WORKING PAPER
ALFRED P. SLOAN SCHOOL OF MANAGEMENT

CULTURAL DIFFERENCES AND INFORMATION SYSTEMS TECHNOLOGY

Ramon C. Barquin and C. Lawrence Meador

March 1979

Sloan School Working Paper #1055-79

CISR #43

MASSACHUSETTS
INSTITUTE OF TECHNOLOGY
50 MEMORIAL DRIVE
CAMBRIDGE, MASSACHUSETTS 02139

CULTURAL DIFFERENCES AND INFORMATION SYSTEMS TECHNOLOGY

Ramon C. Barquin and C. Lawrence Meador

March 1979

Sloan School Working Paper #1055-79

CISR #43

CULTURAL DIFFERENCES AND INFORMATION SYSTEMS TECHNOLOGY

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

PROLOGUE

Although we allude in the title of this paper to Cultural Differences as they impact (and are impacted by) Information Systems (I/S) Technology, it seems necessary at this juncture to clarify what we mean by Information Systems in this particular context. Several new acronyms, terms, and other forms of jargon have developed around the field of information systems during the development of the discipline. Some of these include management information system (MIS), data processing (DP), computer-based information system (CBIS), decision support system (DSS), and so forth. We must admit a frank preference for Informatics, an evolving term for the broader field of information systems which is starting to become widely accepted internationally.

Informatics is concerned not only with understanding, developing, and implementing the technological components of information systems, but also with responding to the needs and requirements of users of the technology as well as evolving a better understanding of the dynamic interaction between the technology, people, and organizations in which they function.

We are concerned, in this discussion, with the problem of determining how best to develop the associated technologies and to deliver and implement them in a multi-cultural international setting. It is in this spirit that we offer the following suggestions and empirical insights.

I. INTRODUCTION

This paper is concerned with questions relevant to the transfer and implementation of computer-based information systems technology in cultural settings different from that in which the technology was originally conceived and developed. Let us consider the following issues:

- How is programmer productivity affected by having a non-English-speaking programmer code in a high level language like COBOL or PL/I, especially when his native language has a totally different lexical and syntactic structure from English?
- In a culture where the value of time is different from our own, what is the role of the system designed to save time?
- An 8-bit byte is quite adequate for encoding in our 26-letter alphabetic world; but what about those alphabets which are different from our own--such as Arabic, Thai, Korean, Cyrillic, Greek, or Hebrew? Or the non-alphabetic written forms with thousands of different characters, such as Chinese?
- Privacy is a big concern in the United States, and data banks are seen as prime targets for potential violations. What are the implications for cultures where privacy issues are treated and viewed in a totally different manner?
- How does the Arabic or Israeli programmer handle the problems associated with the fact that Arabic and Hebrew are read and written from right to left as opposed to the English left-to-right convention?
- What is the effect on the usual operations of a data-processing installation if certain activities which are considered manual

labor (such as keying data or a program into a terminal) are shunned by some segments of the culture?

- How does varying management style within a culture affect the design, implementation and operation of management information systems for different types of private or public sector organizations within the culture?
- How does the exclusion of women from the labor force affect the development of a nation's DP industry in settings where women are not allowed to work in significant professional capacities?
- What happens to productivity in the programming department, or the computer room, in a culture where one must fast for prolonged periods of time due to religious reasons, such as during the month of Ramadan in Moslem countries?
- How does resistance to change impact automation and computerization in different cultures? How does it affect new application development?

II. A FRAMEWORK FOR ANALYSIS

The questions alluded to above present very real problems and issues confronted day in and day out in various parts of the world. They are a clear reminder of the fact that we are all basically different and that each of us is in many ways unique. We must see this as a positive factor, an expression of individualism in the face of potential dehumanization and conformity imposed by the industrial/electronic revolution. But at the same time, it is a major source of complexity and difficulty for those people from "alien" cultures who are forced to use the computer in their daily work.

Technology, in its broadest definition, is no doubt present in virtually every facet of our daily lives. The same is true both in the United States and elsewhere. Technologies, moreover, do not operate in a vacuum. Rather, they are influenced by a series of social-psychological, economic, and political factors which in many ways define and characterize the environment in which the technology must work.

Thus, we can say that every technology operates in a cultural field and is under the effect of its component influences.

That is, each and every one of the components of a technology (i.e., hardware, software, products, standards, technical skills, processes) exist within an environment that is dominated by: social-psychological factors like language, values, customs, traditions, management style; political factors such as bureaucracy, legal structure, degree of nationalism; and economic factors like markets, inflation, taxation, distribution systems, tariffs and so on.

However, Webster's New Collegiate Dictionary defines culture as "the integrated pattern of human behavior that includes thought, speech, action and artifacts; and depends upon man's capacity for learning and transmitting knowledge to succeeding generations." From this definition and for the purposes of this paper, let us equate the term, "cultural," to "social-psychological."

When we limit ourselves to a specific technology--information systems technology, for example--we can identify more clearly some of the points mentioned. What are the relevant components that make up information systems technology? Among other, they are: the data processing hardware, the software, applications, technical skills, procedures, education manuals. And, of course, each and every one of these technological components is influenced by the cultural factors mentioned above.

In order to complete the picture we must add a third dimension. That is, me must assess these technological and cultural variables for each of the different relevant cultures in our world. The root of the problem has been the fact that information systems technology has been developed in a cultural environment that has been overwhelmingly dominated by the United States, while in actuality this technology is utilized and applied in a multiplicity of cultural settings.

What environmental and behavioral factors exist in a Japanese programming department or in a French computer room or in a Brazilian DP education center which make them different from similar installations in, say, San Francisco, Dallas, or Boston? The key must lie in the dynamics of the process whereby the technology impacts the host culture and is itself adapted to better fit the cultural environment in which it operates.

In order to study this problem within an appropriate conceptual framework, let us develop a three-dimensional construct with cultural variables as a first dimension, information systems technology as a second, and different cultures as the third. (See Figure 1.) This approach allows us to isolate any individual point or cell in the matrix identified by the intersection of a cultural variable, V, an information systems component, I, and a specific culture, C. This cell, of course, defines a perimeter of impact and interaction among the three elements. For example, let us take a cultural variable V(i) (i.e., language), an information systems technology component I(j) (i.e., Input/Output devices), and a specific culture C(k) (i.e., Arabic). Then cell (ijk) defines the impact perimeter of the Arabic language on the I/O units of a computer configuration. (See Figure 2.)

In the same manner we may identify the entire matrix of relevant intersections of these three dimensions, and thus isolate in cell (pqr) the impact of, say, Japanese management style on computer operations; or of Thai value system on programmer productivity; or of Hindu attitudes toward manual labor on data entry operations.

A key element in our discussion is relevance. But the concept of relevance is, in the first instance, subjective. What is relevant to one depends very much on who one is and what one's interests are. Yet there are some measures of relevance to larger sectors of humanity which may be gauged along economic, social, or political lines. To that effect, it becomes important to develop an initial list of relevant elements within each dimension of our impact matrix: the cultural variables, the technological components, and the cultures themselves.

THE IMPACT MATRIX

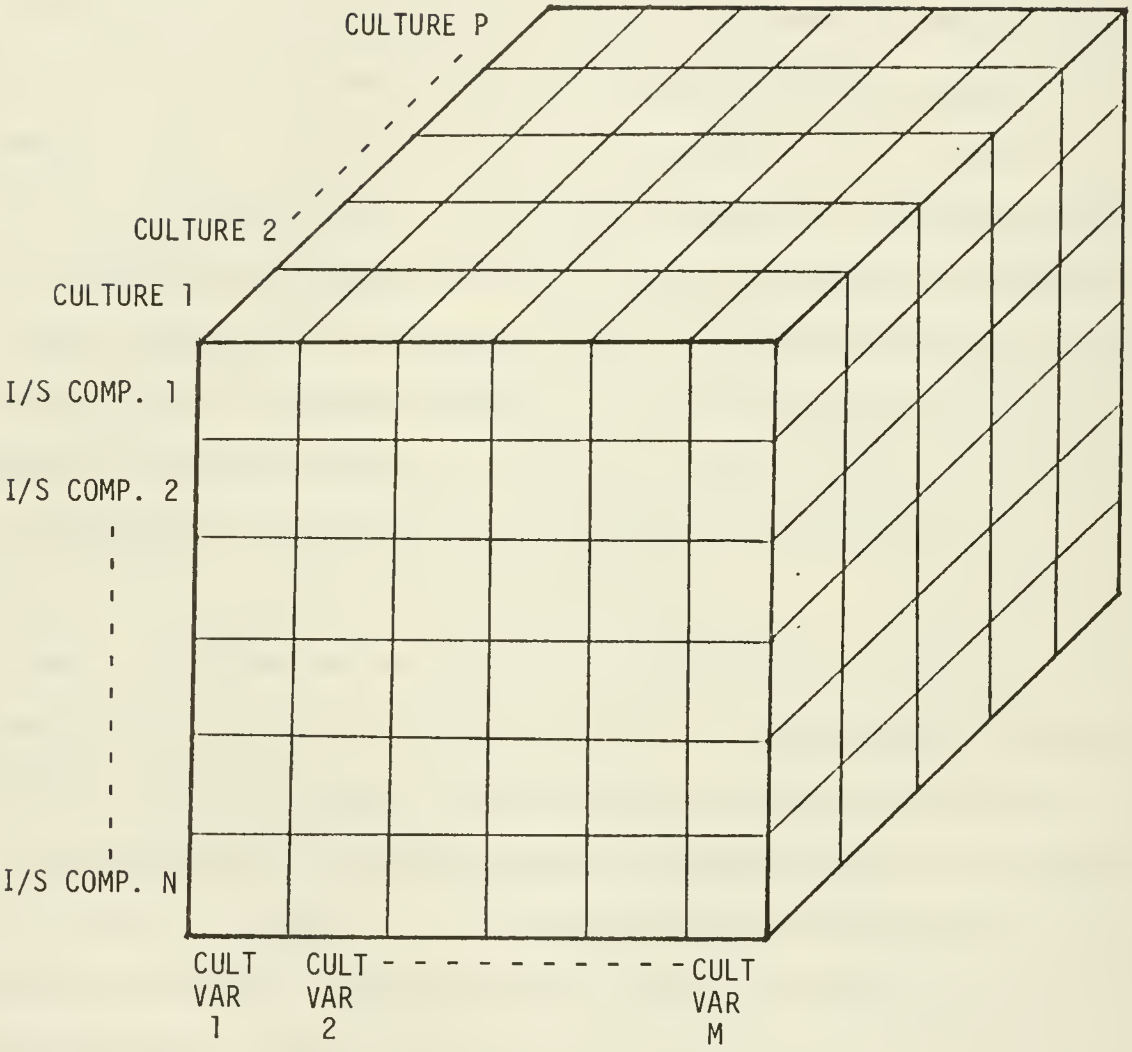


FIGURE 1

CELL (IJK)

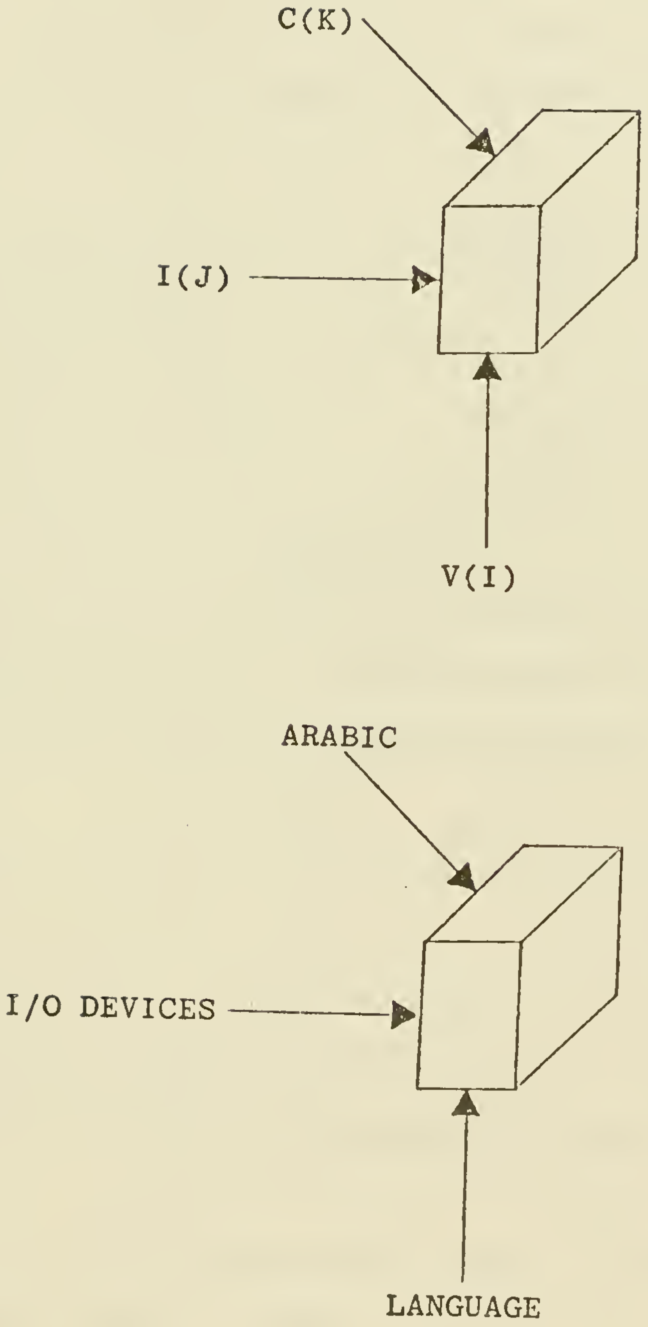


FIGURE 2

Taking, first of all, the components of Information Systems Technology, we can construct a finite set including each and every conceivable component. However, for our purposes here, let us work with a macro list which can be refined into micro lists as necessary. The macro list should include at least:

- Hardware
- Software
- Processes
- Applications
- Databases
- Technical Skills
- Education
- Documentation
- Physical Infrastructure
- Communications Facilities
- Support Services
- Management Skills
- Standards
- DP Organization

Of course, this is not an exhaustive list even at the macro level, but it will serve as a basic departure point of relevant categories. As it becomes necessary to investigate a new area, the list will have to be expanded.

