publishing, this is understandable; each practice item involves a stretch of *discourse* rather than a short phrase or sentence, so that a single exercise occupies a much larger space than usual. The second problem with a traditional textbook presentation is the familiar one of encouraging transfer of skills from the context of exercises to the student's everyday repertoire. Frequently the general discourse features receive no further systematic attention once they have been initially introduced and practiced.

The computer can assist in solving both of these problems. In place of the traditional textbook, the computer-aided course could provide a booklet with a variety of reading passages to accompany the programmed exercises. One main advantage of the separation of reading materials from exercises is that the reading materials in the booklet can be used and reused for a variety of activities. Rather than being tied to any particular exercise, the reading passages function as multi-purpose materials. Shorter exercises can be based entirely on the video screen, while those requiring longer discourse passages will direct the student to the appropriate place in the booklet.

A second benefit of the computer-assisted approach lies in the computer's ability to store information about the student's progress in different skills from one lesson to the next. This could provide a significant advantage in helping students integrate the skills into their regular reading. Such instructional CALL materials could make an individualized, automatic selection from a range of exercises at the start of the next lesson, giving students further practice on their areas of weakness. Specific general discourse features can be recycled until students begin to demonstrate successful recognition and understanding, at which point the computer will shift the focus of activities in subsequent lessons to new features.

In this manner, CALL materials could focus on a large number of general discourse features. For instance, the familiar *pronoun reference* type of exercise might be presented entirely on the video screen as follows:

> Research studies have shown that the beneficial effects of pesticides can be offset by *their* environmental danger. Governmental laboratories are currently investigating insecticides such as dieldrin and DDT.
>
> *their* refers to
> a. research studies
> b. beneficial effects
> c. pesticides

With the screen-based version of the exercise, the meaning relationship between *their* and *pesticides* can be shown dynamically through animation or graphics. With booklet-based activities, far more practice items than usually possible in a textbook could be provided for students having trouble with this type of anaphoric reference. Other familiar exercises on general discourse features such as paragraph main idea, paragraph structure, and general communicative function of sentences and paragraphs could also be enhanced through computerization.

SPECIFIC LANGUAGE FEATURES

Some recent reading courses are based on a detailed and systematic syllabus of specific language notions and functions. In the case of an ESL course dealing with notions such as cause and effect, for example, we might present associated vocabulary (*result, prevent*), connectives (*consequently*), and syntax. With functions such as definition, process description, or classification, similar features are addressed, with additional attention to the structure of paragraphs and passages which present the functions.

All of these aspects of notions and functions can be introduced and practiced in instructional CALL programs. For example, the computer may first present and explain the function. Using screen- or booklet-based materials, the associated vocabulary, connectives, and syntax are directly practiced. Finally, the function may be demonstrated at the discourse level with attention to its role in paragraph or passage structure.

READING IMPROVEMENT

Many second language students have a tendency to read very slowly, approaching text as a word-by-word deciphering task in which it is crucial to establish the meaning of each lexeme before moving onto the next. This approach inhibits the development of reading comprehension and has led to a variety of materials designed to improve students' reading techniques and speed.

At the simplest level, the main methodology has been to provide regular practice in reading short, easy pasages under teacher timing and supervision. Fry's *Reading Faster* (8) and the SRA 'speed builder' reading laboratory component (Parker, 29) are typical examples.

At a more sophisticated level, texts such as *Reading Improvement Exercises for Students of ESL* (Harris, 11) also aim at specific techniques for improvement: reading in meaningful word groups, expanding the size of those word groups, and grasping them with a single eye fixation. However, the print medium is dramatically limited in this regard, since on a printed page there is virtually no way to control what the student sees and does. This has led to the development of educational technology such as the tachistoscope, capable of projecting words and phrases rapidly on a small film screen in front of the student. At best this is a clumsy, expensive, and relatively minor advance over print.

For all of these reading fluency approaches, the computer provides an ideal instructional medium. With the simpler approach, the reading selections can be provided in booklet form. The student touches the keyboard when beginning and when finishing the first passage so that the computer will automatically time the reading. A 'clock face' display on the screen allows the student to check on the time at any point while reading. Comprehension questions are presented interactively on the video screen. The answers are scored immediately, enabling the computer to direct the student to a reading passage which will provide just the right amount of challenge for the next exercise. Information is stored at the end of the lesson for future practice sessions.

In terms of the more sophisticated approach to improving specific reading techniques, the video screen provides a much more effective, controllable, and flexible medium than either print or tachistoscope. For example, meaningful word groups can be highlighted in turn in a sentence on the screen, or the meaningful groups can be presented in sequence vertically underneath each other (Geoffrion and Geoffrion, 9) as in the following:

For example,
meaningful word groups
can be highlighted in turn
in a sentence
on the screen
or
the meaningful groups
can be presented
in sequence
vertically underneath each other

Among the many advantages over the print medium are the dynamic potential of the video screen and the ability of the computer to adapt the speed of presentation and length of the meaningful groups to challenge each student's individual abilities.

VOCABULARY AND ALPHABET

Work with vocabulary is generally an inextricable component of reading courses and textbooks. Often morphology receives some systematic attention, and this is one area in which instructional CALL materials can make a contribution (Scanlan and Vollmer, 33). Affixation and its associated spelling changes can be very effectively introduced using the dynamic capabilities of the video screen. The meaning clues provided by affixes can then be targeted for practice as described in the Strategies section of this paper until they are transferred to the student's active repertoire.

