SOFTWARE LOCALIZATION: NOTES ON TECHNOLOGY AND CULTURE

Kenneth Keniston

Kenneth Keniston is Andrew W. Mellon Professor of Human Development and Director of Projects in the Program in Science, Technology, and Society at the Massachusetts Institute of Technology.

January 17, 1997

Working Paper #26

Program in Science, Technology, and Society

Massachusetts Institute of Technology

Cambridge, Massachusetts 02139

The comments that follow are intended as a preliminary map of some of the territory to be explored if cultural localization is to be understood.

Introduction

"Cultural localization" is the process whereby software written in one culture is adapted to the needs and outlooks of another. As one correspondent puts it, "Culturally localized software is indistinguishable from software written by a member of that culture." Cultural localization presupposes linguistic localization, a topic well studied, but may go far beyond it. For software developed in one culture can carry embedded cultural assumptions that may seem alien or even inimical to users in other cultures. At present, cultural localization almost always entails the localization of packaged software originally written in English by American programmers, since U.S. software dominates the world market. But there is no logical reason why this should be so; the problems of localization from, say, Hindi to French, or German to Xhosha, are not inherently different from those of localization from English to Chinese.

The study of cultural localization is important for at least three reasons.

- Intellectually, it is important to try to understand the assumptions built into the software that is assuming increasing importance in our lives. Just as we seek to understand how television affects how we perceive, think, and behave, so it is important to study how the software with which more and more people in the world work for more and more time each day influences choices they make, assumptions they take for granted, and choices they do not or cannot make. What values, if any, does software carry apart from the ability to solve problems? How does it affect the way we think, work, and live? Is it true, as social scientists argue, that software, like other technologies, is a covert carrier of culture and even of politics? Put differently, it is important to begin to "deconstruct" software, just as we now "deconstruct" film, television, and literature.

- Economically, the United States holds the dominant world position in packaged software production and exports, with about three quarters of the world market and exports valued at many billion dollars in 1995. American programming languages and operating systems are almost universally used across the world. Most major American software companies devote important resources to "internationalization"; current and new versions of Windows, for example, are planned in dozens of languages. But precisely because of the economic importance of U.S. software, American insensitivity to the perceived cultural messages in software could (and should) lead to reluctance on the part of other nations, developed and developing, to import this software. American software manufacturers are understandably concerned that such reluctance could lead to software import restrictions or to "local content" software requirements like those that now exist in some nations with regard to television and film.

- Politically, software localization requires confronting difficult and ambiguous international and domestic controversies. In many nations, cosmopolitan, internationally-oriented business, professional and intellectual elites hold dominant positions. For them, "localization" to vernacular languages and local cultures may be unnecessary and/or even undesirable, since English (or French, or Spanish, or another European language) may provide the best possible access to the rest of the world. Another broadly political question is the choice of languages in developing countries: for example, should children in the Andean regions of Peru use software localized to Quechua or

Spanish? Should businesses, banks, or schools in Calcutta adopt Bengali, Hindi, or English software? How can the legitimate interests of local, tribal, and historic cultures be balanced against the reality of an increasingly interdependent, multinational, and global economy and culture? At what level -- national, regional, local, individual -- and by whom should software decisions be made? Should developing countries try to institute uniform software policies or, like some industrial nations, let so-called "market forces" prevail? What are the educational, technological, and infrastructural preconditions for developing nations to assert control over the software used within their borders?

Software localization, often treated entirely as a technical problem of technology transfer, opens up a series of broader intellectual, economic, and political questions, each of which takes a special form in developing nations where educational and infrastructural resources may be limited.

The thoughts that follow are based on discussions with colleagues in Latin America, on a preliminary effort to explore the rather meager literature on localization and cultural issues in software, and, most of all, on Internet correspondence with professionals and scholars interested in the issue -- some professional localizers working for major and minor software firms, others academics who have studied, or are interested in, how culture affects software. A preliminary posting of a query about cultural localization on the ISWorld Listserv produced a flood of useful replies from researchers and workers the world over; contact with some members of LISA (the Localisation Industry Standards Association, Geneva) produced another set of useful responses. Although largely unacknowledged here, the Internet correspondence that followed has provided many of the ideas in this paper.

These thoughts are an agenda rather than a research proposal. My goal has been to consider a few of the broad issues posed and pointed to by the examination of cultural localization, and in particular to place those issues in two frameworks. The first is the framework of the contrast between the cosmopolitan, "telectronic", international economy and culture on the one hand and local, tribal, at times fundamentalist cultures on the other. The second is the framework of sub-cultural multiplicity and variety in every society in the world, a framework that requires us to distinguish purchasers from users, general public from hackers, elites from locals, leaders from followers, speakers of the official language from speakers of the vernacular, and so on. This pluralism within cultures necessarily complicates the task of studying cultural localization.

This essay is divided into the following sections:

-- the international role of American packaged software

-- technical and cultural localization

-- some "cultural" examples

-- culture, sub-culture, and hacker culture

-- internationalization, "Americanization", and local cultures

-- pro- and anti-Americanism

-- protectionism, copyright, and free trade

-- hardware versus software

-- English speakers and non-English speakers

-- users versus purchasers

-- is localization always desirable?

-- the special problems of the Net and the Web

-- the future of localization

-- research orientations

-- outcomes

-- endnotes with bibliographic discussion.

What follows is incomplete and subject to revision. As befits a working paper rather than a published article, I have at times made claims that I cannot fully substantiate, left footnotes incomplete, and stated dogmatically arguments of which I am in fact less than sure. I hope readers will react critically, correcting errors of fact and assumption, and contributing to the further study of this topic. The Endnotes provide an indication of some of the works I have found useful in thinking about this topic.

The International Role of American Packaged Software

Sales of American packaged software abroad constitute a multi-billion dollar market, with the promise of steady future growth if American software continues to be preferentially adopted in other nations. At present, the United States holds the dominant, indeed in some areas an almost monopolistic, worldwide position in the production of programming languages, operating systems, and, to a lesser degree, broad-based applications.1 Operating systems like DOS, Windows, OS2, UNIX, and Apple systems are of American design; so are most major third and fourth generation programming languages; most of the widely-used word processing, hypertext, spreadsheet, multimedia, Web, and other high-level languages are American; finally, specific applications are usually based on American operating systems and English-language programming languages.