By the same token, each element in the macro list can be broken down into further components as needed for a study. For example, Hardware would be decomposed into:

- Central Processor
- Input Units

- Output Units
- Main Storage
- Auxiliary Storage
- Peripheral Devices
- Communications Devices
- Other

Anyone wishing to study the impact of language on hardware could further decompose, as an example, as follows:

```
Output Units
   ...
  Printers
     ...
    Print Chains
       ...
     Character Sets
```

Looking at the Cultural Variables, we see a similar development taking place. Our macro list would include the following:

- Language
- Values
- Beliefs
- Attitudes
- Expectations
- Vital Assumptions
- Interpersonal Relationships
- Motivators
- Status
- Customs
- Social Structure
- Social Mobility
- Education
- Management Style

As needed, Attitudes, for example, might be further decomposed into:

- Attitudes toward automation
- Attitudes toward change
- Attitudes toward foreign technology
- Attitudes toward privacy
- Attitudes toward manual labor

The category, Values, for example, might necessitate some additional levels of specification in order to apply a meaningful research methodology:

- Value of time
- Value of information
- Value of work

We have given a definition of culture which satisfies our conceptual needs for purposes of this paper. However, in order to satisfy the eventual need to identify each culture with the geographic habitat of the people which share it, our macro list for culture will be a combination of political and geographic subdivisions and of cultural conglomerates. Rather, let us address them as cultural areas. Again, the key is the development of micro lists as necessary.

Two problems must be addressed in order to structure the research more tightly: the issue of political states, and the problem of subcultures. What is a subculture? Sociologists define it as a culture existing within another culture but differentiating itself substantially with respect to some of its component variables or traits. That is, it may often have different relationships, beliefs, and so forth. At the same time, the peoples of a subculture can and often do share many political and/or economic aspects, such as national identity, political organization, or institutions with those of the wider culture which encompasses them. Insofar as it is relevant to study individual subcultures due to their economic and political importance, they should be addressed in a micro list. For example, mainland Puerto Rican or Chicano subcultures in the United States will merit individual treatment in our context if there is a critical mass of actual or potential information systems activity. By the same token, though regarding a quite different issue, it will probably not make sense to isolate Navaho (U.S.) or Yanomamo (South America) Amerindian cultures until they are technologically significant in political and economic terms.

The second issue which we must address is the one of political states. One or more nationalities, each with characteristic cultural traits, may physically dwell in the same political state. In this case, depending on the economic and technological relevance, we may identify and label only the political state or the cultural area in a macro list. As need develops, a micro list will detail the individual cultures or subcultures within the area.

In addition, there are many cultures that cut across the boundaries of political geography. On the one hand, it might be sufficient to study a culture independently of political boundaries. An example is Basque culture. But in all probability the national state makes enough of a difference so that it might be more desirable to look at Spanish Basque and French Basque independently, or the first as a subculture within Spain and the second as a subculture within France.

It might then be more appropriate to develop a macro list which is political-state oriented, and as research is conducted that applies beyond the state addressed, it can be fitted within the larger cultural family. For example, any findings with respect to the French language should be applicable anywhere in the Francophone world. And observation on Spanish attitudes may well apply in many parts of Latin America. In another context, can we learn enough about Walloon and Flemish cultures in Belgium by studying the French and Dutch respectively? Obviously, it will be a function of the individual cultural variable being studied, and of the scope of the research.

So, let us consider the cultural areas themselves. Here we must be much more careful. It is precisely in the eyes of the people of a specific culture or cultural area that this investigation has meaning. It is for the French programmer, or the Mexican computer operator, or the Japanese DP manager, or the developers of technology desiring to better serve their users in these cultures, that this discussion makes sense. Therefore, we can adopt a macro list of cultural areas which might correspond to the principal geographic areas outside the United States with a considerable number of computers installed. However, since there are major cultures that cut across geographical zones, and by the same token, many cultures within the same set of national boundaries, it becomes difficult to elaborate this list. Nonetheless, if we allow the same approach and refine each major category as needed, we can start out with the following macro list:[1]

[1] We have focused on language for the development of this macro list because of its inportance as a cultural variable on information systems technology. In this vein linguistic groupings have been emphasized in our attempted categorization. However, other groupings have been created in a recognizably gross simplification. Presenting the Indian subcontinent, with its many different nations, its religious differences, and its myriad of languages as one cultural area is such a case. Likewise, speaking of Africa in terms of Anglophone, Francophone and Other is clearly imprecise from the sociological point of view. Nonetheless, because of the ability to expand each line item in the macrolist into as detailed a microlist as desired, and taking into account the present level of information systems usage, we have taken the liberty of presenting such groupings for this discussion. Beyond this we recognize the importance of non-linguistic dimensions of culture which are directly relevant to this technology.

- U.S./U.K./English Canadian/New Zealand/Australian
- French European
- Germanic European
- Scandinavian
- Ibero-American
- Italian
- Slavic
- Greek
- Turkish
- Arabic
- Israeli
- Persian
- Indian Subcontinent
- Indochinese/Thai/Burmese
- Korean
- Malay-Indonesian
- Japanese
- Chinese
- Anglophone African
- Other African
- Other

As needed, we might subdivide Ibero-American, as an example, as follows:

- Spanish
- Portuguese
- Brazilian
- Mexican
- Central American

- Ibero-American Caribbean
- Indoamerican
- Southern Cone (Argentina, Uraguay, Chile)

We recognize that there are both redundancies and omissions in our list. But this should not concern us now, for it will be the future researcher's job to define exactly the cultural area he is addressing or investigating, and possibly even provide insights for subcultures comprised in this work.

Thus, this taxonomical exercise is meaningless unless it will allow us to begin dealing with the realities of the Spanish-speaking Peruvian programmer in Lima, or the Nigerian Ibo computer maintenance engineer in Lagos, or the Paraguayan Guarani keypunch operator in Asuncion, or the Cantonese DP manager in Kwangchow.

By studying each cell in detail, a contribution should be made to understanding the interaction between our three dimensions. In the long run, we would hope that all cells be the object of sufficiently detailed study, and that the relevant theoretical problems be solved.

However, what is the more likely sequence of events that we foresee? Probably longitudinal studies will develop along the lines of culture, or of individual cultural variables. That is, the need to understand the impact of, say, Japanese culture on information systems technology will be looked at for each cultural variable and information systems components along the corresponding plane of our impact matrix. At present, we can only guide the researcher by indicating the possible implications of importance, which may lead to doing things differently as a result of further research. (See Table 1.)

Another probable approach is for the cross-cultural study of one variable (e.g., language) and understanding the patterns involved in the interaction between it and one information systems component (e.g., hardware). Thus, it will be looked at along one row, or group of rows, on our matrix. (See Table 2.)

	Hardware	Software	Procedures	Applications	Technical Skills	Education	Documentation	DP Organization	Communication Facility	Support Services	Management Skills	Standards
LANGUAGE	HI	HI	HI	HI	HI	HI	HI	?	HI	HI	HI	MED
VALUES	?	?	LO	HI	HI	HI	?	HI	MED	HI	HI	MED
BELIEFS	LO	LO	LO	MED	MED	MED	LO	MED	LO	MED	MED	?
ATTITUDES	MED	MED	HI	HI	HI	MED	LO	HI	HI	HI	HI	MED
EXPECTATIONS	HI	HI	HI	HI	HI	HI	LO	LO	HI	HI	HI	?
VITAL ASSUMPTION	?	?	?	MED	LO	MED	?	HI	?	MED	MED	LO
INTERP. RELATIONSHIPS	?	LO	MED	HI	HI	HI	?	HI	HI	HI	HI	LO
MOTIVATORS	?	?	?	HI	HI	MED	?	HI	MED	HI	HI	?
STATUS	MED	MED	?	MED	HI	HI	?	HI	MED	HI	HI	?
SOCIAL STRUCTURE	?	?	?	HI	HI	HI	?	HI	HI	HI	HI	MED
SOCIAL INST.	?	?	?	MED	MED	HI	LO	HI	HI	HI	HI	MED
SOCIAL MOBILITY	?	?	?	HI	HI	HI	MED	HI	HI	HI	HI	MED
EDUCATION	?	LO	LO	HI	HI	HI	HI	HI	MED	MED	HI	HI
MANAGEMENT STYLE	LO	MED	MED	HI	HI	HI	HI	HI	MED	HI	HI	HI

TABLE 1.

POSSIBLE IMPACT RELEVANCE OF INFORMATION SYSTEMS COMPONENTS

ON CULTURAL VARIABLE INTERACTION FOR JAPAN

Culture	Information Systems Component
	HARDWARE
U.S./U.K./ English Canadian/Australian/New Zealand	LO
French European	MED
Germanic European	MED
Scandinavian	MED
Italian	MED
Slavic	HI
Greek	HI
Turkish	MED
Ibero-American	MED
Arabic	HI
Israeli	MED
Persian	HI
Indian Subcontinent	HI
Indochinese/Thai/Burmese	V.HI
Korean	V.HI
Malay-Indonesian	HI
Japanese	V.HI
Chinese	V.HI
Anglophone African	LO
Francophone African	MED
Other African	?
Other	?

TABLE 2.

ESTIMATED IMPACT RELEVANCE OF CULTURE ON INFORMATION SYSTEMS HARDWARE

III. SOURCES OF INFORMATION

Extensive work has already been done on the social implications of technology in general and of computer technology in particular; that is, on how information systems technology impacts society. The somewhat parallel but inverse issue of how a culture impacts the utilization and application of a technology has not been dealt with to any significant degree. If we take computer technology specifically, the sources are practically non-existent, with the exception of work done in the area of languages.

Overall, the management style literature provides the elementary background on culture necessary to follow this work. Webber's Culture and Management[1a] and Rhinesmith's Cultural-Organizational Analysis[2] are important in this context, as were Richman[3] and Kluckhohn.[4] From these sources our tentative macro list for cultural variables, as well as some of the basic definitions, were developed.

In dealing specifically with information processing technology and culture, or individual cultural variables, we must mention the work done

[1a] R. A. Webber, Culture and Management, Richard D. Irwin, Inc., Homewood, Illinois. 1969, 598 pp.

[2] S. H. Rhinesmith, Cultural-Organization Analysis, McBer and Co. 1971, 53 pp.

[3] B. M. Richman, "Significance of Cultural Variables, "Academy of Management Journal, Vol. 8, No. 4 (December 1965).

[4] C. Kluckhohn, "Cultural Behavior," in Handbook of Social Psychology, G. Lindzey (ed.), 1st Ed., Vol. ii, Addison-Wesley, Reading, Massachusetts, 1954, pp. 921-976.

by the ILO (International Labor Office) in Geneva.[5] As a part of a general research project on the manpower problems associated with the introduction of automation and advanced technology in developing countries, they conducted case studies of EDP installations in Ethiopia, Brazil, East Pakistan (now Bangladesh) and India. These, together with other automation case studies in Columbia and Tanzania which were part of the same research project, provide valuable insights in resistance to change and other attitudinal factors and processes affecting the utilization of information systems technology in these cultural areas. The object of the research was to study the impact of the technology on employment, and therefore the work is only tangentially useful for our purpose.

Of course, the technical literature and the DP industry journals abound in descriptive reviews of the "state of the arts" in many countries. Western Europe and Japan are certainly major markets for, and users and developers of, information systems technology. Most of the articles in the literature, however, deal with economic and/or marketing aspects of the DP industry rather than with any social-psychological issues. Nonetheless, there are some insights to be gained, especially for any longitudinal study, by reviewing that body of literature.

The next section is totally dedicated to language as a cultural variable. However, some of the most interesting work done on "native language processing" comes from cultures with the most difficulties in handling their language and script with the current DP hardware. Several

[5] The case studies took place in the 1970 timeframe and were compiled in: Automation in Developing Countries, International Labor Organization, Geneva, Switzerland, 1972, 246 pp.

sources have dealt with this problem in Japan and China;[6] Parhami and Mavaddat[7] treat the issue for Iran; and Vikas[8] has compiled an excellent bibliography on non-English language computer issues concentrating on Indian languages.

[6] Various institutions have worked on this problem intensely in Japan and the Republic of China (Taiwan). It is believed that research has also been done in the Peoples Republic of China but not much information is available on this. The International Computer Symposium 1977, held at the National Taiwan University in Taipei, dedicated several sessions to this issue.

[7] See B. Parhami and F. Mavaddat, "Computers and the Farsi Language--A Survey of Problem Areas," 1977 IFIP Congress Proceedings, North Holland Publishing Company, 1977, pp. 673-676.

[8] See Om Vikas, "Use of Non-English Language in Computers--A Selected Bibliography," Indian Electronics Commission, June 1978, 69 pp. This work was done in preparation for the Symposium on Linguistic Implications of Computer-Based Information Systems, New Dehli, India, November 10-12, 1978.

IV. LANGUAGE

Language is the organized body of speech or phonetical utterings through which people communicate ideas, emotions and feelings. All human languages are spoken. Not all languages have a written form, however, and some languages can be written using more than one script. Turkish, for example, was written using the Arabic alphabet until the Roman alphabet was adopted in the wake of the Europeanization of the country that followed the Turkish Revolution. Serb and Croatian are the same language but the first is written with the Cyrillic alphabet and the latter with the Roman.

Since language is a key cultural variable we must grant it special attention. There are three aspects of language which we must consider in our analysis. In one sense, we can address these using our impact matrix concept; first, cultural variable "language" as it intersects information systems components--"technical skills, education and documentation;" second, cultural variable "language" as it intersects component "hardware;" and third, "language" as it intersects "computer programming."

The first problem is one in which the information systems field shares with many other disciplines today. English has become a "lingua franca" for science and technology and the non-English speaking technologist is at a disadvantage almost to the point of exclusion from the field if he cannot at least read English. The principal textbooks, journals, and manuals are sometimes translated into other major languages (i.e., French, German, Russian, Japanese, Italian, Spanish, Portuguese), but this implies a time lag frequently unacceptable to the scientist or technologist. In addition, the subtleties of individual cultures, and of languages in particular, often make translation very difficult or simply awkward leading to failure in its main objective which is communication. Dwelling on these issues will allow us to share Schramm's concern on the need for the "building of bridges."[9]

Language, then, is an essential factor in learning the skills related to any technology--and thus critical in the technology transfer process. But language is especially important in learning the particular skills associated with information systems, such as how to operate a computer or code in a particular programming language. Obviously, the central operator of a modern computer must be able to exchange communications through a central console with the system control programs. The messages and diagnostics put out by the system are in English or English-like code, and the need for real time action on each command ill affords doubts or inquiries relating to the semantics of an English statement.