Computer-assisted techniques can also provide a simple and effective means of checking at the end of each reading unit that the appropriate new items have entered students' active and recognition vocabularies. Through computerized post-tests, students can be given additional on-the-spot practice with expressions they find difficult. Using the record-keeping power of the computer, problem items can automatically be recycled for attention by individual students in the next reading unit.

Generalized vocabulary expansion materials are also useful in learning idiomatic expressions or with students with relatively poor vocabularies. It appears that CALL techniques for vocabulary improvement result in more rapid learning with higher and longer-term retention. This may be due in part to the enjoyable types of activity which are possible on the screen, including game formats. On the PLATO system, for example, there is a version of the 'Concentration' game in Russian (Hart and Provenzano, 13), while both microcomputers and larger computer systems have 'Hangman' games in a wide variety of languages.[5] One unfortunate feature of some existing vocabulary games should be avoided—the tendency to present items without proper context. An attractive feature of some of the vocabulary game programs now available is the provision for teacher input so that teachers can supplement or entirely replace the original vocabulary contents with their own materials. Before leaving the subject of games, let us note that it is easy to misunderstand their nature. Many educational games are purely instructional; essentially, they are an attractively presented form of drill and practice.

At the beginning level, the computer can provide an excellent medium for learning non-Roman scripts. On the PLATO system there are lessons on the Cyrillic alphabet (Curtin and Dawson, 5) as well as on Chinese characters. These non-Roman scripts can also be displayed on some microcomputers[6] although generally with less capacity for fine detail.

New Possibilities in Reading

Most of the CALL applications outlined so far have been instructional, but there are roles the computer can play which represent more striking departures from current classroom practice.

One of the most popular types of commercial software among personal computer users is the fantasy adventure. In one well-known variant the users play the role of heroic explorer of an underground empire. Using simple commands, the users tell the computer what to do next: "go north," "enter room," "attack thief," and the like. The computer gives detailed descriptions of their current location and surroundings, state of health, accumulated wealth, and other pertinent facts. All of the information that the user needs to read, understand, and act on to survive is provided in discourse form on the screen. The object of the adventure is to explore the empire, amass treasure, and stay alive by fighting or fleeing from various unsavory characters. Well programmed fantasy adventures such as the Zork series[7] are often highly addictive to users of all ages.

The user of a text-based adventure is doing many of the things we would like our students to practice in reading classes. This type of activity, when

used in an educa
adventure reading
creating an enviro
the role of initiator
written for native
despite linguistic
difficult vocabula
Mystery House[5] ca
even at low-interm
self-pacing elemen
time to work at a
of such program
microcomputers. H
ture reading ma
vocabulary and car
whole range of rea

A second innova
the student in *crea*
computer might b
several different 'st
the selection is
mystery'—the first
displayed on the scr
ble actions in a mu
the student's selectio
now begin to unfold
followed by new ch
participate in the cr
it to their wishes. A
functions, rhetoric,
possible branches ca
riculum. Thus stud
highly motivated ma
ly selected reading

Both creative read
related to true *simu*
to present a model o
albeit on a very limit
to experiment with.
foreign languages, a
are *Les Campagnes N*
Cutthroats.[8] In these
trol and must make
a restricted number
simulated and show
decisions can then be
tions clearly offer co
experience for the re
to those with adve
programs.

Further opportuni
level can be provided
exercises (Hall, 10)
plemented on the com
examined at more len

used in an educational setting, is referred to as *adventure reading*. It is collaborative in nature, creating an environment in which the student takes the role of initiator. Students find that the materials written for native speakers are highly motivating despite linguistic obstacles such as the sometimes difficult vocabulary. The use of programs such as *Mystery House*[5] can be recommended with students even at low-intermediate proficiency, because the self-pacing element of computer use allows them time to work at a comfortable speed. A number of such programs are already available for microcomputers. How much more effective adventure reading materials with properly graded vocabulary and careful attention to developing the whole range of reading skills would be!

A second innovative use of CALL is to involve the student in *creative reading* experiences. The computer might begin by providing a menu of several different 'story' titles to choose from. Once the selection is made—for example, 'murder mystery'—the first few paragraphs of the story are displayed on the screen, followed by several possible actions in a multiple-choice format. Based on the student's selection, completely different stories now begin to unfold, with several more paragraphs followed by new choices. In this way, students can participate in the creation of the story and adapt it to their wishes. At the same time, the notions, functions, rhetoric, and vocabulary in all of the possible branches can be keyed to the reading curriculum. Thus students would be working in a highly motivated manner with a number of carefully selected reading skills.

Both creative reading and adventure reading are related to true *simulations*. These are an attempt to present a model of some part of the real world, albeit on a very limited scale, for the computer user to experiment with. Examples of simulations in foreign languages, available on microcomputers, are *Les Campagnes Napoléoniennes* and *Cartels & Cutthroats*.[8] In these simulations, the user is in control and must make regular decisions concerning a restricted number of factors. The results are simulated and shown by the computer, and new decisions can then be made on that basis. Simulations clearly offer collaborative types of language experience for the reading class which are similar to those with adventure and creative reading programs.

Further opportunities to work at the discourse level can be provided by techniques such as cloze exercises (Hall, 10) which can rapidly be implemented on the computer. Cloze methods will be examined at more length in the section on testing. However, as Jones (18) has pointed out, testing or 'monitoring' is an intrinsic part of the teaching process in every language classroom in order to check and provide feedback on moment-to-moment learning and is quite distinct from formal 'examination' testing. Thus, cloze and other 'tests' have potential value as teaching instruments, particularly when computerized so as to provide immediate, flexible, and individualized feedback. In one recent project, a mixture of cloze and comprehension-checking techniques was reported to have produced positive results in terms of reading strategies and comprehension (Wells and Bell, 39).