The rapid spread of the Internet and more recently the World Wide Web has tended to reinforce the dominant position of English-language programs and communication systems. In fact, of course, messages can be sent in a variety of languages via the Internet; Web pages can be constructed in Korean, Thai, Farsi, Urdu, and Arabic as well as English, French, or Spanish. But in practice, and in part because of the technical problems of using ideographic or non-roman character sets, right-left or vertical scrolling, etc., on systems and search engines originally designed for

English or West European languages, Internet and the Web tend to be dominated by English-language -- and in general by American -- designs and ideas. A recent article reports that 89 percent of Internet addresses are located in only four primarily English-speaking countries (U.S., Canada, U.K., Australia); and an impressionistic survey of personal Web sites of students at several major universities in the People's Republic of China shows that most of them are, in whole or in part, in English.2 Efforts by Japanese and European software manufacturers to challenge the international hegemony of American packaged software have so far not been successful. Although there are a few notable exceptions like SAP, a German program, in general programmers like users have preferred to work from English rather than Japanese, French, or German.3

But the very success of American-based software creates problems of its own. If, as American manufacturers and designers hope, the growth of American software sales and licensing abroad is to continue, at least two basic requirements must be met: both have to do with what is commonly called "localization" or "internationalization" of programs.

Technical and Cultural Localization

The first and generally used meaning of "localization" has to do with the translation of programs originally written in and for one language into intelligible and user-friendly versions in and for another language. We may call this "technical localization", inasmuch as it usually entails, for example, creating other-language versions of U.S.-English language programs, with appropriate local character sets, numbers, scrolling patterns, dates, colors, box sizes, etc.

Localization was initially approached by American software firms as an "add-on"; i.e., after the original program was fully functional in English, "localizers" were put to work to produce a Spanish, French, Japanese, etc. version. But programmers soon realized that such ex post facto solutions were inadequate; they often produced absurd results; and in many cases they required the re-writing of source code, a costly step that could have been avoided had future internationalization been part of the initial programming plan. Like many other authors, for example, Kano,4 writing primarily about Windows, argues that localization must be part of the earliest design stages of any program, which must be written so as to make internationalization possible without rewriting the program's source code. The importance of this issue can be seen at major firms like Microsoft, which currently release dozens of language versions of new (or upgraded) operating systems and applications.

Anyone who has tried to work with foreign-language translations of American programs -- be they operating systems or applications -- can attest that the problems defined here as "technical localization" are far from universally solved. Absurdities abound; programs that operate well in one language, with one set of characters, crash in another; non-English speaking users may be presented with unintelligible, awkward, or English-language-only help, tutorials, or documentation; and so on. Yet most of these problems are, at least in principle, resolvable by the kinds of strategies suggested by Kano and others: planning for internationalization at the earliest stage of writing source code; consistent interaction of internationalizers with source code programmers; extensive use of indigenous localizers; back-translation of translated texts; adequate beta testing of localized programs; and so on.5

If "technical localization" is complex, expensive, and demanding, the second aspect of localization, which I will call "cultural localization," is even more complex, demanding, difficult even to define clearly, and largely unrecognized in the literature on localization and "user-friendliness".6 By cultural localization I mean the adaptation of programs written in one language by members of one culture to another language and another culture in such a way that they seem fully consistent with the assumptions, values, and outlooks of the second culture. As the European correspondent quoted earlier put it, a culturally localized program "should be indistinguishable from a program written by members of that culture."

Of course, if we view software as nothing but a culturally-neutral tool for solving universal problems, then cultural localization is a non-issue, like addition and subtraction. Indeed, there is clearly a "gradient of culture" that runs from relatively technical and universal programs at one extreme to programs laden with cultural content at the other extreme. Thus, programs for mathematical and statistical operations, like programs in basic science, are likely to be relatively constant regardless of culture. At the opposite extreme, management support systems, educational programs, and accounting programs, which must adjust for very different assumptions in different cultures, are what Claude Pesquet of DEC terms "more prone to cultural misfit".7

But even in the case of something that would seem so fundamental as an operating system, "cultural" differences -- specifically, assumptions on the part of program designers as to the capabilities and desires of users -- clearly make a large difference in the program that results. Anyone who doubts that "technology carries culture" need only examine the different "cultural" values and assumptions about users embedded in the early versions of two major competing American operating systems, initially implemented by Microsoft (DOS) and MacIntosh (Apple). These are, of course, not "international" cultural differences, but they point to the central role, in this case, of the assumptions of programmers about users in developing a complex computer operating system.

Not only is cultural localization complex, problematic, difficult to define, and largely unstudied, but it could well turn out that the negative evaluation of U.S.-written programs on "cultural" or "political" grounds will emerge as a more potent reason for not buying or not using these programs than any success or failure of technical localization.

Some "Cultural" Examples

Examples of cultural localization may help with an initial definition of the problem.

1. About a decade ago, a small Central American country embarked on a program of computerizing its primary schools.8 Requests for proposals were sent to major computer manufacturers. The proposal submitted by IBM, using LOGO in collaboration with its author, Seymour Papert of the Massachusetts Institute of Technology, was accepted because it addressed the concern that programs imported from North America would increase the already large dominance of North American culture in this nation. LOGO is of course a programming language; it was adopted because it was judged by local educational authorities to be "culture free".

Now, ten years later, that same nation faces new problems of how to avoid "North-Americanization" as it begins to develop connections for its high schools with the largely English-based Internet and Web. To be sure, educational authorities in this nation often find it difficult to articulate precisely the "North American" values they see contained in the software or information they do not wish to import. But they stress their own national traditions of collaboration, of peaceful democracy, of caution, and of cooperative problem-solving as values that are poorly expressed or even subverted in some North American software. Equally important, of course, only a few students speak fluent English, but most Internet and Web resources, including the major search engines, presuppose the ability to read and write English.