Information systems education as it relates to the teaching and learning of technical skills used in working with computers is clearly impacted by language.[10]

[9] These problems, which go beyond simple word gaffes, are rooted in the need to be thoroughly imbued in a culture before true translation can take place. Wilbur Schramm speaks of people who can act as bridges between cultures. See Schramm, W., "A Note on the Building of Bridges," in Communication Across Cultures, For What? (J. C. Condon and M. Saito, eds.), the Simul Press, Tokyo, 1976, pp. 7-19.

A realistic account of the problems involved in technical translation can be read in a 1/13/77 Wall Street Journal report by G. Christian Hall headlined "More Firms Turn to Translation Experts to Avoid Costly, Embarrassing Mistakes."

[10] This is particularly true for computer systesm designed to support decisions in complex and semi-structured problem environments. See Meador, C.L. and D. N. Ness,"Decision Support Systems: An Application to Corporate Planning," Sloan Management Review, Vol. 15, No. 2, Winter 1979, pp. 51-68.

Some insights on this problem in Latin America are found in Barquin.[11]

Non-English speaking cultures have had varying degrees of difficulty in applying computer technology. The problems go from the utilization of English in computer-related input/output operations to the development of necessary I/O hardware and software to handle their own language. Most major cultures have now been able to develop some forms of adaptation, but these have been for the most part sub-optimal.

Russians can produce Cyrillic printouts and Israelis can output reports in Hebrew. Present day information systems technology, however, is still a long way from providing a full capability for handling non-English languages. Furthermore, many non-linquistic dimensions of culture have been totally ignored.

By looking at those aspects of written language important in the design and utilization of information systems technology, we hope to take a first step toward an integral approach to the universal problem of language as a cultural variable and computer hardware and software. For this we will now analyze the Roman Alphabet, the English subset of the Roman alphabet, non-Roman alphabets, non-alphabetic languages, and the issue of read/write direction.

[11] There are two relevant documents:

1. R. C. Barquin, The Degree of Penetration of Computer Technology in Latin America: A Survey, MIT Sloan School of Management Working Paper No. 702-74, MIT, April 1974.

2. R. C. Barquin, "On Computer Software, Education, and Personnel in Developing Countries," ITCC Review, Vol. V, No. 16, International Technical Cooperation Center, Tel Aviv, Israel, January 1976, pp. 11-22.

IV-1 Roman Alphabet

The principal features demanded by a culture from its information-processing hardware is the capability to handle the written forms of its vernacular language. Since computer technology has been principally developed in the United States, the original printing devices provided a character set suited for outputting English text; that is, the 26-letter subset of the Roman alphabet and necessary punctuation marks normally utilized in writing English (which we will call the E-set). Likewise, the code structures mainly devised for data input from the earliest times took only the E-set into account. Thus the Hollerith Code provided for the neat mapping of letters into punches in a card.

A	12-1	J	11-1	S	0-2
B	12-2	K	11-2	T	0-3
C	12-3	L	11-3	U	0-4
D	12-4	M	11-4	V	0-5
E	12-5	N	11-5	W	0-6
F	12-6	O	11-6	X	0-7
G	12-7	P	11-7	Y	0-8
H	12-8	Q	11-8	Z	0-9
I	12-9	R	11-9		

This would seem to be adequate--and in many ways there was no choice--for any of the other languages written with the Roman alphabet, such as Spanish, Portuguese, French, Italian, German, and so on. Some of these, however, utilize a superset which includes additional letters, a number of which feature diacritical marks. In Spanish, for example, "ch," "ll," and "ñ" are individual letters of the alphabet. The first two can be handled fairly well by any code devised for English, but the "ñ" necessitates special treatment since substitution of the letter "n" is often not acceptable.[12] Of course, sorting these would imply modifying any software written for the E-set specifically.

A series of diacritical marks also become essential for proper communication in other Romance languages; for example, the circumflex (i.e., côte) in French, or the cedilla in Portuguese (i.e., conceiçao) and French (i.e., garçon). Miller[13] lists a number of these marks necessary to write foreign words in English text.

Umlaut	ü
Circumflex	ô
Tilde	ã
Grave	è
Acute	é
Macron	ā
Cedilla	ç

Solutions to these problems range from straightforward to extremely difficult. For example, the "ñ" was easily introduced quite some time ago as a standard feature in print chains, trains, drums, bars and typing elements for Spanish-speaking countries. By and large the limitations to solving any of these problems are economic in nature rather than technological.

[12] While the meaning is transmitted in writing, say, "senor" for "señor," or "manana" for "mañana," it is not always true. In one case, the frequently used word,"año," which means year, changes its meaning to anus if the tilde is dropped.

[13] See I. Miller, Text Evaluation of World Languages, IBM Technical Report TR 00-2561, Poughkeepsie Laboratory, September 24, 1974, p. 1.

Another issue present here is the actual engineering of the print element. Most often the positioning of a letter in the type bar or drum, and certainly the redundant occurrences of that letter in the element, are a function of its frequency of use. In English, we know that the relative frequency of occurrence of the letter e is 0.131; of the letter t, 0.105; and so on down to the letter z, which is 0.00077.[14]

But the same does not necessarily hold true for other languages. The statistical structure of French, Spanish, German, and so forth, differs from the English and also from each other. In effect, the frequency of occurrence of e in Spanish is 0.113, and of t is 0.036. We can thus expect some possible deterioration of performance in any printing device which has been engineered for one language and is used for printing another. And, of course, we must also alter the relevant software to optimize the handling of each condition. Naturally, we should optimize hardware performance by designing print elements around character frequency of occurrence for the target culture's language. Of course, all that has been said for printing devices also applies for character recognition input devices.

IV-2 Non-Roman Alphabet

As we look at cultures whose languages are written in non-Roman alphabets, the problems grow exponentially from the point of view of established information processing technology, especially the hardware enabling us to input and output data in formats which had not been taken into account in the initial equipment design. Of course, a frequent

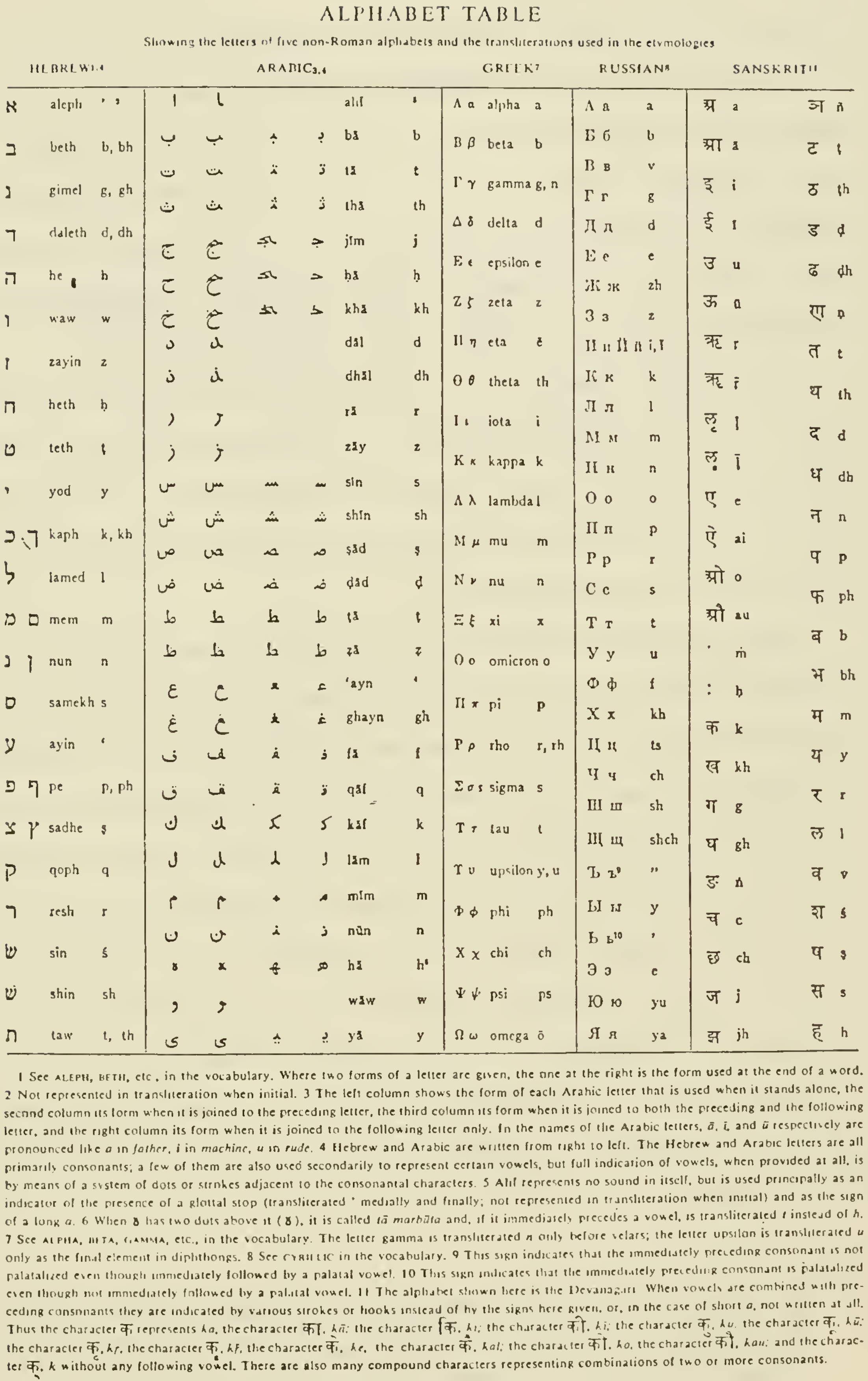
[14] This means that the letter e will occur on the average 131 times, in every 1000 letters of English text; the letter t, 105 times; and the letter z, 0.77 times. For a more detailed explanation, see: M. Schwarz, Information Transmission, Modulation and Noise, McGraw-Hill, New York, 1959, p. 14.

approach is Romanization; that is, the use of Roman letters in phonetic reconstruction of speech normally written with another alphabet or script. This often occurs where there is no capability for handling the non-Roman script or for non-alphabetic writing, and would be a logical route to take for the processing of information from non-written languages.[15] What happens, though, when we take the generic approach of adapting the technology to handle a culture's language?

Let us take Greek as an example and see what the impact is on data processing as we know it. The Greek alphabet consists of 24 letters. About 9 of these are identical (in upper case) both in format and in function to Roman letters: A, B, E, I, K, M, N, O, and T. Given the historical nexes between the Greek and Roman alphabet, we would expect there to be some similarities. Obviously, a slight change to the Hollerith Code would be a feasible solution for most data entry, and an appropriate modification to the input/output software routines should enable us to handle the problem easily.

The printing hardware itself, whether a typebar, a ball element, a drum, or a chain, will now have to feature the 24 letters of the Greek alphabet plus numerals and punctuation. This is not much of a problem, since Greek printing elements can be easily adapted to most present DP hardware. However, it is expected that the information content of the Greek language and its statistical structure should be taken into account when engineering the hardware adaptations and modifying the necessary software.

[15] This has happened in the case of almost all American Indian languages. Moreover, Romanized Bibles have been printed in hundreds of languages with no previous written form.



ALPHABET TABLE

Showing the letters of five non-Roman alphabets and the transliterations used in the etymologies

HEBREW[1,4]		
א	aleph	ʼ[2]
ב	beth	b, bh
ג	gimel	g, gh
ד	daleth	d, dh
ה	he	h
ו	waw	w
ז	zayin	z
ח	heth	ḥ
ט	teth	ṭ
י	yod	y
כ ך	kaph	k, kh
ל	lamed	l
מ ם	mem	m
נ ן	nun	n
ס	samekh	s
ע	ayin	ʻ
פ ף	pe	p, ph
צ ץ	sadhe	ṣ
ק	qoph	q
ר	resh	r
שׂ	sin	ś
שׁ	shin	sh
ת	taw	t, th

ARABIC[3,4]					
ا	ـا			alif	[5]
ب	ـب	ـبـ	بـ	bā	b
ت	ـت	ـتـ	تـ	tā	t
ث	ـث	ـثـ	ثـ	thā	th
ج	ـج	ـجـ	جـ	jīm	j
ح	ـح	ـحـ	حـ	ḥā	ḥ
خ	ـخ	ـخـ	خـ	khā	kh
د	ـد			dāl	d
ذ	ـذ			dhāl	dh
ر	ـر			rā	r
ز	ـز			zāy	z
س	ـس	ـسـ	سـ	sīn	s
ش	ـش	ـشـ	شـ	shīn	sh
ص	ـص	ـصـ	صـ	ṣād	ṣ
ض	ـض	ـضـ	ضـ	ḍād	ḍ
ط	ـط	ـطـ	طـ	ṭā	ṭ
ظ	ـظ	ـظـ	ظـ	ẓā	ẓ
ع	ـع	ـعـ	عـ	ʻayn	ʻ
غ	ـغ	ـغـ	غـ	ghayn	gh
ف	ـف	ـفـ	فـ	fā	f
ق	ـق	ـقـ	قـ	qāf	q
ك	ـك	ـكـ	كـ	kāf	k
ل	ـل	ـلـ	لـ	lām	l
م	ـم	ـمـ	مـ	mīm	m
ن	ـن	ـنـ	نـ	nūn	n
ه	ـه	ـهـ	هـ	hā	h[6]
و	ـو			wāw	w
ى	ـى	ـيـ	يـ	yā	y

GREEK[7]		
Α α	alpha	a
Β β	beta	b
Γ γ	gamma	g, n
Δ δ	delta	d
Ε ε	epsilon	e
Ζ ζ	zeta	z
Η η	eta	ē
Θ θ	theta	th
Ι ι	iota	i
Κ κ	kappa	k
Λ λ	lambda	l
Μ μ	mu	m
Ν ν	nu	n
Ξ ξ	xi	x
Ο ο	omicron	o
Π π	pi	p
Ρ ρ	rho	r, rh
Σ σ ς	sigma	s
Τ τ	tau	t
Υ υ	upsilon	y, u
Φ φ	phi	ph
Χ χ	chi	ch
Ψ ψ	psi	ps
Ω ω	omega	ō