It should be apparent that reading and vocabulary constitute an area of the curriculum in which computer-aided techniques appear to have impressive potential. The possibilities include instructional, collaborative, and facilitative types of programs. However, there is an urgent need for further research and testing in order to verify that the theoretical benefits can be realized. One writer with considerable experience in self-instructional materials has found great difficulty in working with unsimplified readings in Latin (Lacey, 21). However, these self-instructional materials were not computer-based, and it may be that the power of the computer would have helped to overcome the problems.

Listening

Overview of Listening

In the area of listening skills, it is necessary to acknowledge the limitations inherent in computerization before making a meaningful assessment of the potential contribution of CALL. In the case of reading, basic computer systems are quite capable of delivering on any types of activity we have discussed. For listening activities, however, the basic systems alone are generally insufficient.

The least expensive way of adding speech capability to a computer system is the obvious one of providing a separate, manually controlled cassette player and tape cassettes to accompany the usual computer programs. The program would first display instructions on the computer screen, telling students which part of the tape they must play to begin the lesson. After listening to the passage on the cassette, students answer questions by interacting with the computer keyboard and screen. Confirmation of answers, other hints and messages, and all branching actions could only be provided by the computer. The tape would necessarily be limited to providing the listening materials. For example, in the course of many activities, it might be desirable for students to listen again to specific small segments of the recorded material. With this basic hardware, and the cassette player under manual control, students would do the 'branching' themselves in order to locate the appropriate segment. In short, while manual operation of the player might be workable in some activities, it would generally be a clumsy and uncertain method.

With mini- and mainframe computers designed specifically for educational purposes, sophisticated random access audio devices[9] may be available. With certain microcomputers, the situation can be slightly improved by adding an inexpensive remote controller by which the computer program would take over cassette operation.[10] Unfortunately, stopping and starting with such devices can be inexact, for the starting point of the desired segment is not always precisely located. Even worse, errors may arise if the controller misses a stopping signal on the tape and continues on to the next stop signal. From this point on, the tape will remain out of synchronization with the computer program.

Much greater convenience and precision can be achieved with a random access audio device. In one model, the listening passages are recorded on a large floppy diskette and rapid, exact access can be obtained to any speech segment on the disk.[9] Feedback and branching is now no longer limited to the computer screen. Spoken responses, hints, and instructions can be selectively played back to students, depending on their answers, from the audio unit. Unfortunately these devices are often more expensive than the microcomputers or terminals to which they are attached. Alternatively, the use of a videodisc player under computer control could provide many of the same advantages together with the capacity for full accompanying visual materials. Videotape recorders can also provide visual and audio materials under program control. While quite effective, they suffer from slowness of operation and lack of precision when compared with videodiscs.

In summary, the potential of CALL in the area of listening skills depends very much on the type of hardware available to the teacher. At the low end of the scale, the manually controlled cassette and computer combination offers some significant advantages over the classroom situation in terms of self-pacing, individualization, and interaction with immediate feedback on errors. On the other hand, more expensive hardware such as random access audio devices, and particularly videodisc systems have the potential to make a tremendously greater impact.

It would be a disservice to leave this discussion without a few words about the often-cited comparison between CALL and the language laboratory. Listening skills have often been relatively neglected in the lab context, with speaking receiving a much greater emphasis. Both areas have suffered from a dearth of interesting and imaginative commercial materials. However, a case can be made that listening comprehension is an area in which the traditional language lab can play a more valid role than in the development of speaking skills.

Even with well conceived listening materials, the lab suffers from one crucial disadvantage—its lack of interactivity. Immediate feedback on inaccurate comprehension, hints on what to listen for, transcriptions in cases of difficulty to show the contractions and reduction in speech—all of these have generally been absent in the lab. However, all of them can be provided as necessary via the computer screen. The computer is not likely to 'go the way of the language laboratory;' rather, just one of its functions will be to work with the tape recorder to considerably enhance the role of the lab in the area of listening skill development.

CALL and Standard Listening Activities

Listening skills have generally received formal attention at all levels of language learning. Recently, a number of factors have led to renewed interest and reexamination of this area of the language curriculum. The stages in the development of listening skills have been documented (Taylor, 36), and approaches to designing and developing listening comprehension courses have been examined in detail (Richards, 32). We will attempt to assess the role of the computer in some of the main aspects of listening as outlined by these and other authors.

In skill-building exercises aimed at developing the ability to *discriminate* important speech features, CALL has a wide variety of applications. This type of exercise includes the recognition of differences between pairs such as *p/b, can/can't,* and *sixty/sixteen* either in isolation or in the normal stream of speech. Students vary widely in their ability to make these discriminations and in their need for practice. Using taped materials combined with questions, hints, and answers on the video screen, computers can offer a high level of individualization for more effective, student-centered instruction than is possible with tapes alone.

Using a remote-controlled cassette player, language materials of this type have been developed for certain microcomputers.[10] With its random access audio device and touch-sensitive screen, the

PLATO system offers some sophisticated materials at this basic skill level. In a Chinese lesson on numbers, students listen to a number and must touch one of several characters displayed on the screen to indicate accurate comprehension (MacDonald and Chen-Rohrbach, 23).

Perception as well as production of the prosodic features of language can be effectively learned through largely self-study materials (Mantini, 24) once students are past the absolute beginner stage. It is to be expected that the additional power of the computer in self-study applications should enhance this process.