2. The newsletter, "Information Technologies in Developing Countries", covers a wide range of issues relevant to the localization of packaged software. Many of these are studies of key industries in developing nations or discussions of the technical or educational requirements for software development in the less developed countries. It is nevertheless noteworthy that cultural concerns recur in this technical context. For example, at the recent International Federation of Information Processing (IFIP) world congress in Canberra, a panel of members from Vietnam, Australia, India, Chile, and Barbados attempted to identify key issues in the use of emerging information technologies in developing countries. The participants expressed several anxieties about current trends in information technologies; e.g., that they will "widen the north-south gap further and inhibit the developing countries from integrating into the global economy", that direct foreign investment to create information infrastructure might provide an avenue for "cultural invasion through such highways", that in the past "infrastructure meant for social purposes like education was eventually hijacked by the entertainment industry", and "the information highways may widen the gap between the rich and poor segments within a developing country...."[9] Another discussion of trainers of information analysts in Africa, organized by the University of Nairobi in cooperation with IIM Ahmadabad, expressed the concern that the "Information Super Highway (ISH) might therefore turn out to be another media for cultural imperialism if care is not taken."[10]

These apprehensions relate above all to the emerging information technologies like Internet and Web, but they also apply to software. The developing nations have long and sometimes unhappy experience with imported technologies that promised to alleviate local economic and cultural problems, but that in the end perpetuated or increased the dominance of foreign economies and cultures.

3. A British correspondent comments that since the Bangemann report in 1994, the European Commission has become "very worried" about (American-) English-language cultural hegemony, particularly in the field of information technology. He reports a number of EC-funded initiatives to try to reinforce the linguistic and cultural diversity of Europe, e.g., that the Council of Ministers of the European Nations is shortly to be asked to endorse a new program for a "Multilingual Information Society". Today, for example, France, always protective of the historic integrity of its language, culture, and industry, insists that imported software be fully localized for France. European users, programmers, and purchasers are understandably concerned about American dominance in information technology, and are actively and officially seeking ways of counteracting it.

4. In 1992, when it opened its Beijing office, Microsoft introduced Chinese-language programs that used the Mandarin character set used in pre-revolutionary China and still used today in Taiwan. But the government of the People's Republic of China (PRC), after assuming power in 1949, had modified these traditional character sets, introducing simplified characters now universally used in the PRC. PRC authorities were therefore predictably offended by Microsoft's practice that programmers in Redmond, Washington, rather than local agents, define character set standards.11 The PRC's attitude to Microsoft allegedly softened only after Microsoft president Bill Gates spent an extended period in China and developed a series of multi-million dollar collaborative projects to train technicians at Chinese centers and universities, to develop in China Chinese-language operating systems, and to work with Chinese researchers on interactive TV, speech, and handwriting recognition programs.12 The discovery, in the summer of 1996, that some Microsoft programs localized in Chinese carried hidden anti-regime slogans has again strained Microsoft-PRC relations.13

The Microsoft episode illustrates the difficulty in separating technical, cultural, and political issues: what from one point of view seems a narrowly technical decision -- namely the choice in the United States of the character set used in a Chinese-language version of an operating system -- had major cultural and political meanings to Chinese authorities, and resulted for a time in a virtual ban on the official import of Microsoft products. (During this time, however, pirated Chinese- or English-language Microsoft systems were widely used in China, the world's largest potential market for software.)

5. The Argentine province of Mendoza, on the western border of Argentina with Chile, recently began an ambitious project of computerizing the elementary schools in the province. Operating through the provincial Ministry of Culture, Science and Technology, a vigorous Educational Computing Center was developed, including extensive teacher training and a software evaluation team. Eventually a North-American educational software suite was adopted for use in the schools. But although well localized from a technical point of view, this suite did not reflect what some educational authorities and teachers called "Argentine realities". In part, this complaint referred to the unavoidable fact that an educational program developed in one of the southern United States could hardly mirror the geophysical and human realities of an arid, prosperous province that lies in the shadow of the 23,000-foot peaks of the Andes. But educators also complained that the software was "too individualistic": for example, it presupposed that each student would sit alone solving problems as efficiently as possible, rather than collaborating with other students -- a preferred mode of problem-solving in many Mendozan primary schools. Both the configuration of the hardware (i.e., one keyboard per computer) and that of the software militated against what some Mendozan educators would have preferred as a "more Argentine" solution.

Partly in response to such problems, authorities in Mendoza proposed to develop a local multimedia development center, in collaboration with several departments in two local universities, the Director-General of Education, the Ministry of Culture, Science and Technology, and colleagues from MIT -- all working at a new Center for Technological Innovation. In the end, this project stalled because the parties involved have had difficulties in agreeing upon a single set of clear priorities. But even in the absence of such a multimedia development center, young programmers in the Ministry of Culture, Science and Technology have begun to develop multimedia software that deals specifically with the environmental problems of the Mendoza region.

The general point is obvious: the conviction that American software does not correspond with local values and needs constitutes a strong incentive to develop indigenous software and, conversely, not to introduce even technically well-localized American software. Perhaps there is a second point as well -- creating local multimedia-software development centers is difficult, even in a relatively prosperous, highly educated and sophisticated region like Mendoza.

6. Another example, as described by Asadi-Khomami, is the French hypertext system called LYRE. Asadi-Khomami writes that LYRE was successfully used to teach poetry in [France]. Student could see and analyze a poem from various viewpoints, but they were not allowed to add new viewpoints since that capability was reserved for the teacher. Limiting the student in the framework set up by the teacher is socially acceptable in France, and indeed an alternative design might be unacceptable because it undermines the teacher's authority. This limitation was not socially acceptable in the Scandinavian countries because the students' potential for independent discovery is limited.14

7. A recent recommendation that English be taught in all Japanese elementary schools has ignited a debate over the role of American culture in Japanese life. While most teachers, parents, and students are said to applaud the early development of proficiency in English, not everyone agrees. Among the dissidents is Yukio Tsuda, Professor of International Communications at the Graduate School of International Development at Nagoya University, and author of Invasion of English, Counter Attack of Japanese. Arguing that accepting English is tantamount to accepting the rule of the stronger, he warns that "global English language mass media like CNN and the English-dominated Internet help spread the U.S. way of thinking, adversely affecting Japanese culture and tradition."15

Professor Tsuda's fear of American cultural dominance, while perhaps a minority view, is by no means unique. In every nation, Internet, CNN, the Web, and American information technology are seen by some, both conservatives and radicals, as potentially subversive of the deeper, more important values of the local culture. How widespread such views are, and how influential a role they play in information technology policy, are among the matters that need to be studied.