RUSSIAN[8]	
А а	a
Б б	b
В в	v
Г г	g
Д д	d
Е е	e
Ж ж	zh
З з	z
И и Й й	i, ĭ
К к	k
Л л	l
М м	m
Н н	n
О о	o
П п	p
Р р	r
С с	s
Т т	t
У у	u
Ф ф	f
Х х	kh
Ц ц	ts
Ч ч	ch
Ш ш	sh
Щ щ	shch
Ъ ъ[9]	″
Ы ы	y
Ь ь[10]	′
Э э	e
Ю ю	yu
Я я	ya

SANSKRIT[11]			
अ	a	ञ	ñ
आ	ā	ट	ṭ
इ	i	ठ	ṭh
ई	ī	ड	ḍ
उ	u	ढ	ḍh
ऊ	ū	ण	ṇ
ऋ	ṛ	त	t
ॠ	ṝ	थ	th
ऌ	ḷ	द	d
ॡ	ḹ	ध	dh
ए	e	न	n
ऐ	ai	प	p
ओ	o	फ	ph
औ	au	ब	b
ं	ṁ	भ	bh
ः	ḥ	म	m
क	k	य	y
ख	kh	र	r
ग	g	ल	l
घ	gh	व	v
ङ	ṅ	श	ś
च	c	ष	ṣ
छ	ch	स	s
ज	j	ह	h
झ	jh		

[1] See ALEPH, BETH, etc., in the vocabulary. Where two forms of a letter are given, the one at the right is the form used at the end of a word. [2] Not represented in transliteration when initial. [3] The left column shows the form of each Arabic letter that is used when it stands alone, the second column its form when it is joined to the preceding letter, the third column its form when it is joined to both the preceding and the following letter, and the right column its form when it is joined to the following letter only. In the names of the Arabic letters, *ā*, *ī*, and *ū* respectively are pronounced like *a* in *father*, *i* in *machine*, *u* in *rude*. [4] Hebrew and Arabic are written from right to left. The Hebrew and Arabic letters are all primarily consonants; a few of them are also used secondarily to represent certain vowels, but full indication of vowels, when provided at all, is by means of a system of dots or strokes adjacent to the consonantal characters. [5] Alif represents no sound in itself, but is used principally as an indicator of the presence of a glottal stop (transliterated ʼ medially and finally; not represented in transliteration when initial) and as the sign of a long *a*. [6] When ه has two dots above it (ة), it is called *tā marbūṭa* and, if it immediately precedes a vowel, is transliterated *t* instead of *h*. [7] See ALPHA, BETA, GAMMA, etc., in the vocabulary. The letter gamma is transliterated *n* only before velars; the letter upsilon is transliterated *u* only as the final element in diphthongs. [8] See CYRILLIC in the vocabulary. [9] This sign indicates that the immediately preceding consonant is not palatalized even though immediately followed by a palatal vowel. [10] This sign indicates that the immediately preceding consonant is palatalized even though not immediately followed by a palatal vowel. [11] The alphabet shown here is the Devanagari. When vowels are combined with preceding consonants they are indicated by various strokes or hooks instead of by the signs here given, or, in the case of short *a*, not written at all. Thus the character क represents *ka*, the character का, *kā*; the character कि, *ki*; the character की, *kī*; the character कु, *ku*; the character कू, *kū*; the character कृ, *kṛ*; the character कॄ, *kṝ*; the character के, *ke*; the character कै, *kai*; the character को, *ko*; the character कौ, *kau*; and the character क्, *k* without any following vowel. There are also many compound characters representing combinations of two or more consonants.

*Source: Webster's New Collegiate Dictionary, Merriam-Webster Co., Springfield, Mass., p. 33.

TABLE 3

Because of the historical development of the technology and the predominance of English as the technical language there are strong reasons for having a Roman alphabet capability in one's information processing system. This is true in almost all cultural areas. Thus the card punches assigned will generally not overlap those in the standard Hollerith code reserved for the E-set. Rather a new assignment is made and usually suboptimal solutions are implemented. Let us look at another case: the Cyrillic.

The Cyrillic alphabet, to a large degree derived from the Greek and Roman, is used in writing many Slavic languages such as Bulgar, Russian, Serb, and Ukrainian.[16] While there are some variations of this alphabet, similar to the different subsets of the Roman discussed previously, the Russians utilize a 31-letter set.

Many of the points brought out in our Greek example also apply here; and we can see that the Hollerith Code, to be applicable, would necessitate the utilization of some unassigned combination of punches. That is probably a better solution than attempting to substitute each Cyrillic letter for its nearest Roman counterpart in the code, since there would be five excess characters in need of special treatment, and logic would suggest some sequential order. Nevertheless, the problem is not a complex one from the point of view of technical feasibility. The implemented solution here has generated the seemingly illogical pattern of Table 4 .

Because the Cyrillic has more letters than the Roman, all subroutines handling I/O must take into account the expansion of tables to accommodate their new dimension.

[16] An interesting theory maintains that the alphabet folows religion. This seems to hold true for the Slavic languages since the Orthodox nations adopted Cyrillic while the Catholic (Poland, Czechoslovakia, Lithuania, Croatia, et cetera) adopted the Roman. As was mentioned, Serb and Croatian are the same language but Serb is written with Cyrillic letters while Croatian uses the Roamn alphabet. Pointedly, the same holds true for Hindi and Urdu. Urdu, the official language of Moslem Pakistan, is written in Arabic script, while Hindi is written mainly in Devanagari script.

CYRILLIC ALPHABET

CARD CODE ASSIGNMENTS

LETTER	CARD PUNCHES
А	12-11-0-9
Б	12-11-0-8-2
В	11-0-9-8-5
Г	12-11-0-8-7
Д	12-11-0-8-4
Е	12-11-0-8-5
Ж	11-0-9-8-4
З	12-11-0-9-8-2
И	12-0-9-8-3
Й	12-0-9-8-4
К	12-0-9-8-5
Л	12-0-9-8-6
М	12-0-9-8-7
Н	12-11-9-8-2
О	12-11-9-8-3
П	12-11-9-8-4
Р	12-11-9-8-6
С	12-11-9-8-7
Т	11-0-9-8-2
У	11-0-9-8-3
Ф	12-11-0-8-6
Х	12-0-9-8-2
Ц	12-11-0-8-3
Ч	12-11-0-9-8-6
Ш	12-11-0-9-8-3
Щ	12-11-0-9-8-5
Ы	11-0-9-8-7
Ь	11-0-9-8-6
Э	12-11-0-9-8-4
Ю	12-11-0-8
Я	12-11-9-8-5

TABLE 4

In addition, of course, the keyboard layout must comply with a Cyrillic only, or a Roman/Cyrillic data-entry mode.

Given the present computer architectures, working with an 8-bit byte as the basic processing unit, we are still comfortable in that we have up to 256 different representations possible and EBCDIC has provided an acceptable vehicle for solving most of our data processing necessities. EBCDIC is a representation scheme in common use in computers and communication systems. It stands for the 8-bit Extended Binary Coded Decimal Interchange Code. Two other representation schemes, ASCII (American Standard Code for Information Interchange) and ISCII (International Standard Code for Information Interchange), contain seven bits of information and thus can handle only 128 different character representations.

Some alphabets, however, are deceptive in that the number of letters is not directly indicative of the cardinality of their correspondent character set. The Arabic alphabet, for example, has only 28 letters, but each one has four forms depending on its position within a word: isolated, beginning, middle, and ending. Thus the issue of ligatures generates a 112-character set necessary to handle Arabic.[17] In addition, although we refer to our number system as Arabic, and it was undoubtedly derived from it, the format of the characters themselves differ substantially.

Parhami and Mavaddat[18] illustrate the problems referred to in their insightful treatment of Farsi (Persian), which utilizes a superset of the Arabic alphabet.

[17] There is the additional problem of varying heights and widths in printed and handwritten texts. Though some standardization has been necessary for commercial printing and business text, these are seen as having a negative impact on the esthetics of calligraphy.

[18] See Parhami and Mavaddat, op cit, p. 673.

The ligatures in writing Sanskrit or Hindi (Devanagari script)[19] take what is basically a 48-letter set to approximately 400 characters. Of course, we now see that 256 possible combinations provided by the 8-bit byte no longer suffice to handle the Hindi language written using the Devanagari alphabet. If we add the needs for numerals and punctuation as well as the necessary reserved combinations for control characters and the like, we realize the sub-optimality of our architecture and code scheme to handle the processing of information in some languages.

IV-3 Other Alphabetic Issues

There are many more alphabets. Some are handled easily by our present schemes and some are not. Two more languages should be mentioned here in terms of their implications for the processing of information: Korean and Japanese.

Korean is written basically using the Han'gul alphabet, consisting of 24 letters.[20] The one characteristic of Korean script that we want to focus on here is the position of characters in the structure of a symbol or

[19] . Devanagari, which means "divine script" in Sanskrit (classical language of India and Hinduism), is its principal written form. It has many syllabaric characteristics since when vowels are combined with preceding consonants, they are indicated by strokes or hooks rather than by their independent signs. The Hindi language, strongly influenced by Sanskrit, is also written using the Devanagari script.

[20] . Developed in the fifteenth century, Han'gul (or Hankul) originally had 25 letters. It consists presently of 14 vowels and 10 consonants, which are written in clusters by syllables.

word. For example, a Korean symbol can be created from components according to a number of different schemes. Some of these are illustrated in Figure 3. In a way it is similar to a cluster grouping of our own letters to spell, say, "call."

C	A
L	L

Thus, in addition to the issues already raised concerning statistical structure of languages, cardinality of alphabets, adequacy of codes, and so forth, we also have the problem of letter positioning within a symbol or word. This implies, at the input level, the development of an unambiguous algorithm for data entry and internal processing and, at the output level, some special typing and/or printing controls related at least to carriage movement, line density, and print redundancies to achieve acceptable output. Again, it is feasible--printing processes and hardware features have been developed; but it adds complexity to the whole operation.

IV-4 Alphabetic and Non-Alphabetic Combinations

Koreans also use many Chinese symbols in their writing. These are not alphabetic constructs but rather ideographs with full individual meaning. This mixture of both alphabetic and non-alphabetic script creates further difficulties for any automated handling of written information. Rather than treat this issue based on the Korean example, let us talk about Japan.

Japanese is written using a combination of three different elements:

SOME FORMS OF KOREAN SYMBOL GENERATION

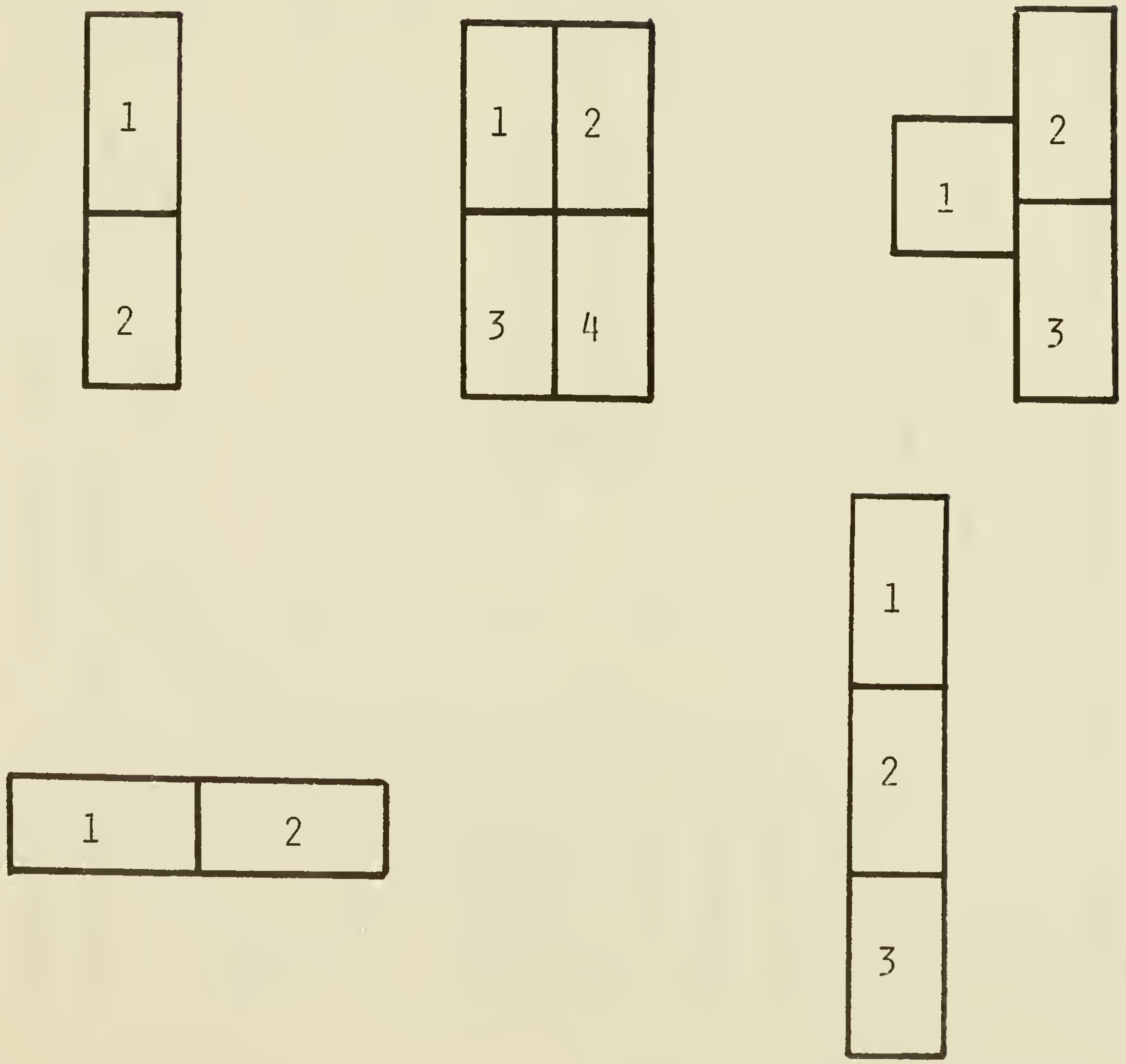


FIGURE 3

kanjis (Chinese symbols as in Figure 4) and katakana or hiragana characters[21] (see Figure 5). The katakana and hiragana jointly are called the kana and constitute a syllabary, or alphabet whose symbols phonetically represent syllables. The kana has 48 elements (which expand to 73 with diacritical marks) and they are used for certain phonetic construction of words in Japanese writing. Traditionally the kana are used for: foreign words, grammatical inflections, functional words not represented by kanjis, or to indicate pronunciation next to a kanji.