CALL also offers advantages in more *extended listening* activities. Consider a story or lecture presented for listening comprehension via a random access audio recorder under computer control. The passage could be stopped at appropriate points to check on students' comprehension of general points and important details and to ask them to predict what is likely to come next. Should they be unclear on such points, they can automatically be routed back to listen to the section containing the information they missed before continuing with the passage. For example, Russian lessons on the PLATO system involve the presentation of a short situation comedy in which students can also see the dialogue displayed on the screen while they are listening to it. This is followed by questions to check on comprehension (Dawson and Provenzano, 7).

Note-taking activities can similarly be enhanced through CALL with random access audio equipment. At the end of a spoken passage, for example, students use their notes to help them complete a partial outline displayed on the screen. At each step, gaps in their notes or misunderstandings can be handled by a computer-managed return to the appropriate section of the listening passage. If they have further trouble, a variety of hints and suggestions can be provided, possibly including a transcription of that listening segment as a last resort.

New Possibilities in Listening Activities

In general terms, the main new contribution of computer-assisted learning in the area of listening is likely to be a change in the role of the learner. CALL makes it possible to turn over much control and responsibility to the student while providing very attractive course materials with the desired curricular requirements built in. It should be stressed again, however, that the realization of this potential in listening activities will be more highly dependent on hardware availability than in other skill areas. For most basic activities, the computer system would need to be supplemented with a cassette recorder and headset as a minimum, and some of the most worthwhile activities would require further hardware.

An attractive possibility with even the basic equipment is a series of *listening laboratory* materials. This would comprise a set of graded listening passages, recorded on audio cassettes, with a good selection of passages at each level. Given such a choice, students would be likely to find passages at each level on topics that were genuinely interesting to them. Computer programs would accompany each taped passage. Students would first listen to a passage and then work with a range of different types of computer-based activities.

Materials similar to this have already been developed for self-study without a computer (Byrnes *et al.*, 4). With computerization, one of the main advantages would be increased flexibility. In a workbook approach, students check themselves by looking at the correct answers. If they are wrong, there is obviously no further possibility of working toward the correct answer. With the computer, however, the correct answer need not be revealed when students' answers are first evaluated. If wrong, the computer can provide students with a wide range of hints, prompts, and partial answers. CALL programs can also provide various 'help' materials on points of rhetoric, culture, and the like. Using these tools at their discretion, students can listen to sections of the passage again and continue to try to work out the answers for themselves.

One further activity feasible with even the basic hardware is a modified version of the *cloze* format. In this adaptation, activities are based on listening passages recorded on cassette tape with accompanying computer programs. Students begin by listening to the passage once. In the spoken cloze approach, they must then complete the blanks in the same passage, now displayed on the computer screen.

Listening-oriented cloze activities of this type can easily be prepared by the teacher provided that the appropriate 'utility' programs are available. In this case, the utility program would enable teachers to type in their cloze materials at the keyboard. These would then be stored in diskette or other form for student use. No knowledge of computer programming would be needed in this process. The cassette recordings would be prepared in the usual way. The exercises could thus be focused on language and topics drawn directly from current class materials. The activities could also be given a competitive or game-like atmosphere by introducing points added or lost depending on the answers, bonus points for several consecutive right answers, and so on. It is instructive to compare these activities with a traditional pen-and-paper or workbook dictation exercise. Similar as the objectives are, the computer-based activities have very compelling advantages in terms of learning opportunities, motivating power, and sheer enjoyability.

Dictation exercises could also be enhanced through computerization. During the initial phase of a dictation, the computer would essentially function as a word processor, with students typing the passage in a straightforward manner. Following this, the computer would begin to check through the dictation a sentence at a time. Errors would be indicated so that the student could listen again and try to correct the mistakes. If random access audio were available, the replay of the appropriate listening segment could be provided automatically. Various 'help' functions could also be provided. Dictation techniques will be discussed further in the section on testing.

Of all computer-assisted listening activities, however, those with the greatest potential impact will require an additional item of hardware—a computer-controlled random access videodisc player. There are two main types of videodisc player currently available: *laservision* and *CED*. The CED technology, popularized by RCA, has been unsuitable for educational use because it lacks a random access capability. (RCA has begun demonstrating a prototype CED player with this feature, however.) The laservision format is inherently capable of random access and has been increasingly used for educational purposes. In its most common current form, the laservision player can provide rapid access to any of the 54,000 picture frames on one side of the videodisc. Each frame can be shown on the screen, alone or with superimposed computer displays, turned on or off, or used as the start of a movie sequence which can be 'frozen' at any of its frames. While a film sequence is running, either or both of two hi-fi audio tracks on the disc can also be played. In short, the videodisc and player can provide a full visual context as well as the usual audio material, and can operate very rapidly and accurately under computer or manual control. This capability has tremendous implications for the design of listening skill materials.

The presentation of the visual context along with the spoken material has a number of important effects. Perhaps the most obvious and dramatic is the power to involve students and hold their attention. Well produced disc materials are generally extremely enjoyable to use without sacrificing any instructional potential. Few traditional materials can consistently match this capacity, and its value should not be underestimated. A second important feature is the presentation of the full normal range of extra-linguistic clues to meaning together with speech. This permits students to grasp or confirm the meanings and structures of spoken language elements which were not previously within their competence to a much greater extent than would audio material without a visual channel. Videodisc-based CALL can thus provide opportunities which maximize the potential for language acquisition. Three examples of videodisc-based programs are Macario (Melendez, 25) and Montevidisco (Schneider and Bennion, 34), both in Spanish, and Klavier im Haus (Luckau, 22).