8. An MIT colleague studying the deployment of software in American firms tells of a commercial packaged program designed by programmers determined to provide maximum customizability to individual users in corporate settings. Inspired by a "democratic" and decentralizing ethos, the resulting packaged software was strongly resisted by some corporate information system managers and general managers, who saw the possibility of customization by individual employees within the firm as subversive of management control and authority.

These examples illustrate some of the issues involved in any discussion of "cultural localization". They suggest the importance of awareness of these issues and the sensitivity of other cultures to the embedded presuppositions of U.S.-based information technologies. The Microsoft example is perhaps the most dramatic: the world's leading software company risked exclusion from the world's largest potential market because of a "cultural" failure of understanding. It may be, too, that the assumption that international standards should be set by programmers in Redmond, Washington reflects an outlook that could ultimately influence negatively the place of American software in the world.

Having outlined some facets of what I am calling "cultural localization", I now want to try to analyze that problem into a series of more discrete topics. Central is the variety of groups and subcultures in any society. It is of course often useful to speak of nations as if they had unitary cultures, and as if these cultures had more or less consistent, identifiable views about, for example, "American culture" or "computers". But in fact, this simplifying assumption is invariably incorrect. Any detailed study of the reception of American software in other nations must take into account: 1) the variety of American culture -- its division into multiple subcultures; 2) the complexity and "subculturation" of most nations of sufficient size and complexity to be purchasing software; 3) the complex relationships between those who use software and those who make decisions about its purchase; and 4) the universal fact of ambivalence -- simultaneously holding both positive and negative views -- about American culture, about computers, and about the role that computers should play in a society. Unless these factors are considered in detail, the result will be useless generalizations about pro- and anti-Americanism in the field of software.

Culture, Sub-culture, and Hacker Culture

The study of national character and national culture, popular among social scientists three decades ago, has languished in recent years. In part this is because some of the more extravagant claims about the national character and shared culture of Americans, Russians, or Frenchmen have turned about to be empirically incorrect, unverifiable, or oversimplified. Then, too, a famous article encouraging greater methodological sophistication in the study of national character,16 although intended to advance this study, imposed such high intellectual hurdles that it had the opposite effect. Perhaps most important, in recent years, pluralism and multicultural diversity have been stressed by social scientists more than national or cultural unity, partly because earlier studies of national character and national culture often turned out on closer examination to be studies of a dominant male minority. Today, as a result, studies of cultural diversity and of the differences within societies have largely replaced studies of the commonalities of national character and national culture.

But despite these shifts, observers continue to note modal differences between cultures and, specifically, between the values held by the average member of those cultures. So, too, they find these differences important in understanding reactions to the creation of software and its acceptance. Especially important is the work of Hofstede, who distinguishes five major axes of cultural difference that affect work values.17 Hofstede finds, for example, that of all the nations studied, America is highest on the value of individualism -- a finding consistent with the complaint of non-American users of American packaged programs that they presuppose "solitary problem solvers".

Hofstede's work to the contrary not withstanding, modern critics are of course correct to emphasize that almost every modern society is complex both culturally and characterologically. This point is especially relevant in considering the cultural origins of American software, which is not written by modal Americans but by members of a particular sub-culture of programmers whose characteristics are said to be those of "hackers", as Erran Carmel notes in an insightful analysis of the factors underlying American hegemony in packaged software.18

Carmel singles out several features of this hacker subculture -- its high degree of individualism, embodied in the image of the admired computer "wizard"; rebelliousness and lack of respect for established authority and tradition; enormous respect for achievement in the realm of software development as contrasted with lack of respect for ascribed characteristics like college degrees, social class status, and so on. He further notes the highly entrepreneurial culture of American programming, with its emphasis on competitiveness, risk taking, independence and creativity. Finally, he stresses the emphasis on innovation and rapid prototyping, the lack of dependence on established models, and the diversity of national and cultural origins of those who form part of, and excel in, the American software culture. Carmel notes that this sub-culture is in no sense ethnocentric, and that many of its leading figures or active participants are foreign born or recent immigrants drawn to the United States because of high salaries and because of the greater opportunities available in this American "hacker" sub-culture.19

The characteristics of "hacker culture" are of course also those that, in general, also differentiate American culture from the culture of other nations. Individualism, risk-taking, rebelliousness, irreverence, lack of respect for tradition, hierarchy and authority, emphasis on achievement rather than ascribed characteristics, positive valuation of novelty -- all have been attributed to American culture as a whole. We may therefore think of the culture of programmers as being "hyper-American" -- i.e., as embodying to an exaggerated degree the qualities which tend to differentiate Americans as a group from other cultures. If packaged software were written by Spanish-Americans in New Mexico, by New England Quakers, by Southern Baptist Fundamentalists, or by the members of any of a dozen other North American sub-cultures, we might find different modal characteristics and values.

Two points regarding localization follow from these observations. The first is that in so far as packaged software written in the United States embodies specific values, they are likely to be the values of the programmers, the "hackers", who collectively author the programs. It is unlikely that American-written programs would reflect values not embodied in that programmers' sub-culture. Thus, individualism, risk-taking, irreverence, absence of hierarchy, informality, etc., etc., are likely to be the dominant implicit cultural values of American packaged software programmers.