The kanjis represent Chinese words or morphemes with their own sound and meaning. Because Japanese is a completely different language from chinese, however, the kanjis are a source of ambiguity since each one has a "kun" and an "on." The "kun" of a kanji is a Japanese word that has the same meaning as that of the kanji. At the same time, each kanji also has an "on," which is the Japanese vocalization of the Chinese sound for the kanji. The Japanese will thus write words, then, by either using a kanji directly to represent its actual meaning or by constructing the Japanese word phonetically using the "on" of each kanji.

As can be imagined, this is a source of ambiguity and thus complexity in terms of information processing, though it is a spring of much artistic beauty and inspiration through the incessant interplay of multiple meanings within the same scripture.

Because of the difficulty in the automated processing of Japanese script in its usual form, katakana has been utilized extensively in a non-traditional manner by the syllabaric construction of words which would

[21] Katakana and hiragana are composed of different but equivalent characters. Hiragana characters are cursive whereas katakana symbols are more angular and square. They were developed in the 8th and 9th centuries.

SOME EXAMPLES OF CHINESE SYMBOLS

田
(FIELD)

水
(WATER)

火
(FIRE)

石
(STONE)

星
(STAR)

山
(MOUNTAIN)

FIGURE 4

KANJI IS FUNDAMENTAL TO INTERACTIVE DATA PROCESSING IN JAPAN

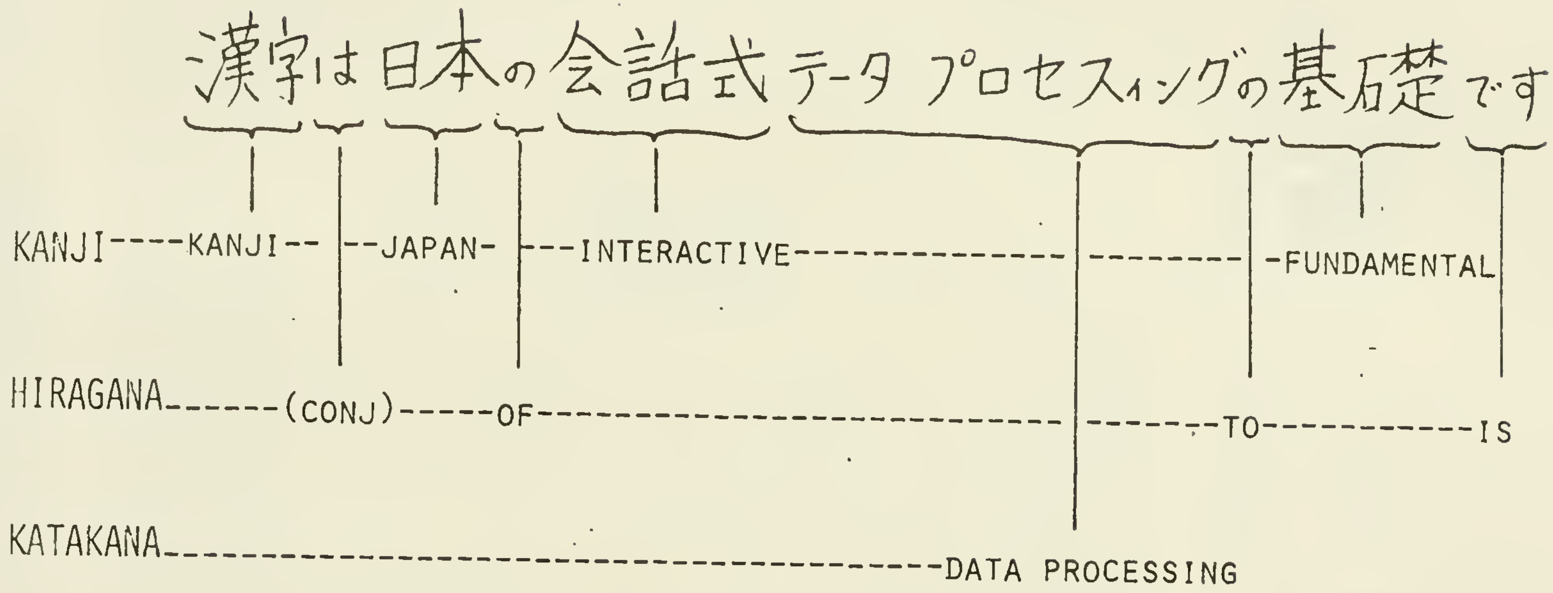

漢字は日本の会話式データプロセスィングの基礎です
KANJI----KANJI-- --JAPAN- ---INTERACTIVE-------------------------- -FUNDAMENTAL
HIRAGANA------(CONJ)-----OF-------------------------------------TO-----------IS
KATAKANA---------------------------------------DATA PROCESSING

FIGURE 5

normally be written with kanjis. While this approach has allowed at least a partial solution to some technical DP problems, the society still does not find it completely acceptable given the deep roots of symbolic scripture through kanjis. These are greatly preferred, almost exclusively, for proper names, addresses, and many abstract concepts easily grasped through well-known characters but difficult to conceptualize through syllabic agglutination.

IV-5 Non-Alphabetic Script

The discussion of Japanese has led us into the subject of non-alphabetic writing. The principal modern languages utilizing ideographs in their traditional written forms are the Chinese languages and dialects (although Chinese is best characterized as logographic--the use of symbols representing entire words). These are Mandarin, Cantonese, Hakka, Wu, Min and others. In addition to Japanese and Korean, other cultures which were strongly influenced by China utilize these symbols at least partially in their own forms of writing. The Chinese characters play a role similar to Latin and Greek roots in our own language.

Languages usually written with non-alphabetic scripts are important for various reasons, but most telling because well over one billion people communicate in them (see Table 5). From the point of view of information systems, they are interesting and challenging because they are not efficiently handled through the technology as it has been developed to date.

One basic problem they present is the vast number of different symbols. Estimates for the cardinality of the character set for Chinese script are in tens of thousands; 40,000 characters or so are included in most Chinese dictionaries, 10,000 are in use for telegraphic purposes and about 2,000

LANGUAGES OF THE WORLD SPOKEN BY AT LEAST 100 MILLION PERSONS

(Midyear 1977)

LANGUAGE	MILLIONS
Mandarin Chinese	670
English	369
Russian	246
Spanish	225
Hindi*	218
Arabic	134
Portuguese	133
Bengali	131
Japanese	113
Malay-Indonesian	101

*Hindi (official language of India) and Urdu (official language of Pakistan) are essentially the same language; but Hindi is written in Devanagari script and Urdu in Arabic script.

SOURCE: Sidney S. Culbert, Assoc. Prof. of Psychology, University of Washington from The World Almanac and Book of Facts 1978 New York, 1978, p. 186.

TABLE 5

are needed for minimum literacy requirements. Even for what is considered average literacy one need talk about several thousand. This implies, on the one hand, that we have informational difficulties due to data entry coding considerations, internal processing problems stemming from the constraints imposed by the 8-bit byte, and output complexities for effective and efficient printing. This also implies, of course, the necessary software to handle the hardware developed.

Many approaches have been followed by Chinese and Japanese to attack the problem, none totally satisfactory to date. They range from stored "symbol dictionaries" to algorithmic techniques based on the radical components of each symbol.[22]

But one inescapable fact in dealing with non-alphabetic writing is that orderings become much more complex and difficult. Alphabetic sequencing is one of the foundations of information processing and by definition no such possibility exists with non-alphabetic script. Thus, one of the great capabilities of the computer, its ability to sequence,

[22] Chinese character processing is so difficult that telegraphic transmission is still done by coding each symbol into a four digit number. Typewriters are very bulky and achieve a maximum speed of about 10 characters per minute. General attempts have been made to develop phonetic schemes for writing Chinese. A Kana-inspired national phohetic system was developed in 1919 during the First Republic. Although it never became popular, there is some use of this system still today in Taiwan. Romanization was widely used by English-speaking missionaries, and various systems were developed along this line, one of the most frequently used being the Wade-Giles system. Among the systems developed in this century were: the 1929 National Romanization project in which the famous author Lin Yu-Tang was involved, and the 1930 Communist experiment, Latin Xua. Since 1958, the People's Republic of China has been trying to popularize Pin-Yin, a new romanization system.

A prototype drum printing device seems to have been developed by two Englishmen capable of handling 4,356 characters. (See "Two Britons Devise a Computer That Can Communicate in Chinese," by R. W. Apple, Jr., in the New York Times, January 25, 1978.)

In Japan several DP manufacturers also offer various degrees of Kanji processing capability.

arrange, and search for information according to an alphabetic code is lost. (So important is this feature that in French and Spanish, computers are called ordinateurs/ordenadores respectively due to their ability to place elements in order.) Sorting thus becomes a major issue in the handling of information written in Chinese characters. Of course, Chinese symbols are ordered according to radical composition, to the number of strokes necessary to write them and the calligraphic sequence of said strokes. But this lacks many of the analytical advantages of alphabetic ordering.[23]

IV-6 Read/Write Direction

The last item we would like to touch upon here deals with the direction in which a language is read and/or written and the implications this has for the automated processing of information.

English, as well as all modern European languages, is normally read and written from left-to-right, character following character, along the same horizontal line. Lines follow each other vertically, from the top to the bottom of each page or facsimile. But not all languages are read/written in the same manner. Arabic and Hebrew, for example, are both read and written from right-to-left. (See Figure 6.)

What are the implications involved here for DP as we know it? Data entry (keypunch/verification) hardware must, of course, take this into account. Given that humans do not easily read/write backwards, the direction of insertion of the medium to be coded and/or the actual physical coding must be reversed. Similar modifications must be made in any I/O

[23] See Japanese Industrial Standard (JIS) code of the Japanese Graphic Character Set for Information Exchange (JIS-C-6226-1978)m 1/1/78.