The videodisc may also be an important medium for theorists who advocate extensive exposure to spoken language before requiring students to begin active language production. In practical terms, it certainly makes possible the use of relatively advanced material by lower-level students who can use the visual clues and view the difficult sections repeatedly until the material is understood. They can also be provided with various hints and other types of help from the computer as needed. Their motivation in the face of material that might otherwise be frustratingly difficult is maintained at a high pitch through the attractiveness of well designed disc programs.

Using computer-controlled videodiscs, a very wide range of listening activities is possible. For example, after viewing a section on the disc, student comprehension of important detail could be checked via the screen. Whenever a mistake is made, the specific speech segment that the student misunderstood can be replayed rapidly and precisely, with accompanying video. In addition, a full range of 'help' materials could be provided by the computer for student use when needed. These would probably include assistance with standard points of vocabulary, speech features and the like, but some new items would become possible with the visual medium. In particular, cultural aspects such as table manners and distance between speakers could be very effectively addressed. In fact, a wide range of diverse aspects of language and culture could be covered with the same film material through different computer programs accessing a single disc in completely separate sequences in order to focus on different points.

Finally, let us return to a more basic computer system with a cassette recorder under computer

control to discuss
acts as game mas
the student "list
or conversational
are explained as r
telephone, and s
trying to make c
ted, identities of
obtained, etc. St
and short conve
one at a time.
whether they are
Each new conve
but students ga
Whenever the fi
maining conver
skipped, the sec
series of clues f

This type of a
the computer to
necessarily reve
provide very fle
on those answer
ty has been intr
to illustrate a po
activities: while
by individual st
groups of stude
interactions.

Testing the Rec

Overview of Te

It seems clea
significant part
of the compute
proach to testi
been the subject
the last decade.
have argued in
discrete-point te
data as provid
'unitary factor
competence un
argues against
language as se
research which
separation—in
ferred (Palmer
differences, co
integrative test

In terms of
point testing o
few apparent p
in all compute
main. Test any

control to discuss an activity in which the computer acts as game master. In these *deduction* activities, the student "listens in" to various conversations or conversational fragments. In one scenario, these are explained as resulting from a tap on a suspect's telephone, and students play the role of detective, trying to make deductions about crimes committed, identities of the criminals, value of the loot obtained, etc. Students are given the first question, and short conversations or fragments are played one at a time. After each, students are asked whether they are ready to make the first deduction. Each new conversation makes the problem easier, but students gain fewer points for solving it. Whenever the first question is answered, the remaining conversational clues for that point are skipped, the second problem is explained, and the series of clues for that problem begins.

This type of activity again uses the capability of the computer to judge students' answers without necessarily revealing the correct solution, and to provide very flexible types of responses depending on those answers. However, the deduction activity has been introduced at the end of this section to illustrate a point that is true of many 'listening' activities: while they can be worked on separately by individual students, they can also be used with groups of students to stimulate a whole range of interactions.

Testing the Receptive Skills

Overview of Testing

It seems clear that the computer can play a significant part in language testing. The precise role of the computer, however, depends on the approach to testing which is adopted, and this has been the subject of intense research and debate over the last decade. Oller and Perkins (27) and others have argued in favor of integrative rather than discrete-point testing and have interpreted research data as providing considerable support for the 'unitary factor' hypothesis. This posits a general competence underlying all language ability which argues against the testing as well as teaching of language as separable skills. Others have cited research which suggests that at least a twofold separation—into speaking and reading—can be inferred (Palmer *et al.*, 28). To accommodate these differences, computer-assisted discrete-point and integrative testing will be treated separately below.

In terms of computer applications in discrete-point testing of the receptive skills, there are very few apparent problems. One general consideration in all computer-based testing is the affective domain. Test anxiety and general nervousness could lead to "keyboard shock" if students are faced with an unfamiliar computer keyboard for the test. Simply exposing students to the same keyboard during the semester should solve this problem for all but new arrivals. The limitation in reading tests of the small amount of text displayable on some computer screens can be overcome in a number of ways including the use of the screen as a "window" which can be moved backwards and forward through a long reading passage. In listening tests, the basic computer system would need to be augmented with at least a computer-controlled audio cassette player. However, for reasons of hardware reliability and test security it would be desirable to have a random access audio device or videodisc under computer control. Provided these limitations are avoided or overcome, the computer offers impressive advantages in test development, administration, and scoring.

CALL and Standard Testing Methods

Computer-assisted techniques can contribute in a number of important ways to traditional testing activities in reading and listening. One possibility is the provision of a systematic method of building a *test bank* of items. At the simplest level, this would require only that the teacher have access to a computer, as the actual tests would be printed out, duplicated, and administered to students in the traditional pen-and-paper form. The computer would function as a structured word processor, allowing teachers to input and edit sections of related test items over a period of time. Once the item bank has been established, the computer can be asked to generate a 'new' test by randomly choosing ten questions from section one, five questions from section two, and so on, printing them out in correctly numbered form with the standard instructions included automatically. For listening tests, the provision of the audio material would be greatly simplified if the teacher could use a random access audio device for test administration in class. In this way, 'new' tests can be generated (together with answer keys if included in the program) each time the course is given with virtually no further teacher effort. This ability to produce 'new' tests also has the effect of enhancing test security (Berger *et al.*, 2).

At a more ambitious level, the items in each section could be standardized so that the computer generates and prints out a different test for each student. The answers could then be entered into the computer, which would perform the necessary processing of raw scores to equalize the test results. While attractive on several counts, this sophisticated test generation approach would require far more time and expense in its development.