Secondly, what is true of America is also true of other major nations. Most nations today are multicultural and multilingual; many, like India, to the point that no single vernacular language is spoken by a majority. Even an apparently homogeneous nation like France is in fact complexly divided. To be sure, most Frenchmen tend to share, on average, certain outlooks and personal characteristics. But they also form part of diverse sub-cultures, some of them regional (i.e., Bretons versus Parisians versus Provencals), others occupational or class-related (the grande bourgeoisie as contrasted with cadres or the traditional working class), and so on. Since cultural localization entails adaptations of foreign software to try to make it indistinguishable from software written by members of the culture, but since individuals belong to sub-cultures rather than global cultures, localization to a "single" language and nation could in fact require multiple versions, one for each subculture.

Internationalization, "Americanization", and Local Cultures

In his recent book, Jihad versus McWorld,20 Benjamin Barber argues that the world is witnessing two contradictory but interacting trends. On the one hand, we are seeing the progressive multi-nationalization of national economies and the emergence of a worldwide popular culture symbolized by such phenomena as Coca Cola, McDonald's, MTV, Disney, CNN, and the popularity of mostly American films, songs, stars, and TV programs. With the digitization of world communications, decisions made in Hong Kong affect prices in London and San Francisco; events in Bosnia are instantaneously transmitted to Johannesburg and Bangkok; e-mail from Tokyo arrives in a few seconds in Cambridge, Massachusetts; Michael Jackson's marriage is discussed in Moscow and Buenos Aires.

For the small but influential minority of the world's citizens who are plugged into the electronic world networks of communication, who usually speak English, who travel routinely by jet from continent to continent -- for this minority, a new, syncretic world culture, closely related to but not identical with American culture, is emerging. This is the new culture of international corporations, international banking transactions, international entertainment companies, and internationally-oriented universities. Arguably its members constitute, perhaps for the first time in world history, a new worldwide ruling class. For purposes of convenience, we can think of the identifying symptoms of membership in this class as the regular use of a cellular telephone and the possession of an Internet address.

Elsewhere on the spectrum are 99 percent of the world's six billion people, organized into the myriad national, local, at times tribal, cultures that have persisted or re-emerged in recent years, Barber argues, partly in reaction to the flattening and uniformization of this new, cosmopolitan international culture. As nation-states become less important vis-a-vis the multinational world political economy, as a commercial cosmopolitan world culture mixes CNN with Disney and, at the high culture end, the "Three Tenors" at the Baths of Caracalla on CD and educational TV, as an international audience develops a common MTV culture among the young -- strong local counter-reactions also develop. In every nation, some reactions entail fundamentalism -- emphatic reassertion of older ethnic, national, parochial, or tribal identities, affirmation of ancient territorial and/or cultural claims, insistence on historical (or invented) values as a counterweight to the "materialism", "vulgarity", "shallowness", or "obscenity" of the emerging cosmopolitan culture, mistrust bordering on xenophobia vis-a-vis aliens, foreigners, and immigrants; protectionism in the international economic realm. Paradoxically, Barber argues, the ever-greater power of the electronic world culture and the multinational economy potentiates the reactions of localism, tribalism, regionalism, and fundamentalist affirmation of ancient (or nostalgically imagined) identities seen as threatened by this "new world order".

Attitudes toward the widening use of computers, the new electronic networks, and packaged American software systems are related to this international-cosmopolitan versus local-tribal spectrum. To those with cellular telephones and Internet addresses -- the new multinational elite -- "Americanization" may be more desired than resisted. One response to American "cultural hegemony" is to learn English fluently and use American computer programs; to study if possible at an American university or business school; to develop active correspondence via e-mail with other English-speaking correspondents. From this perspective, "localization" is at best a step necessary for children or for adults who do not (yet) speak proper English, and at worst a retrograde move that could prevent the society from "entering the modern world". Especially in nations and subcultures dominated by an ideology of "free trade", "liberalization", and internationalization of culture and commerce, opposition to American software is likely to be found less often among economic and political elites, and more often among those who resist the submergence of real or imagined local cultures in what is seen as "Americanization".

In terms of this cosmopolitan-local spectrum, opposition to computerization in general and American-based software in particular is not limited to any one point on the left-right political spectrum. But fundamentalists and "tribal reactionaries" who reject modernity and internationalization in the name of allegedly more "spiritual" or "ancient" values are likely to be vehement in their opposition to networked computers, software, new media, and electronic networks perceived as carrying cosmopolitan messages and themes. The cultural content of software, games, e-mail, Internet, and the Web are perceived as subversive to traditional spiritual values. In locations as diverse as fundamentalist Iran and modern America, critics emphasize the allegedly "pornographic" content of Internet communications and the World Wide Web. Societies and subcultures dominated by traditional, fundamentalist, and/or authoritarian values argue that free access to Internet and the Web undermines traditional or public values, is subversive of official power or national unity, or can be used by renegades, criminals, and revolutionaries for illegal purposes.21

Pro- and Anti-Americanism

A closely related but analytically distinct spectrum is the extent to which American influences are seen as noxious or benign. In Central America, for example, repeated American interventions into the political and economic life of that region have created, especially among intellectuals, a generalized mistrust of "North-American imperialism" that is in no way associated with right-wing anti-modernism or fundamentalism. Individuals with these outlooks may be enthusiastic proponents of the computerization of their society, especially of the schools. But they are also eager to avoid the hegemony of North-American cultural outlooks. Similarly, in Europe, intellectuals of the Left are often highly critical of the influences of American culture as seen in television, film, journalism, Internet, World Wide Web, and software, even as they may encourage the development of local computer, software, and telecommunications industries (often at some expense in terms of international trade balances). Although opposed to fundamentalism, left anti-Americans share with fundamentalists their objections to the superficiality, vulgarity, and materialism of the international popular culture they view as emanating from America and as threatening the indigenous culture of their societies.

At the opposite extreme are those who welcome American culture, ideas, electronic networks (and packaged software) precisely because they are American and precisely because their widespread adoption could help transform a society perceived as lethargic, tradition-bound, conservative and unwilling to adapt to the modern, multinational, multicultural world. Such advocates feel that if competitive, individualistic, risk-taking, and materialistic values are implicit in American computer systems, then they should be diffused as widely as possible so as to help awaken the society from its somnolence. "Modernization" is desirable, and if American software can be an agent in that transformation, then the use of American software, whether localized or in English, should be encouraged.