THE PRINCIPAL WAYS TO READ/WRITE ONE PAGE OF TEXT IN SOME MAJOR CULTURES

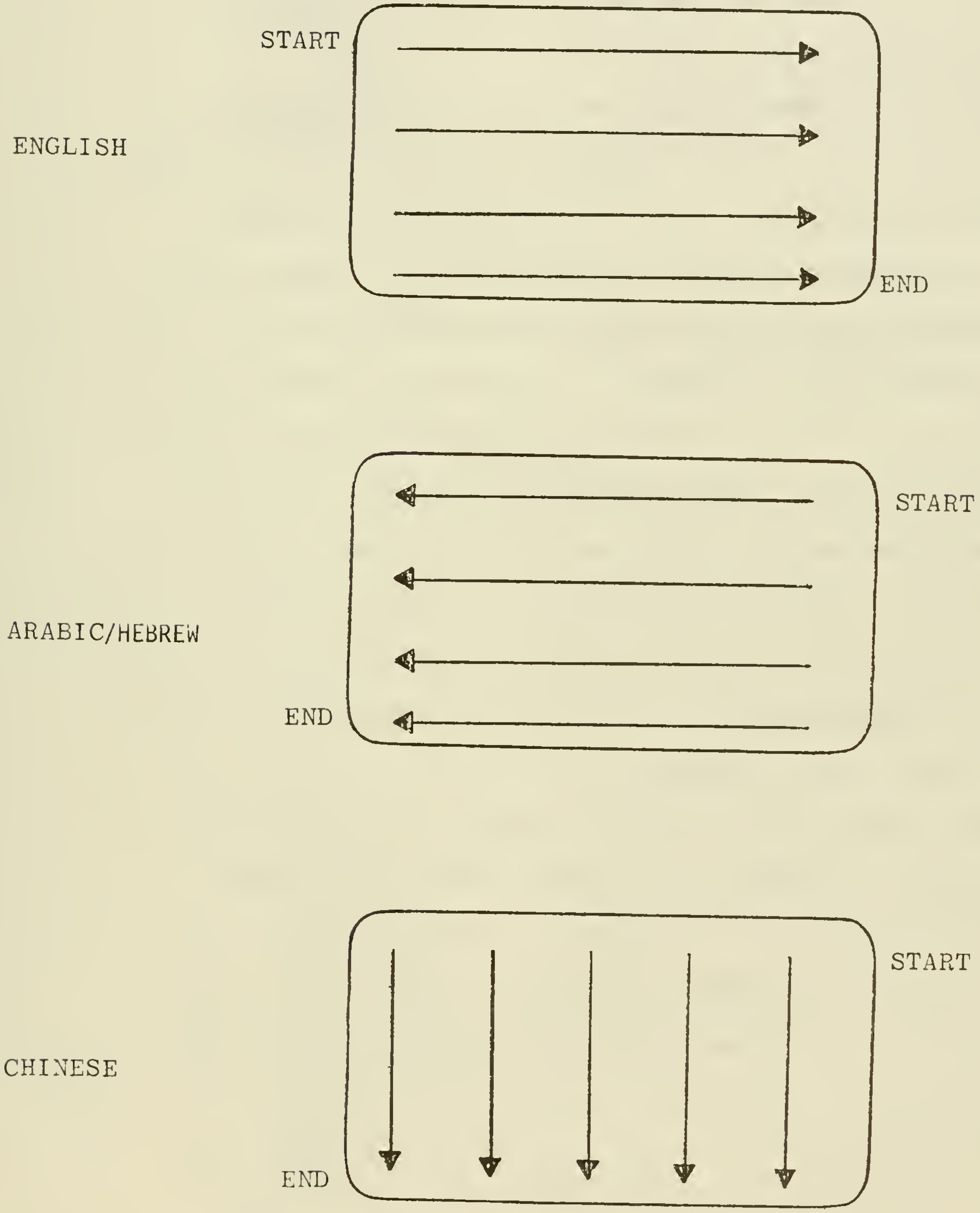


FIGURE 6

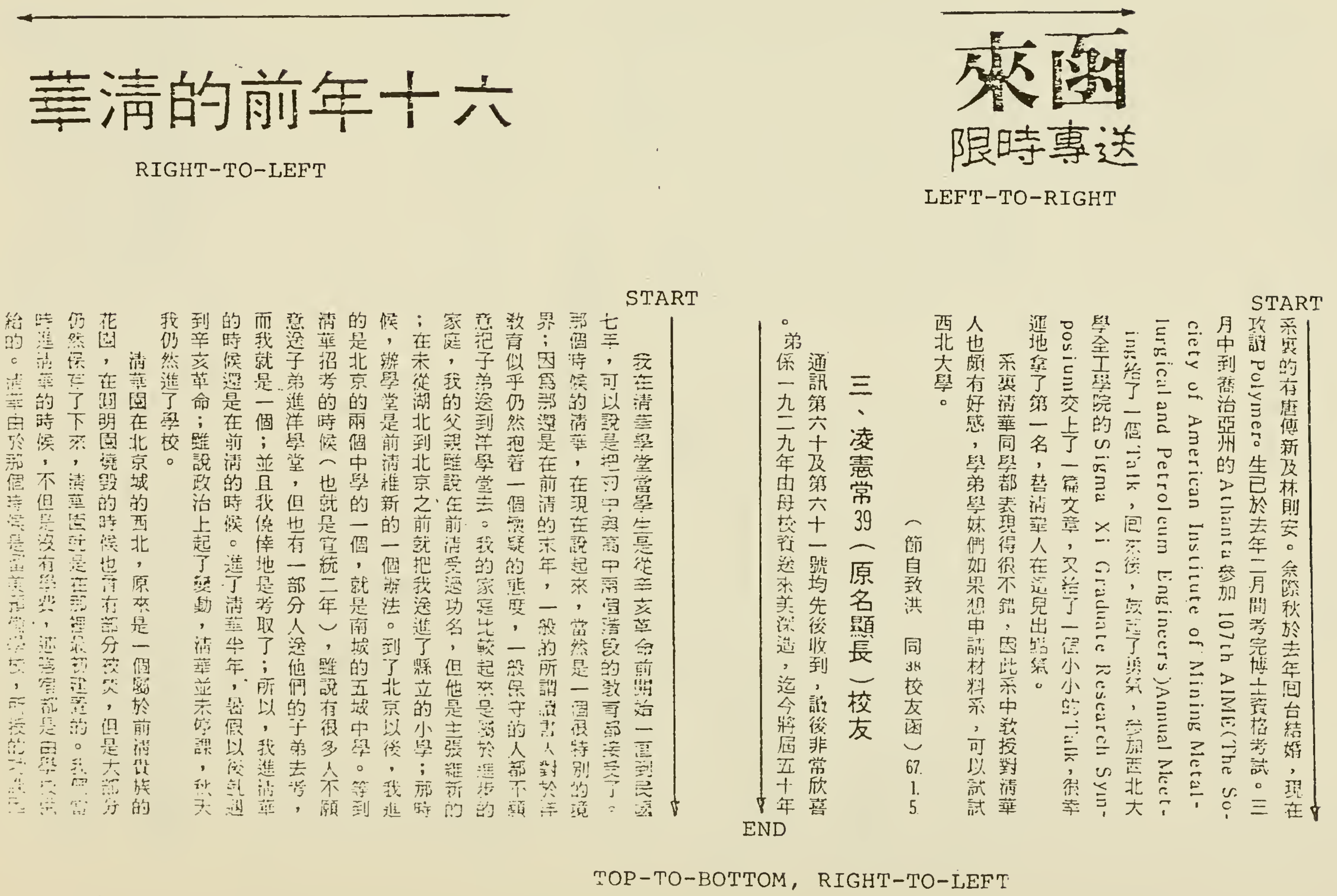
devices, and/or special software routines developed to re-sequence input and output. Printed lines must be structured accordingly, with minimal impact once the necessary changes are made to the actual print element. In search of efficiency, bi-directional techniques have already been developed in sorting and printing.

Chinese, on the other hand, is most often read/written from top to bottom, symbol following symbol, along the same vertical column. Columns follow each other from the right to the left-hand side of a page or facsimile. Because of the non-alphabetic character of Chinese symbols, the language is sometimes also read/written row-wise from left to right or from right to left. (See Figure 7.) Here we have an added level of complexity, both in terms of data entry and printing output. Unless we are to print characters sideways and output facsimiles on their sides, in a line printer this implies the need to develop a whole page of output before a line can be physically put out. This is not a problem, of course, for devices which output a page at a time, such as the very fast printers recently developed using ink jet and/or laser technology, which contain adequate internal buffer memories.

Another important item concerns a primitive writing method called "boustrophedon," apparently abandoned by most cultures in their development of scripture. The term means "as the ox plows" in Greek. Text scripted in boustrophedon would be written (and thus read) from left to right on one line and from right to left on the next. (See Figure 8.)

READ/WRITE DIRECTIONS IN CHINESE: THREE EXAMPLES FROM SAME DOCUMENT

SOURCE: TSING HUA UNIVERSITY GAZETTE, NO. 63, APRIL 29, 1978. PG. 4 & 122


LEFT-TO-RIGHT
RIGHT-TO-LEFT
START
END
START
END
TOP-TO-BOTTOM, RIGHT-TO-LEFT

來函

限時專送

系裏的有唐傳新及林則安。余際秋於去年回台結婚，現在攻讀 Polymer。生已於去年二月間考完博士資格考試。三月中到喬治亞州的 Athanta 參加 107th AIME(The Society of American Institute of Mining Metallurgical and Petroleum Engineers)Annual Meeting給了一個 Talk，回來後，[illegible]，參加西北大學全工學院的 Sigma Xi Graduate Research Symposium交上了一篇文章，又給了一個小小的 Talk，很幸運地拿了第一名，替清華人在這兒出鋒氣。

系裏清華同學都表現得很不錯，因此系中教授對清華人也頗有好感，學弟學妹們如果想申請材料系，可以試試西北大學。

（節自致洪　同38校友函）67. 1. 5.

三、凌憲常39（原名題長）校友

通訊第六十及第六十一號均先後收到，讀後非常欣喜。弟係一九二九年由母校資送來美深造，迄今將屆五十年

華清的前年十六

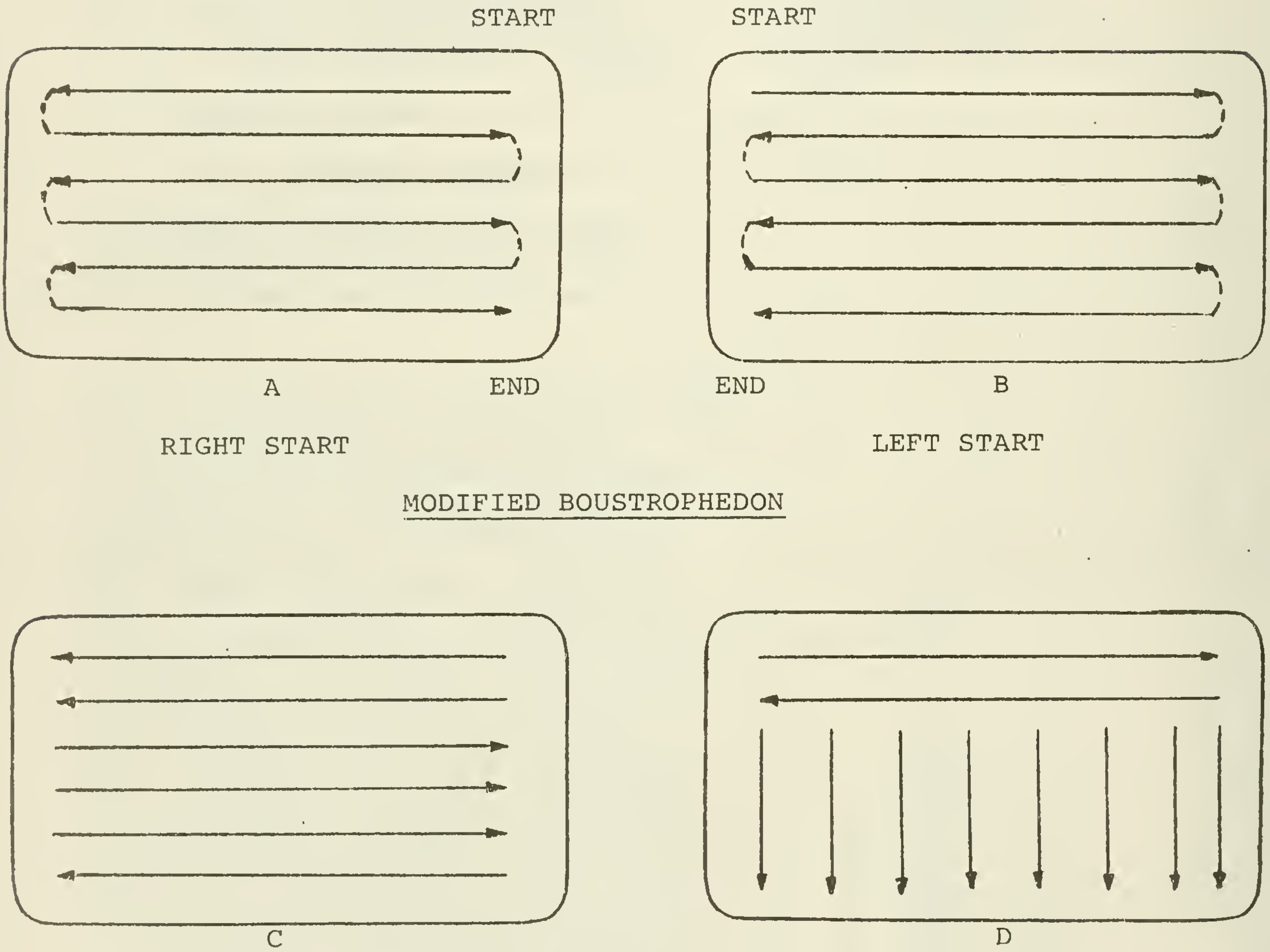
我在清華學堂當學生是從辛亥革命前開始一直到民國七年，可以說是把初中與高中兩個階段的教育都接受了。那個時候的清華，在現在說起來，當然是一個很特別的境界；因為那還是在前清的末年，一般的所謂讀書人對於洋教育似乎仍然抱着一個懷疑的態度，一般保守的人都不願意把子弟送到洋學堂去。我的家庭比較起來是屬於進步的家庭，我的父親雖說在前清受過功名，但他是主張維新的；在未從湖北到北京之前就把我送進了縣立的小學；那時候，辦學堂是前清維新的一個辦法。到了北京以後，我進的是北京的兩個中學的一個，就是南城的五城中學。等到清華招考的時候（也就是宣統二年），雖說有很多人不願意送子弟進洋學堂，但也有一部分人送他們的子弟去考，而我就是一個；並且我僥倖地是考取了；所以，我進清華的時候還是在前清的時候。進了清華半年，寒假以後就遇到辛亥革命；雖說政治上起了變動，清華並未停課，秋天我仍然進了學校。

清華園在北京城的西北，原來是一個屬於前清貴族的花園，在圓明園燒毀的時候也有部分被焚，但是大部分仍然保存了下來，清華園就是在那裡建造的。我們當時進清華的時候，不但是沒有學費，連膳宿都是由學校供給的。清華由於那個時候是[illegible]

FIGURE 7

EXAMPLES OF BOUSTROPHEDONIC WRITING

"AS THE OX PLOWS"



RIGHT START

LEFT START

MODIFIED BOUSTROPHEDON

ENGLISH SYSTEM MESSAGES IN ARABIC TEXT

TRILINGUAL DISPLAY: ENGLISH, HEBREW AND CHINESE

FIGURE 8

Boustrophedonic[24] writing may be coming back in more ways than one. More important, there are fairly obvious applications for some form of boustrophedon in information systems technology. Let us consider a terminal attached to a system in an Arabic country where the direction of read/write is right to left. Naturally it would be desirable for an Arabic terminal system user to be able to enter textual and other required data in the normal right-to-left convention of his language (which is opposite to that used in the design of most computer terminals). But it would also be useful to have the capability on the same Arabic terminal for handling internal system diagnostics and messages in English, which, of course, uses the left-to-right read/write convention. A boustrophedon-like direction-reversing capability would handle this problem nicely.

Another example involves the use of full screen CRT editing applications, so common today in many situations. For certain types of text, the edit process could be made more efficient if the cursor could automatically skip from the right end of one line of data to the same end of the one below it and then proceed from right to left until reaching the left end of the second line. At the end of the second line it would drop to the left end of the third line and proceed in the normal left-to-right direction. This is exactly the procedure used in boustrophedon and it would eliminate

[24] It is interesting to note that boustrophedonic writing, to be unambiguous, needs an asymmetrical or directed alphabet. Otherwise, for example, we might read in English TOM both as the male name (Tom) and as the noun meaning witty saying (mot). R.M. Barquin has developed such an alphabet based on the Roman alphabet, and he has also copyrighted a method for reading and writing English and Spanish text bi-directionally. Some early experiments show up to 25% efficiency increases in reading time using this method. (New Reading and Writing Method. Copyright A 545167; Dual Purpose Alphabet--Consonants and Vowels. Copyright A 545168)

moving the cursor up to a full line to the left to begin processing a new row of data which is typically necessary with conventional terminals.

Summarizing, we can say that information systems technology has been overwhelmingly developed to handle data read/written from left to right with the English subset of the Roman alphabet. Languages which are read/written with other subsets of the Roman alphabet, with non-Roman alphabets, with non-alphabetic script and in other than the left-to-right direction offer added complexity in their handling with the DP hardware and software we presently have.

V LANGUAGE AND COMPUTER PROGRAMMING

The computer program is the means through which people communicate to the machine the parameters, sequence, logic, and scope of its desired operations. As a means of communicating ideas, it is only reasonable that we do it through a programming "language."