If sufficient computer systems are available, tests can also be administered to the student at the computer keyboard. Again, reading tests can be handled satisfactorily with the most basic computer equipment, whereas convenient administration of new versions of listening tests demands a random-access audio device. This type of *direct administration* represents a higher level of computer involvement in testing but offers numerous benefits. Naturally, all scoring and grading of the tests can be done instantly by the computer, with the results available for display on the video screen, storage on diskette, or printing out in report format. For research and development applications, much of the routine labor involved in item analysis, statistical computations, correlation studies, and test standardization would be eliminated. Diagnostic tests can also be directly tied into specially tailored courses of study, including CALL materials if desired.

Turning the focus to integrative tests, the two major candidates for computer implementation appear to be cloze and dictation. The cloze technique is not limited to reading—'spoken cloze' methods can also be used for testing purposes. Apart from the additional hardware required for the listening version, cloze methods appear to be highly adaptable to the computer. Computer administration is simple, and since students are not involved in large amounts of typing, keyboard obstacles are minimal. In one type of cloze, answer judging and scoring are limited to simple right/wrong evaluation. In more flexible approaches to cloze, where alternate words or spelling mistakes are permitted, the more sophisticated pattern markup judging that is necessary (Pusack, 30) can be provided even on microcomputer systems.

Dictation techniques, however, present considerably more difficulties for computerization. As mentioned above, the amount of typing at the keyboard required in dictations poses a serious obstacle to same students, introducing an undesirable variable into the testing process. Satisfactory computerized answer-judging might also be difficult to achieve. In the technique recommended by Oller (26), a degree of subjectivity is introduced because exact spelling of words is not required. Many teachers also score dictations by subtracting points for deletions, insertions, and the like, a method which expands the scope of subjective judgment. These approaches might cause problems in a computerized pattern markup and scoring system. Fortunately, recent research seems to indicate that a relatively rigid, objective approach to dictation scoring can produce comparable results and similar reliability (Cziko, 6). This type of scoring would still demand a pattern markup approach, but it should be generally well within the capabilities of educational computer equipment.

A parallel to the concept of dictation has been developed for reading skills—the 'copytest' technique. In one implementation, the reading passage is first briefly shown on the screen in its entirety, then in small segments followed by pauses (with a blank screen) to permit students to write down what they remember, and finally again in its entirety for a short period (Cziko, 6). The initial assessment of this copytest technique was very positive. Clearly, it has the potential to be transferred entirely to the computer medium. Using a split screen, the target passage could be presented exactly as described above. However, students would 'write' their version of the passage on the bottom half of the screen, using the computer as word processor. When finished, the computer could then automatically score the test as with any dictation. It is apparent that there are many questions, which can only be answered by research, about the feasibility of this interesting option. Indeed, verification of the effectiveness of computer-aided methods is needed in many areas, although the studies so far are encouraging (Ragosta *et al.*, 31).

New Possibilities in Testing

Computer involvement offers a very attractive possibility, particularly for discrete-point assessment methods—*computer-adaptive testing*. In one form of computer-adaptive test, the test items are divided into sections covering different language points, but within each section several levels of difficulty are distinguished. To take an ESL example, section five might deal with pronouns; items on *you* and *we* might be level one (beginning) in difficulty, whereas test items on *yourself* and *ourselves* might be considered level three (intermediate). The computer-adaptive proficiency or placement test would continually adjust to the student's level as established in the test so far. Thus, students whose answers showed they were beginners would rarely or never receive items at the intermediate or advanced level since these would be much too hard for them, and their wrong answers would prove little and waste time. Instead, the computer would automatically present items pitched at around the student's perceived ability level until a consistent estimate of proficiency was established.

Compare this procedure with a traditional paper-and-pencil reading test, in which many of the fixed set of items are either much too easy or much too difficult for particular students. It is clear that branched tests are potentially more accurate and less time-consuming to administer than traditional tests. They are also more humane in that fewer of the items will be beyond the students' competence. Again, an adaptive listening test would require a random-access audio device as a minimum to permit the necessary branching to occur quickly and reliably.

Despite their labeling as "new possibilities," branched test concepts have long been understood although they have received little general attention so far. In fact, branched tests in non-oral skills do not require computer equipment for their administration. Branching based upon students' answers can be built in to an otherwise conventional-looking test booklet. Students do not move from one question to the item immediately following it, but instead, depending on the answer selected from the alternative choices to one question, students are directed to turn to quite different 'next items' in the booklet. Even with this simple form of administration, many of the advantages of adaptive testing are present: improved test security, accurate assessment of student proficiency, and better diagnosis of individual students' problem areas (Boyle *et al.*, 3). The branched French test described by Boyle, although developed and scored with the aid of a computer, does not use computers in its administration. With the increased availability of the computer, however, it would provide tremendously greater convenience to complete the process by offering it in computer-adaptive mode.

Conclusion

This paper has attempted to present and assess a wide range of possibilities for computers in the learning and testing of reading and listening. Within this limited scope it has been necessary to set aside a number of important issues, not least among which is the relative importance of different types of computer involvement. What should our priorities be? This is a subjective yet crucial topic. A second important question not addressed in this paper is the time, energy, and expense involved in different types of projects (Holmes, 14). Where is it most feasible to begin? If the paper has established a framework and perspective in which questions such as these can be more clearly discussed, it will have served its purpose.

NOTES

[1]PLATO: computer-assisted learning system based on mainframe computers. [PLATO; Computer-Based Education Research Laboratory (CERL); University of Illinois; Urbana, IL 61801; or PLATO; Control Data Corporation; 3100 South Avenue; Minneapolis, MN 55440.]