But finally, the near-universality of ambivalence about American culture needs to be stressed. The same French intellectuals who deplore the Americanization of French culture, who loathe French Disneyland and despise Parisian McDonald's, are often ardent fans of American literature and of aging American rock stars. The same Central

Americans who opted for LOGO because it was thought to be free of North-American culture now propose that all children in their nation start learning English in the second grade. The leaders of authoritarian countries who deplore the counter-revolutionary influences of American culture try to see that their children receive American graduate educations. One of the paradoxes of opposition to the introduction of American information and communication technologies is that those who have the knowledge and power to oppose them are so often members of the very cosmopolitan communities in which the use of American systems is virtually mandatory. Study of the cultural meanings of localization must deal with this ambivalence.

Protectionism, Copyright, and Free Trade

Another central determinant of attitudes toward American software is obviously economic. The legal import, purchase, and licensing of American software, like the use of American search engines for Internet and the World Wide Web, by non-American nations costs dollars.22 From the United States' point of view of course, it is a favorable item in a generally unfavorable balance of trade. For other nations, however, it is an unreimbursed expense. Every nation is therefore motivated, to greater or lesser degree, to try to minimize the cost of importing American software, to develop whenever possible indigenous centers for the creation of everything from programming languages to applications, and at times to close its eyes to pirated (i.e., unlicensed, "free") versions of American information systems.

With the issue of pirated software, we enter the vexed arena of international software copyrights and intellectual property rights, far too complex to be dealt with here. Suffice it to note the obvious: national compliance with copyright agreements varies enormously, with the most publicized negative extreme being the PRC, which has allegedly tolerated the pirating of books, CDs, CD-ROMs, and software. But given the cost of legally importing software and the technical ease with which it can be copied, it is easy to predict that in many nations, pirating software will become a popular, remunerative cottage industry. The responses of American firms are likely to be complex: on the one hand, we can expect vigorous corporate objections to pirating, and corporate encouragement of U.S. government action to punish nations which permit it. But on the other hand, in some cases at least, we may see secret satisfaction with the fact that pirated software systems help establish large market shares in nations like China, thus guaranteeing future market domination and future revenues. From Microsoft's point of view, for example, it may be more rational economically in the long run to tolerate the widespread pirating of Windows 95 and Windows NT in the PRC than to encourage the legal purchase and licensing of MacIntosh systems in that nation.

The French position on the film industry in the Uruguay Rounds illustrates some of the issues involved in protectionism versus free market of culture, and the difficulty in disentangling commercial and cultural arguments. In Uruguay, the French insisted on limits on the importation of American films to France so as to protect the French film industry. In fact, as Barber shows, the overwhelming majority of the most popular films in France (as in every other nation surveyed) are American. In 1991, for example, eight out of the ten most popular films in France were American, led by "Dances with Wolves" (1), "Terminator 2" (2), and "Robin Hood" (3).23 In their desire to protect the French film industry, French negotiators were also joined by French intellectuals who deplored the mindlessness or violence of the American films which the French public preferred.

This fusion of commercial and intellectual interests illustrates the difficulty in disentangling economic from cultural arguments against imported software. No hypocrisy need be assumed on the part of those who argue on primarily cultural grounds: many of those who wanted to protect the viability of the French film industry did so not only to reinforce the French balance of trade, but because they believe in the cultural value of French films. A similar mixture of motives can surely be expected in objections to American software and American-English-based communication systems.

Hardware versus Software

It may initially seem clear just what is hardware and what is software: computers are "just machines", whereas software may contain cultural content. But as computer manufacturers have known for at least two decades, the design of computers -- from chips to keyboards to printers and monitors -- embodies assumptions about users' needs, values, working styles, etc., which must ultimately be termed as "cultural" as are distinctive styles of teaching or problem-solving. Indeed, the common Latin American (and Pacific-Asian) complaint that American software does not encourage collective consensus-building, a culturally valued approach, is a complaint as much about hardware as about software. For if hardware is designed to support one, and only one, input device at a time (keyboard, joy stick, mouse, or modem) then several individuals using the computer must necessarily take turns, "standing in line" as they input information, ideas, program steps, or proposed solutions. When computers are scarce -- as they often are in classrooms -- it is common to have two, three, or more students gathered around a single computer, "sharing" the keyboard. But given the configuration of the hardware, only one student at any given time can use it.

Resolving this apparently simple problem involves changes in both hardware and software. A project at MIT's Sloan School of Management is attempting to confront the problem of multiple inputs by a simultaneous modification of both hardware and software. Others believe that through networking, joint or collective problem-solving can be achieved without major modifications of hardware.

But in any case, the point is that hardware configurations send cultural messages, and these messages may be alien to users. One Japanese correspondent notes the difficulty some Japanese computer users have had in shifting from traditional Japanese -- a hand-written character-based, largely ideographic language that traditionally starts each page at the top right corner of a 20 x 20 grid -- to a keyboard-based language with a restricted and rigidly defined number of keystrokes.24 For those accustomed to typewriters, in contrast, the shift to a computer keyboard is relatively simple and "natural." (The difficulty, as the Japanese observer notes, is finding an adequate way of inputting Japanese characters into a computer except through the use of a standard U.S.-European keyboard and accompanying software.)

The general point is obvious: some or many objections to American-based software may in fact be objections to the physical configuration of computers themselves, organized around a standardized, uniform set of devices for inputting, processing, and outputting information. Just as objections to American software may be related to, or reflect, a more diffuse critique of American cultural imperialism or American economic dominance, so objections to software may be related to objections to the hardware configurations of virtually all computers.

English Speakers and Non-English Speakers

Among the languages of the world, English is the second most frequently used. Chinese, with perhaps 1.5 billion mainland and overseas Chinese, leads the pack by far.25 English, dominant in the United States, Great Britain, Australia, New Zealand, most of Canada, and parts of South Africa and the Caribbean, is a distant second with about one-third the Chinese usage. Spanish, with approximately 400 million fluent speakers, is a close third.26 Thereafter, the number of speakers per language grows smaller, with Hindi, French and Arabic following.