Although the first programs were written directly in binary code, the trend has been dramatically shifted toward the development of higher level languages that minimize the programming effort by approximating a combination of natural language and algebraic notation. This has been expecially true for commercial data processing culminating in the development of COBOL (Common Business Oriented Language), by far the most widely used programming language in the world.[25] Of course, natural language in this case has meant English for all practical purposes, and thus COBOL is English-like in its syntax, vocabulary and grammatical structure. PL/1 resembles COBOL in this respect. While it would be unreasonable to say that COBOL even approximates a natural language similarity, we must admit that it is English that it has attempted to mimic in the structural characteristics mentioned above.

In addition, the process continues toward the development of English language capabilities for communicating with general purpose application programs, principally for inquiry purposes.

[25] This excludes program generators such as RPG and RPGII, which are widely used in small computers, but cannot be truly classified as programming languages. Some figures for Latin America can be seen in Barquin (1974), op cit.

Of course, to the non-English-speaking world, these phenomena pose two different questions. First, there is the problem of efficiency and effectiveness in learning; and second, the issue of linguistic interference in the non-English-speaking programmer.

There seems to be some suggestive evidence indicating that the fluent or native English speakers[26] learn high-level programming languages more efficiently (and perhaps more effectively) than those who are not proficient in English. It would seem logical that this be so since many high level languages resemble English, as we have pointed out, by developmental intent. In addition, the alphabetic word construction, Roman alphabet usage, English-like vocabulary, and the left-to-right read/write direction provide an environment biased toward the English speaker. Let us examine this through an example. Assume that two programmer trainees, one fluent in English and the other not fluent in English, but of equal intelligence and aptitude, simultaneously enter a COBOL programming course. Then we hypothesize that,given equal effort,the English speaking student (let us call him "native English speaker") will reach a level of average programming productivity in less time than the one who is not fluent ("non-native English speaker"). Figure 9 illustrates this proposition graphically. The cross-hatched wedge represents an economic cost, which must be aggregated for all applicable cases.

[26] The concept of "native or non-native English speaker" might be misleading. We are not dealing with a discrete binomial here, but rather a multidimensional continuum with full competency and total ignorance at opposite extremes for reading, writing, comprehension, syntax, vocabulary, et cetera. These do not necessarily depend on having been born into a linguistic area. However, for purposes of our example, the terms are useful.

TRAINING OF COBOL PROGRAMMERS
IN A NON ENGLISH SPEAKING CULTURE

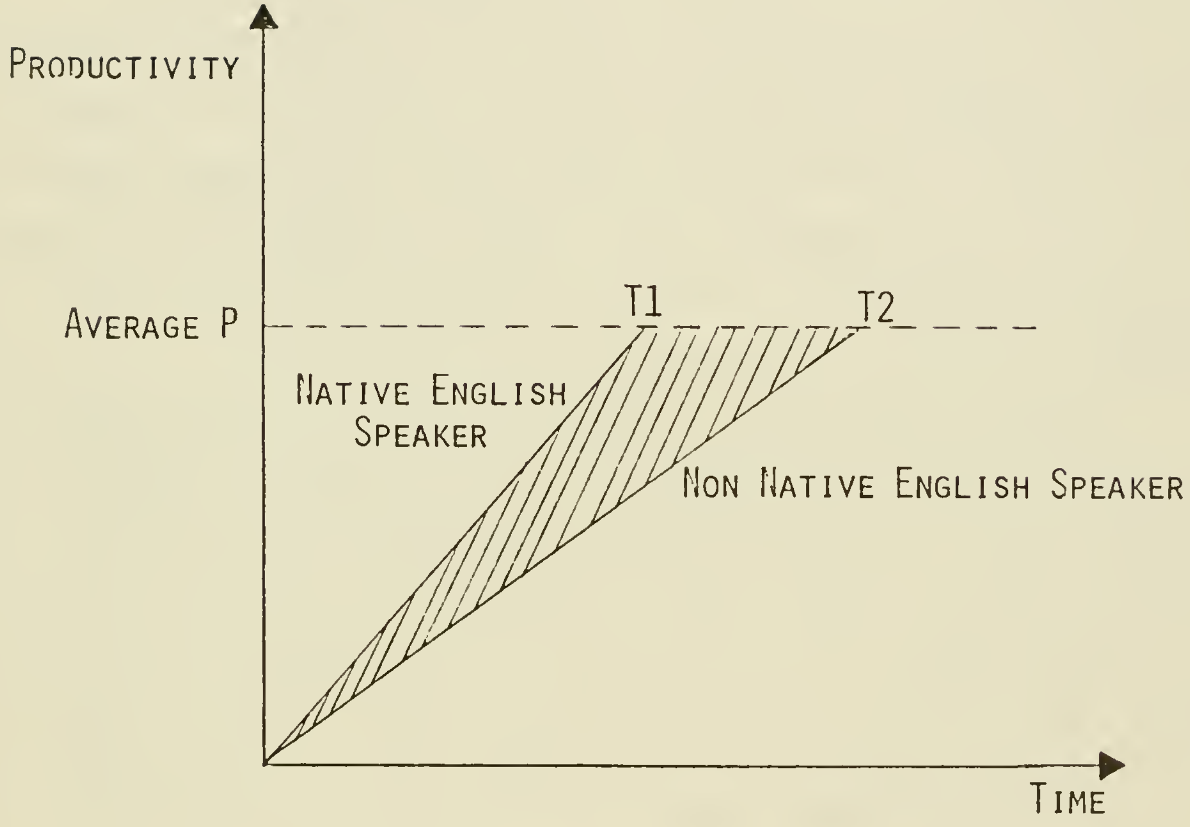


FIGURE 9

We hypothesize that there is an efficiency-in-learning problem that goes beyond the problem of student-teacher communication. Even if the "non-native English speaker" were to learn COBOL in his vernacular, there would still be a cost associated with the problems mentioned before.

Part of the rationale here, of course, deals with the issue of English being the language of science and technology, as was previously discussed. But programmers, systems operators, systems analysts and systems end-users may not generally be expected to be as competent in English as the research scientist or technologist. Section VI describes an experiment which we conducted in order to further explore some of these issues.[27]

This does not necessarily suggest that we should go out and develop a myriad of foreign language compilers. In fact, there are a number of instances where this has been attempted without dramatic change in usage or performance patterns.[28] However, the cost to a culture of this hypothesized inefficiency merits at least some consideration leading to quantification and a decision on the merits of developing an acceptable solution. An English-like programming language is, in a sense, a cultural element. Hall warns us (and, in fact, shows some evidence) that "when cultural elements are borrowed, there may be a mismatch between the borrowed item and the borrowing culture."[29]

[27] K.S. Seo, "A Study of Linguistic Issues in the Utilization of Information Systems," Unpublished Master's Thesis, Sloan School of Management, MIT, January 1978, 80 pp.

[28] A COBOL subset exists in French and the Japanese have a katakana COBOL. A group under Prof. W. Setzer at the University of Sao Paulo (Brazil) developed a Portuguese FORTRAN.

[29] E.T. Hall illustrates this point with the example of the wide American automobile in the narrow traditional streets of Europe or Japan. See E.T. Hall, "In the Wake of Technology," in Condon and Saito (eds.), op cit, p. 94.

Finally, there is the important issue of the relation between language and thought. In Whorf's writings[30] two basic points seem to be emphatically argued: first, that all higher levels of thinking are dependent on language; second, that the structure of the language one habitually uses influences the manner in which one understands his environment. This clearly implies, then, that thought patterns and behavior are culturally dependent. In effect,Whorf focuses on this through his intensive studies of the American Indian languages and culture (especially the Hopi).[31]

To visualize the potential implications with respect to computer programming, let us resort to the following example: Consider a non-English speaking Japanese programmer who conceptualizes largely through a language written with thousands of distinct non-alphabetic characters, phonetic ambiguities, a sentence structure diametrically different from English, and an inverted read/write direction. To communicate with his computer in COBOL, he now must manipulate symbolically the rudiments of this new "language" into a program. There seem to be levels of complexity involved in this task for the Japanese programmer which are not present for the native English speaker.

[30] . This is the direct view of Stuart Chase in his foreword to the selected writings of B.L. Whorf, Language, Thought and Reality, edited by John B. Carroll, MIT Press, Cambridge, MA, 1956, p. vi.

[31] . Whorf's analysis dwells on the thought patterns of the primitive American Indian communities which he studied. In "An American Indian Model of the Universe," op cit, pp. 57-64, he starts by explaining that the Hopi language contains no words or expressions that refer directly to what we call "time" or to the past, present, or future. From this he tries to construct a Hopi view of the universe which is intelligible to us.

Also directly relevant in arguing this point are Whorf's "The Relation of Habitual Thought and Behavior to Language," op cit, pp. 134-159, and "Language, Mind and Reality," op. cit, pp. 246-270.

In addition, algorithms are the hard core of any program, and algorithmic construction is essentially a logical process. But logical analysis is also related to the linguistic context in which it is done.[32] This implies that potential constraints are imposed on some programmers in the development of algorithms as a result of the linguistic translation process we force him to go through. In a way, this problem is related to the equivalent of Chomsky's concept of acceptability and grammaticalness. "The more acceptable sentences are more likely to be produced, more easily understood, less clumsy, and in some sense more natural."[33] The distance between English and the coder's vernacular in terms of basic structure will probably determine the degree of acceptability of the written program. But the forced search for competence (grammaticalness) through acquired linguistic skills does not insure acceptability (performance) and in fact could hamper the end result which is the successful construction of the algorithm itself.

[32] In Whorf's "Languages and Logic," op cit, pp. 234-245, he uses the Shawnee and Nootka languages and compares them to English in illustrating the constraints to logical analysis presented by the linguistic context.

[33] See N. Chomsky, Aspects of the Theory of Syntax, MIT Press, Cambridge, MA, 1965, p. 11.

VI LINGUISTIC INTERFERENCE AND PROGRAMMER LEARNING

The concept of linguistic interference can be very broad. Let us try to identify three distinct types of linguistic interference which should be addressed independently.

First of all, we have the broad problems related to man and his technological milieu. That is, how man communicates and learns about his technology. Of course, when the technological environment is dominated by a language different from that of the technologist, interference will probably occur. This is the case with information systems technology whose education is frequently conducted in English, and with English documentation. Non-English speaking technologists will have varying degrees of difficulty as a result of this aspect, which we shall call learner linguistic interference.

A second type of linguistic interference is that which occurs when man as end-user communicates with a computer; that is, man as user of the technology. Where the machine is programmed to produce English output, and the user is not English speaking, clearly problems arise. Examples of this could be the case of a physician not fluent in English using an English computer-aided diagnostic decision support system; or of a non-English-speaking high school student using a computerized career education program, such as CVIS or DISCOVER. Let us call this aspect operator linguistic interference.

Finally, we have the issue of linguistic interference as it affects the designer and implementer of a computer-based system through a programming language. We control the machine by communicating instructions in the form of a program. This program is written utilizing any one of a number of programming "languages." These languages, however, have themselves been developed by designers who have projected the characteristics of their native tongues--generally English--onto the programming language. Thus we wind up with a programming language like COBOL which, like English, is written from left to right, using the Roman alphabet, English words as operators, and having a syntax and grammar similar to English by design. When a programmer who is not fluent in English communicates with the computer through this means, it is quite probable that some linguistic interference occurs. Let us refer to this aspect as designer/implementer linguistic interference.

The experiment described below was designed principally to explore ramifications of learner and designer/implementer linguistic interference.

Although we have identified several issues which could potentially lead to linguistic interference in any learning situation, it is relevant to note here that the learning problem is compounded in the case of computer programming because the computer language is the object of the learning experience itself (unlike the problem of learning history or physics in a non-native language).

In order to test some of these ideas, a simple experiment was done in which data was collected on several undergraduate and graduate university students from multiple cultural settings who were taking introductory courses in computer programming. This experiment was designed to provide some preliminary insight into several questions related to linguistic interference and programmer learning:

- Do non-native English speakers have more difficulty in learning English-based programming languages (EBPLs) than do native English speakers (i.e., does the postulated linguistic interference actually occur)?

- Is degree of English competence in the written form versus the spoken form a factor in programmer learning of EBPLs?

- Are non-native English speakers aware of linguistic interference as a factor in their learning of EBPLs?

- Are there specific aspects of programming languages (syntax, mnemonics, reserved words, read/write direction, etc.) which are more troublesome than others for non-native English programming students?

- Are there factors other than the programming language itself (such as error diagnostics, documentation, etc.) which add to the hypothesized linguistic interference phenomena?

- What are the implications of linguistic interference for the development and teaching of programming languages?

VI-1 Previous Research

Although we do not know of any specific research to date that provides unambiguous answers to these questions, a foundation of prior work in related areas lends strong suggestive evidence to the proposition that linguistic interference might be a significant problem. The whole bilingual education argument can be brought to bear on one aspect of this problem when English is also the teacher's language and that of the texts. The work of Chomsky and Whorf in psycholinguistics has already been cited along with other contributions in the general socio-cultural contexts. Further suggestive evidence can be found in the literature on psychological research which presents learning "principles" which may be potentially useful in practice.[34]

[34] A good review of this work can be found in E. R. Hilgard and G. H. Bower, Theories of Learning, Appleton Century Crofts, New York, 1966.

The reason for writing "principles" in quotation marks is that these generalizations reflect empirical evidence that seems to hold widely. However, they are not stated at the level of detail necessary to be considered "laws" of learning. Several of these generalizations which may be particularly relevant to programmer learning are listed below.

VI-1.1 Principles Derived From Cognitive Theory

The perceptual features according to which the problem is displayed to the student are important conditions of learning. These include issues such as figure-ground relations, directional signs, "what leads to what," and organic-interrelatedness. Perceptual features are key determinants of the representation system (a computer language derived from a natural language) that a programmer must learn to cope with. If certain perceptual features are alien to the programming student, he may have more trouble in understanding and manipulating them.

The organization of knowledge that is to be presented is an essential element in the success of the learning experience. This suggests, among other things, that the basic building blocks of the subject to be taught should be understandable as complete logically consistent subunits of the larger subject of concern. Acronyms, mnemonics, and reserved words in English-based programming languages may not meet this criteria for non-native programmer trainees.

VI-1.2 Principles Derived From Motivation and Personality Theory

The group atmosphere of learning tends to affect both the satisfaction in learning as well as the products of learning. Important group factors include competition versus cooperation, authoritarianism versus democracy, and individual isolation versus group identification. It can be postulated that inability to communicate as efficiently and effectively as native English students could lead non-native students toward greater isolation and provide fewer incentives for attempted cooperation and competition with peers.