[2]TICCIT: computer-assisted learning system based on minicomputers. [TICCIT; Hazeltine Corporation; 7680 Old Springhouse Road; McLean, VA 22101.]

[3]CCC (Computer Curriculum Corporation): a computer-assisted learning system based on minicomputers. [1070 Arastradero Road; Palo Alto, CA 94304.]

[4]CONDUIT: non-profit organization which distributes tested educational software and endeavors to make it available on a wide variety of computers; journal *Pipeline*, twice a year. [M310 Oakdale Hall; University of Iowa; P.O. Box C; Oakdale, IA 52319.]

[5]Gessler Publishing: large range of foreign language courseware for microcomputers. [900 Broadway; New York, NY 10003.]

[6]The Linguist: a program with a wide range of alphabets and characters is available for some microcomputers. [Synergistic Software; 5221 120th Avenue SE; Bellevue, WA 98006.]

[7]Zork: adventure game series available for microcomputers. [Infocom; 55 Wheeler Street; Cambridge, MA 02138.]

[8]Le Professeur: organization distributing a wide range of French adventure programs and simulations for microcomputers. [P.O. Box 301; Swanton, VT 05488.]

[9]EIS (Education and Information Systems): random access audio device controllable from microcomputers. [804 North Neil Street; P.O. Box 1774; Champaign, IL 61820.]

[10]Hartley Courseware: audio cassette control device operated from a microcomputer program; pronunciation lesson programs which use the device. [P.O. Box 431; Dimondale, MI 48821.]

REFERENCES

1. Baudoin, Margaret and Ellen Bober, Mark Clarke, Barbara Dobson, and Sandra Silberstein. *Reader's Choice*. Ann Arbor, MI: University of Michigan Press, 1977.
2. Berger, Louis *et al. A Methodology to Achieve Secure Administration of English Comprehension Level Tests*. San Antonio, TX: Southwest Research Institute, 1971.
3. Boyle, Thomas A., W. Flint Smith, and Ruth G. Eckert. "Computer Mediated Testing: A Branched Program Achivement Test." *The Modern Language Journal* 60 (1976): 428-40.
4. Byrnes, Adelheid S., Stefan R. Fink, and Aurelia Roman. "Enhancing Second Language Acquisition by a Focus on Listening Comprehension: The Potential of the Cassette." *Foreign Language Annals* 15 (1982): 37-46.

5. Curtin, C. and C. Dawson. *The Cyrillic Alphabet* [software program]. Urbana, IL: University of Illinois, 1976.

6. Cziko, Gary A. "Improving the Psychometric, Criterion-Referenced, and Practical Qualities of Integrative Language Tests." *TESOL Quarterly* 16 (1982): 367-79.

7. Dawson, C. and N. Provenzano. "PLATO Sitcom Dialogues for Russian." *Studies in Language Learning* 3 (1981): 92-97.

8. Fry, Edward. *Reading Faster*. Cambridge: Cambridge University Press, 1963.

9. Geoffrion, Leo D. and Olga P. Geoffrion. *Computers and Reading Instruction*. Reading, MA: Addison-Wesley, 1983.

10. Hall, Wendell H. "A Cloze Procedure Program for Testing Reading Comprehension." *MICRO* 1 (1980): 17-20.

11. Harris, David. *Reading Improvement Exercises for Students of English as a Second Language*. Englewood Cliffs, NJ: Prentice-Hall, 1966.

12. Hart, Robert. "Language Study and the PLATO System." *Studies in Language Learning* 3 (1981): 3-6.

13. ________ and N. Provenzano. *Russian Concentration* [software program]. Urbana, IL: University of Illinois, 1973.

14. Holmes, Glyn. "Creating CAL Courseware: Some Possibilities." *System* 11 (1983): 21-32.

15. ________ and Marilyn Kidd. "Serving Learners' Needs: From Teletype to Micro." *System* 9 (1981): 125-32.

16. Hosenfeld, Carol, Vicki Arnold, Jeanne Kirchofer, Judith Laciura, and Lucia Wilson. "Second Language Reading: A Curricular Sequence for Teaching Reading Strategies." *Foreign Language Annals* 14 (1981): 415-22.

17. Jamieson, Joan and Carol Chapelle. "Language Lessons on the PLATO IV System." *System* 11 (1983): 13-20.

18. Jones, Christopher. "Computer Assisted Language Learning: Teaching or Testing?" *ELT Journal* 37 (1983): 247-50.

19. Kemmis, Stephen, Roderick Atkin, and Eleanor Wright. *How Do Students Learn? Working Papers on Computer-Assisted Learning: UNCAL Evaluation Studies*. Norwich, England: Centre for Applied Research in Education, 1977.

20. Krashen, Stephen. *Second Language Acquisition and Second Language Learning*. London: Pergamon, 1981.

21. Lacey, Douglas N. "The Limits of Self-Paced Instruction in Latin." *Foreign Language Annals* 13 (1980): 399-402.

22. Luckau, Paul. "Klavier im Haus—An Annotated Version," pp. 127-37 in *Proceedings of the FLINT Conference*. Monterey, CA: Defense Language Institute, 1983.

23. MacDonald, W. and P. Chen-Rohrbach. *Numbers* [software program]. Urbana, IL: University of Illinois, 1981.

24. Mantini, Lawrence C. "The Use of a Structured Sequence of Recorded Materials for Language Stress-Rhythm and Intonation Study." *Foreign Language Annals* 13 (1980): 373-77.

25. Melendez, Gloria S. "Interactive Instruction in Spanish Interfacing the Microcomputer and the Videodisc." *MICRO* 1 (1980): 11-12.