In nations where English is the dominant or "native" language, American or British English-language software presents relatively fewer problems. To be sure, localization from the United States to Great Britain requires minor spelling changes and adjustment of currencies and weights. Admittedly, too, users in New Zealand and Australia sometimes argue that American software is yet another expression of a cultural imperialism to which they object. But compared to larger problems of linguistic and cultural localization, these are minor issues.

These larger problems indeed arise in societies where only a minority speaks English, and where the interests of the English-speaking minority and the non-English speaking majority may differ. The "tigers," in particular Hong Kong and Singapore, are prominent examples on the Pacific Rim; but a far larger single group who speak fluent English as a second or third language is in India, where, with a population of more than 900 million, two or three percent (i.e., about 20 million people) speak fluent English in addition to one or more of the Indian languages. In India as in other former British colonies, this English-speaking minority is distinguished by greater affluence, cosmopolitanism, social influence, political power, and access to the international commercial and cultural worlds. It provides the skilled computer programmers and designers who have made India one of the Third World's leading software producers, usually working in partly- or wholly-owned subsidiaries of American firms.27 In Russia, too, another substantial group of English-speakers exists; Russia today is one of the ten nations with the largest number of Internet addresses in the world.28

In a nation with an English-speaking minority and non-English- speaking majorities (as in India, where Hindi is the minority national language and there are many additional major linguistic groups and dozens of less-frequently-used languages), the English-speaking minority and the others may have divergent, or even opposed, interests. For the

English-speaking Indian or Russian, "localization" may be neither necessary nor desirable. Indeed, both India and Russia provide American software firms with large numbers of gifted programmers who write parts of "American" programs in their native countries, transmit them to American software firms, and are paid wages that may be high by local standards but are fractional by American standards. For such programmers, English is essential and localization is unnecessary. But for the great majority of Russian-, Kannada-, Armenian-, Chechen-, Urdu-, Hindi-, Malayam-, Bengali-, Kazakh-, etc., speaking citizens of Russia or India, localization is essential, and lacking a knowledge of English, their electronic participation in the cosmopolitan world culture is currently minimal. As a result, insofar as the non-English-speaking population of most countries has formed any opinions at all about U.S.-written software, or Internet and the Web (and these opinions will often be apparent vis-a-vis educational programs), their opinions can be very different from the opinions of more cosmopolitan English-speaking groups.

Users versus Purchasers

In the wealthy and so-called "free market" industrial nations, tens of thousands of advertising messages are addressed each month to potential purchaser-users to encourage them to buy a particular brand of mini-computer or a specific operating system, programming language, or application.29 But in highly industrialized nations, software sales are critically determined by corporate and governmental purchasers, whose decisions affect millions of institutional users. Institutional purchasing officials or software managers may rarely consult potential users, but they invariably consult those who finance the purchases of entire systems involving multiple users. In the United States, for example, with its highly decentralized educational system, decisions about educational software (and hardware) are usually made for school systems as a whole; and in major cities they can entail the centralized purchase of thousands of copies of operating systems and educational software packages.

In societies less wealthy, smaller, and/or centrally-planned economically, few if any individuals may be able or allowed to purchase "personal" computers according to their own preferences.30 In such circumstances, the purchaser-user distinction may be large, for purchasing decisions by central authorities may be guided by principles very different from those which would have guided users. In most developing nations, government and corporate decision-makers about software purchases tend to be closer to the cosmopolitan-multinational world described above;31 but users, if they are railroad employees, bank tellers, local teachers, or students, are more likely to be "locals" -- culturally and linguistically far from the international world of telecommunications, rarely fluent in European "colonial" languages like English, French, or Russian, more often attached to traditional ways of thinking. Furthermore, decision-makers in the area of telecommunications, electronics and software may be especially open to the influences (including financial rewards) of international corporations desirous of their business. This susceptibility can lead to greater responsiveness to the interests of multinational corporations than to those of local users.

Finally, in multicultural or multilinguistic nations, software purchasing decisions made at a national (or even a provincial) level may run counter to, or be in deliberate contravention of, local cultures. In the name of national unity, central governments in multilinguistic, multicultural nations may desire to impose a common language (Hindi, French, Swahili, Hausa, Arabic, Russian) and a common culture on a linguistically and culturally diverse population. Software purchasing decisions may be intended to push computer users to use programs "localized" in a language other than their vernacular language. Whether this is judged progress in the interest of national unity or subversion of regional and ethnic subcultures is invariably a matter of controversy. But it should be noted that conflicts between national and regional languages and cultures are not limited to developing nations: Breton and Catalan nationalists are as eager to see Breton and Catalan versions of French educational materials as are Xhosha-speaking South Africans or members of "scheduled" tribes in India.

To add further to the complexity of this issue, educational or banking software decisions in many nations often involve at least three distinct levels: national or provincial authorities who make the fundamental purchasing decisions; teachers, bank tellers, or salespeople at railroad or airline offices at the local level who teach from or work with the software; and finally, students, clients, or passengers for whose benefit it is intended. Needless to say, each of these three groups may have distinct interests, distinct cultures, and distinct attitudes toward any given software system. All of this makes it impossible to decide a priori for whom the software "should" be localized.

Is Localization Always Desirable?

Throughout these comments, I have insisted on a broad distinction between cosmopolitan and local orientations, a distinction that seems useful in many nations, whether industrialized or developing. I have further largely assumed that some kind of "localization" -- both technical and cultural -- is desirable. In part, it obviously seems desirable economically to American firms seeking to sell their packaged programs abroad to clients who will not purchase them unless they are adequately translated and adapted to local culture. In part, too, it may be desirable because the preservation of cultural diversity through localization -- both technical and cultural -- keeps the world a richer, more varied place, limits economic and cultural imperialism, and promotes a sense of national, local or regional identity which some observers consider desirable both for individual self-esteem and ultimately for world harmony.