Learning is culturally relative. The wider culture and the subculture (aspects of which are expressed through the language) of the learner may affect his learning. This important research finding generally supports contentions that have been made earlier in this paper.

VI-1.3 Problems in Recognition of the Spoken Language

It is also true that non-native English-speaking students who are very competent in written English may have trouble in acoustically recognizing words that they easily identify in written form if they are lectured in English. Students transferring from non-English cultural settings for advanced training at the undergraduate and graduate university level are often relatively competent in the reading and writing of the language but significantly less competent in producing and recognizing the spoken version of the language. Suppes, et al, show experimental evidence, for instance, of the significant difficulty of phoneme-allophone discrimination in listening to a foreign language.[35]

[35] An allophone is an acoustic variant of a phoneme. For instance, the three p's in speech, peach, and topmost are not equally explosive, and thus they are allophones of the phoneme p. See P. Suppes, E. Crothers, R. Wier, and E. Trager, Some Quantitative Studies of Russian Consonant Phoneme Descrimination, Stanford University, Institute for Mathematical Studies in the Social Sciences, Technical Report 49, CA 1962.

VI-2 Experimental Design

Six groups of students consisting of a total population of 456 subjects were studied during their introduction to the FORTRAN, PL/1 and/or BASIC programming languages. The students were registered at two U.S. universities and data was collected on a subset of the sample via a self-report type questionnaire. It contained questions regarding several issues thought to be potentially significant in detecting and better understanding the effect of linguistic interference in programmer learning. The questionnaire included data on a wide variety of issues thought to be potentially relevant to the learning of a programming language.[36] Foreign students were asked questions to indicate their relative competence in speaking, reading and writing in English. They also answered questions related to their tendencies to "think" in English, the amount of cultural exposure to computers which they had experienced in their home countries, and any specific problems with aspects of the programming languages themselves such as read/write direction, type of alphabet, et cetera. All students were asked to rate the extent of problems they had with lectures, class materials, documentations on the programming languages, mnemonics, reserved words, syntax, system diagnostic messages, or other related items.

[36] The results included information on age, sex, nationality, education level, area of educational concentration, natural languages understood and estimated degree of competence in these natural languages, programming languages known, ratings on ease of programming language learning, personal preferences for specific programming languages, ratings of different aspects of presentation (textbooks, lecture, exams, problem sets), course programming language, and amount of time spent on course. Information on the grade of all students was acquired directly or through the university records office.

The Kolb learning style inventory was also administered to all students answering the questionnaires in order to control for potential differences in individual's methods of learning.[37]

VI-3 Results

The data collected yielded both quantitative and qualitative results. Grades received in the course were taken to be a measure of student performance. The number of hours spent per week on the course were interpreted as a measure of student effort. Hypotheses about differences between native and non-native English speaking subjects were tested by T Statistic. The principal findings were:

- Non-native English-speaking (Non-E) subjects have more difficulty with programming language mnemonics than do native English speaking (E) subjects (T=3.18, p=.003)*
- Non-E subjects have more trouble with programming language reserved words than do E subjects (T=2.85, p=.006) .
- Non-E subjects receive lower grades in the programming courses than do the E subjects (T=2.47, p=.014)

*T statistic and its associated p are given in parentheses.

Hypotheses about the relationships between answers to questions on relative English competence, programming language difficulties, linguistic back-

[37] The Learning Style Inventory (LSI) was designed to assess an individual's method of learning. It is based on an "experimental learning model" in which learning is conceived of as a four stage cycle: (1) concrete experience (CE), (2) observations and reflections (RO), (3) formation of abstract concepts and generalizations (AC), and (4) testing implications of concepts in new situations (AE). Each stage requires corresponding abilities. Most people develop learning styles that emphasize one or more of these learning abilities. The LSI measures an individual's relative emphasis on the four learning abilities. Kolb's research shows a high correlation between choice of college major and LSI scores. Certain learning styles seem to either develop because of one's education or lend themselves to certain majors. In our study, no noticeable cross-cultural implications of learning styles were detected. See D. Kolb, I. Rubin and J. McIntyre, Organization Psychology: A Book of Readings, 2nd edition, pp. 27-42.

ground, course effort, and course performance were tested with a Pearson correlation coefficient (C). Because of the lower sample sizes in these groups the significance levels (S) tended to be higher than we would desire, and a cutoff of .15 was used. However, the directions of the coefficients for the native English versus non-native subject were quite supportive of the hypotheses. For these tests, the populations were separated into Non-E and E subsamples. The results[38] indicated that non-E subjects who stated they tended to think more in English when they programmed got higher course grades. In addition, non-E subjects who rated themselves higher in English competence received better grades. Interestingly, written competence was more significant than spoken competence.

[38]
- Non-E subjects who were having problems with programming language documentation tended to spend more hours per week on the course (C=.581, S=.078).*
- Non-E subjects who were having problems with programming language mnemonics tended to get lower grades (C=.887, S=.114).
- Non-E subjects who had problems with programming language reserved words tended to spend more time on the course (C=.651, S=.113).
- Non-E subjects who had problems with programming language syntax tended to spend more time on the course (C-.474, S=.141).
- Non-E subjects who had difficulty with system diagnostics tended to spend more time on the course (C=.474, S=.116).
- Non-E subjects who rated themselves higher in English competence received higher course grades (C=.730, S=.100). Written competence was more significant than spoken competence in this respect.
- Non-E subjects who stated that they tended to think more in English when programming got higher course grades (C=.755, S=.083).

Non-E students who indicated they were having problems with programming language mnemonics tended to get lower grades in the course. And as would be expected, non-E students who said they were having difficulties with programming language documentation, reserved words or syntax tended to spend more time on the course.

These statistically significant differences in results for non-native English-speaking subjects would seem to support the linguistic interference argument; especially so in light of the fact that they were students with some competence in English, and that they were dealing with programming languages that were not as English-like as, for instance, COBOL.

IV-4 Implications for Future Research

The findings presented in the experiment have provided supportive evidence that linguistic interference may play a significant role in programmer learning effort and performance for non-native English-speaking students. Several issues should be noted here, including limited sample size, potential response biases on self-report type questions, population selection, and prior English training of the foreign subjects as complicating factors in interpreting the results. Such issues invariably affect the design and analysis of research that involves humans as subjects. Nevertheless, hypotheses about the expected role of linguistic interference in programming student performance and effort were clearly supported.

At this stage, a series of larger scale experiments conducted under more tightly controlled circumstances in a foreign (non-English) setting would be desirable. It would be very interesting to see the experimental performance and effort measures which we used earlier tested in a learning setting that exposed controlled groups to an English and a native-language version of the same high-level programming language compiler. COBOL would be the most desirable language processor because of its extensive use world-wide. In the next experiment we would advocate a random group two-by-two factorial design in order to isolate problems and effects associated with programming language documentation/teaching materials and the programming language itself. The following diagram illustrates the design.

		Documentation and Teaching Material (D/TM)	
		English	Native
Programming Language Processor (PLP)	English	English D/TM English PLP	Native D/TM English PLP
	Native	English D/TM Native PLP	Native D/TM Native PLP

This experiment should yield insights into the desirability of providing translated documentation and operational native language compilers for each cultural context that is of interest. It would also provide insight, if carried out in multiple non-English cultural contexts, into the relevance of other specific language characteristics such as read/write direction, alphabetic versus ideographic written forms, and other syntactic and semantic constructs.

VII SUMMARY

Mesthene has proposed that new technologies, if applied, must lead to social change and to change in social and individual values; and that the incidence of this process will make change the principal characteristic of the world we deal with.[39] But, by the same token, cultures tend to mold their environments in order to minimize the social instability generated.

This preliminary research has presented suggestive evidence that linguistic interference, one of many cultural issues raised, may be a significant factor in programmer learning performance. The principal results of this study indicated that students from non-native English backgrounds spent more time on course material and had more trouble with programming language documentation, syntax, mnemonics, reserved words, and system diagnostics than native English-speaking students. They did better in the courses if their English language proficiency was higher or if they tended to "think in English."

It is precisely this impact of culture on technology that we have tried to focus on in this paper by isolating information systems technology as an individual case, and developing an analytical framework to guide future research. As intellectual interest, economic imperatives or simply market demand expand these studies into specific areas, we suspect that patterns will emerge providing insights for other technologies as they interact with Earth's many cultures. In a time when the world clamors for "appropriate" technologies, it seems more important than ever to proceed emphatically with this investigation.

[39]. See E.G. Mesthene, "How Technology Will Shape the Future," in H. von Foerster, et al, (eds.) Purposive Systems, Spartan Books, New York, 1968, p. 67.

VIII ACKNOWLEDGMENTS

Work herein has been supported in part by the IBM World Trade Americas/Far East Corporation, the IBM Cambridge Scientific Center, the MIT Center for Information Systems Research and the MIT Alfred P. Sloan School of Management.

It is very difficult to acknowledge all of the MIT, IBM and other personnel who have contributed directly or indirectly to this effort. Jack Rockart, current Director of the Center for Information Systems Research, provided valuable insights and guidance from the earliest days of the project along with Michael Scott Morton, prior Director of the Center and now Associate Dean of the Sloan School. Karen Seo, a Sloan School graduate student, carried out important aspects of the empirical investigation which was described in the paper. Joe Saliba of IBM insisted throughout the study that the research be geared toward the development of practical and implementable results and provided many suggestions and opinions at critical stages. Other important advisors and contributors included Dick Robinson (MIT), Jorge Domingus (Harvard), Bob Fano (MIT), Dick Mackinnon (IBM), Gerry Fodor (MIT), Al Marcotte (Boston University), Joe Weizenbaum (MIT), Joseph Chou (IBM), George Wu (IBM), and Jay Yablon (IBM).

IX BIBLIOGRAPHY

Automation in Developing Countries, International Labour Office, Geneva, Switzerland, 1972, 246 pp.

Apple, R. W. Jr., "Two Britons Devise a Computer That Can Communicate in Chinese," New York Times, January 25, 1978.

Barquin, R. C., The Degree of Penetration of Computer Technology in Latin America: A Survey, MIT Sloan School of Management Paper No. 702-74, MIT, April 1974.

----, "On Computer Software, Education and Personnel in Developing Countries," ITCC Review, Vol, V, No. 16, International Technical Cooperation Center, Tel Aviv, Israel, January 1976, pp. 11-22.

----, "Computation in Latin America: An Annotated Bibliography and Other Sources of Information," Latin American Research Review, Vol. 11, No. 1, pp. 75-102.

Barquin, R. M., "New Reading and Writing Method," Copyright A545167, 1971.

----, "Dual Purpose Alphabet--Consonants and Vowels," Copyright A545168, 1972.

Bauer, R. A., Second-Order Consequences, MIT Press, 1969, 240 pp.

Bernal, J. D., The Social Function of Science, MIT Press, 1939, 482 pp.

Chomsky, N., Aspects of the Theory of Syntax, MIT Press, 1965, 251 pp.

Condon, J. E. and Saito, M. (Eds.), Communication Across Cultures, For What?, The Simul Press, Tokyo, 1976, 141 pp.

Diringer, D., The Alphabet: A Key to the History of Mankind, 3rd Revised Edition, Vol. 2, 1968.

Goulet, D., The Uncertain Promise: Value Conflicts in Technology Transfer, IDOC/NA., Inc., New York, 1977, 320 pp.

Hill, G. C., "More Firms Turn to Translation Experts to Avoid Costly, Embarrassing Mistakes," Wall Street Journal, January 13, 1977.

Klemm, F., A History of Western Technology, MIT Press, 1964, 400 pp.

Ladriere, J., The Challenge Presented to Cultures by Science and Technology, UNESCO, Paris, 1977, 165 pp.

Lindzey, G. (Ed.), Handbook of Social Psychology, 1st Edition, Addison-Wesley, Reading, Massachusetts, 1954.

Meador, C. L., and Ness, D. N., "Decision Support Systems: An Application to Corporate Planning," Sloan Management Review, Vol. 15, No. 2, Winter 1974.

BIBLIOGRAPHY, Continued

Menninger, G., (Ed.,), Number Words and Number Systems: A Cultural History of Numbers, MIT Press, 1977, 479 pp.

Miller, I., Text Evaluation of World Languages, IBM Technical Report TR 00-2561, Poughkeepsie Laboratory, September 24, 1974, 22 pp.

Parhami, B., "On the Use of Farsi and Arabic Languages in Computer Based Information Systems," Arya-Mehr University of Technology, Tehran, Iran, March 1975, 11 pp.

Parhami, B. and F. Mavaddat, "Computers and the Farsi Language--A Survey of Problem Areas," 1977 IFIP Congress Proceedings, North Holland Publishing Co., 1977, pp. 673-676.

Pool, I. de Sola (Ed.), The Social Impact of the Telephone, MIT Press, 1977, 502 pp.

Rhinesmith, J. H., Cultural-Organization Analysis, McBer and Co., 1971, 53 pp.

Richman, B. M., "Significance of Cultural Variables," Academy of Management Journal, Vol. 8, No. 4, December 1965.

Schumacher, E. F., Small is Beautiful, ABACUS Edition, London, 1973, 255 pp.

Shepard, J. M., Automation and Alienation, MIT Press, 1971, 163 pp.

Silvert, K. H. (Ed.), The Social Reality of Scientific Myth, American Universities Field Staff, New York, 1969, 239 pp.

Smith, F. nad G. A. Miller (Eds.), The Genesis of Language, MIT Press, 1966, 400 pp.

Vikas, Om, "Use of Non-English Language in Computers--A Selected Bibliography," Indian Electronics Commission, June 1978, 69 pp.

Webber, R. A., Culture and Management, Richard D. Irwin, Inc., Homewood, Illinois, 1969, 598 pp.

Whorf, B. L., "Language, Thought and Reality," MIT Press, 1956, 278 pp.

Wiewall, M., Jr., "Algunos Datos Sobre Caracteristicas de Informacion del Idioma Espanol," Paper presented to Pan American Congress of Mechanical, Electrical and Related Engineering, San Juan, Puerto Rico, September 6-13, 1969, Session 4, Number 194.