26. Oller, John. *Language Tests at School*. London: Longman, 1979.

27. ________ and Kyle Perkins, eds. *Research in Language Testing*. Rowley, MA: Newbury House, 1980.

28. Palmer, Adrian, Peter Groot, and George Trosper, eds. *The Construct Validation of Tests of Communicative Competence*. Washington, DC: TESOL, 1981.

29. Parker, D.H. *SRA Reading Laboratory IIIb*. Chicago: Science Research Associates, 1963.

30. Pusack, James P. "Answer-Processing and Error Correction in Foreign Language CAI." *System* 11 (1983): 53-64.

31. Ragosta, Marjorie, Paul Holland, and Dean Jamieson. *Computer-Assisted Instruction and Compensatory Education: the ETS/LAUSD Study*. Washington, DC: U.S. Department of Education, 1982.

32. Richards, Jack. "Listening Comprehension: Approach, Design, Procedure." *TESOL Quarterly* 17 (1983): 219-40.

33. Scanlan, R. and M. Vollmer. *Word Power* [software program]. Urbana, IL: University of Illinois, 1975.

34. Schneider, Edward and Junius Bennion. "Veni Vidi Vici Via Videodisc: A Simulator for Instructional Conversations." *System* 11 (1983): 41-46.

35. Sonka, Amy. *Skillful Reading*. Englewood Cliffs, NJ: Prentice-Hall, 1981.

36. Taylor, Harvey. "Learning to Listen to English." *TESOL Quarterly* 15 (1981): 41-50.

37. U.S. Department of Education. *Instructional Use of Computers in Schools*. [Fast Response Survey System]. Washington, DC, 1982.

38. Weible, David M. "Teaching Reading Skills

Through
Language

39. Wells, Be
to Teachi
and Comp
Technolo

40. Widdowso
municatio

Cultural M
ject, Hampto

Designed to
in beginning c
African herita

La Famil
a traditio

Un Conte
using anin
oral litera

Dos Cami
writer fro
to leave a

Through Linguistic Redundancy." *Foreign Language Annals* 13 (1980): 487-93.

39. Wells, Bethany and Scott Bell. "A New Approach to Teaching Reading Comprehension: Using Cloze and Computer-Assisted Instruction." *Educational Technology* (March, 1980): 49-51.

40. Widdowson, Henry G. *Teaching Language as Communication.* Oxford: OUP, 1978.

41. Wyatt, David H. "Computer-Assisted Language Instruction: Present State and Future Prospects." *System* 11 (1983): 3-11.

42. ________. *Computers and ESL.* Washington, DC: Center for Applied Linguistics, 1984.

43. ________. "Teaching the Receptive Skills," pp. 45-50 in *Proceedings of the FLINT Conference.* Monterey, CA: Defense Language Institute, 1983.

Limitations of Current Microcomputers for Foreign Language Training

Paul K. Aoki
青木健ポール
Department of Defense

The Department of Defense (DOD) has a foreign language requirement to train students up through the Interagency Language Roundtable's (ILR) skill level three, especially in the receptive skills of reading and listening. In addition to this requirement, the DOD is faced with several resource challenges in the teaching of foreign languages. These include a shortage of well-trained language instructors, both native and non-native, an increasing number of students, and a limited period of time for training. Given the language requirement and the resource challenges, the DOD is investigating automated technologies with the hope that they will provide substantive tools to supplement, enhance, and accelerate foreign language training.

Level 3 Descriptions for Reading and Listening

The ILR level three description for reading includes almost complete comprehension at normal speed with authentic prose in native orthography on unfamiliar subjects. Texts consist of news stories, routine correspondence, general reports, and technical material in the reader's field of expertise. These will all include hypothesis, argumentation, and supported opinions. Such texts also typically contain grammatical patterns and vocabulary ordinarily encountered in professional reading. Misreading should be rare. The student should almost always be able to correctly interpret material, relate ideas, and make inferences. There should be little need to pause over or reread general vocabulary. It should be noted that a majority of the languages taught at the DOD are not of Indo-European origin, and therefore, for graphic purposes, require orthographies other than Roman scripts with minor diacritic variations. Such orthographies are not necessarily written horizontally from left to right and some have very large character sets, e.g. Farsi and Chinese.

Listening at level three should include the ability to understand the essentials of all speech in a standard dialect. The student should have effective understanding of face-to-face speech, delivered with normal clarity and speed in a standard dialect, on general topics and areas of special interest. Hypothesizing and supported opinions should be understood. The student should possess a broad enough vocabulary that there is little need to ask for paraphrasing or explanation. Accuracy sufficient to follow the essentials of live or reasonably clear recorded conversations between educated native speakers, radio broadcasts, oral reports, some oral technical reports, and public addresses on non-technical subjects is expected.

The level three descriptions indicate that the student should have a sufficient background in the fundamentals of morphology and syntax, and that the desired skill is the ability to extract the appropriate semantic content from text and discourse passages. In many cases, students already have four years of college foreign language study and are able to handle meaning at a sentential level to varying degrees. The additional training that they require is learning how to establish coherent semantic content from larger than sentential units. Notice that the ability to comprehend individual sentences of a text does not insure that a concatenation of those sentences will result in a comprehension of the full text as an integrated whole.

The Computer Contribution

In his paper on computer-assisted testing and teaching, Wyatt proposed innovative activities of a collaborative and facilitative nature, in addition to the more familiar instructive ones. The innovative activities appear to hold more promise than the instructional ones, especially in view of DOD's experience that traditional foreign language teaching methods have a poor success rate in bringing students up to ILR skill level three. It is possible that traditional methods are inherently in-