Yet we also need to ask whether there are limits to localization. In the United States, an analogous issue is teaching sub-cultural dialects or languages to students in American schools. On the one hand, it is claimed that people who speak "non-standard" dialects or languages -- e.g., African-Americans or Hispanic-Americans -- should be schooled in these languages where they have no disadvantage, where they can draw on their own familial linguistic patterns, ethnic histories, and cultural traditions, and where they are not implicitly obliged to acknowledge a kind of implicit cultural inferiority vis-a-vis the prevailing "Anglo" culture. But on the other hand, critics claim that American children who are taught to speak and write only Black English or Mexican-Puerto Rican Spanish will, as adults, be at an enormous disadvantage occupationally, culturally, and socially in the United States, where the lingua franca is "Anglo" English. They claim it is essential that, without denigrating the value and importance of African-American or Spanish-American traditions, students from these backgrounds be taught to excel in standard English.

Parallel questions can be raised with regard to software localization in other nations. At the extreme, might it not be better for all children to be taught programs that use English or some other international-colonial-imperial language like Chinese, Spanish, Arabic, Russian, or French? Knowing these languages, with software as one part of the instrumentation of pedagogy, would truly equip students to enter the modern, multinational electronic world as full participants. To be entirely educated in Wolof, Quechua, Xhosha, or Taghik -- whatever its merits in terms of national pride and identity -- could produce a generation of students disqualified for work and participation in the most vital sectors of the international economy and culture.

To state the choices in terms of stark oppositions is doubtless to overstate them. In fact, as Laitin argues, the frequent pattern in many multicultural, multilingual developing nations is for students to be familiar with "three plus or minus one" languages: their native or vernacular language, the national language, and the former colonial language -- which may remain, as in India, the official language of the higher civil service, the courts, and the legislature.

Thus, educated Indians who live in Bangalore need three languages: Kannada, Hindi, and English; Indians whose vernacular language is Hindi, the "national" language, need only learn English in addition (three minus one), whereas Tamils who migrate to Calcutta have Tamil as their vernacular, need Bengali as a second vernacular, and must learn Hindi and English as national and official languages (three plus one). In most of Sub-Saharan Africa, Laitin32 argues, the language of the former colonial power retains its hold as an official language of the upper civil service, the courts, and in some cases, of government and business. Only a few countries, like Tanzania, have a single language (Swahili) that effectively is adopted as vernacular, national, and official language. More common is the pattern of, say, Kenya, where for highly educated Kenyans, facility in Luo or Kikuyu, Swahili, and English may be commonplace.

In short, monolingual societies are rare, the products of special historical circumstances like the linguistic unification of European nations and Japan in the 15th-20th centuries, the effective destruction of native languages by colonists in nations like Australia and the United States, or the special circumstances described by Laitin that account for the adoption of Swahili as the single language for Tanzania.

Multilingualism vastly complicates the issue of localization, inasmuch as it requires localizers and purchasers to decide which of the various languages used in any given area is the language -- or the languages -- to which software should be localized. But in any case, we should not simply assume that everyone in the world "needs" software localized to his or her vernacular language. Other choices may be more productive, more useful, more desired. Above all, when computer use can open up extraordinary and instantaneous access to the vast information resources in the rest of the world through Internet and World Wide Web, the best localization choice may be the language that provides easiest access to these information resources. Indeed, as I note below, the advent of Internet and the World Wide Web vastly complicates the whole issue of software localization.

The Special Problems of the Net and the Web

In the preceding comments, I have focused on the problem of localization of packaged software from one culture to another, i.e., "one-way" localization from A to B. I have tried to show that this issue is enormously complicated, given the variety and complexity of cultures and sub-cultures in the world. But the study of cultural localization from A to B at least has the virtue of relative simplicity, inasmuch, in any given case, we are dealing with only two cultures or sub-cultures.

With the phenomenal growth of the Internet and the World Wide Web in the last years, the problems of cultural localization are vastly compounded, unless we foresee the de facto or mandatory adoption of a universal lingua franca (English) for all world electronic communications. For with Internet and Web exchanges between many cultures, not only must A communicate with B, but B with A; and as we add additional languages and cultures to the instantaneous and simultaneous communications of the Internet and the World Wide Web, the number of possible inter-linguistic and inter-cultural communications soars geometrically. Assume, for example, that Internet-Web communications are supported in "only" eight major international languages: Chinese, English, Spanish, French, Russian, Arabic, Hindi, and Swahili. Already, we have 8 x 8 or 64 needed simultaneous linguistic and cultural localizations, instead of the single localization involved in the internationalization of, for example, packaged American software to another language-culture.

As the freestanding computer gives way to the networked computer linked not only to a local network and server, but to the entire world -- and to millions of other networks and computer users -- via the Internet and Web, issues of linguistic and cultural localization, already complex in dealing with localization from A to B -- have become vastly and even unmanageably more complicated. As more of the world's computer users become interconnected, assuming the pattern of technological dispersion that characterized utilities like the refrigerator and the automobile as they spread to spread to the growing middle classes of the developing nations, issues of localization will become ever more important.

At present, the Net/Web internationalization problem is minimized by the fact that almost all of those who have access to these resources are English speakers. As noted above, 87 percent of Internet addresses are currently located in four predominantly English-speaking countries, the United States, Great Britain, Australia, and Canada. It is almost certain that most of the remaining 13 percent know enough English to find their way around in this predominantly English-language world. In India, South Africa, Nigeria, Singapore, and Argentina, Net/Web users are usually part of that cosmopolitan elite for whom knowledge of English -- usually as a second or third language -- is more or less taken for granted. For the moment, then, English is established as the de facto lingua franca of international telecommunications, and the linguistic-cultural problem is not acute.33

But in the future it may well become pressing. The case of Costa Rica may be predictive. That small nation (population 3,000,000) has an advanced telecommunication structure, involving fiber-optic links between banks in major cities. Work is underway to network secondary schools in the country to each other and to the Internet and World Wide Web by using the currently under-utilized fiber-optic system of the banks. When this technology is operative, the problem of "localization" -- and specifically, the English-language basis of Net/Web -- will become more acute. For only a few high school students in Costa Rica speak or write fluent English; those who do not will necessarily be limited to communicating with Net/Web sites in other Spanish-speaking countries. But as Guy de Teramond, the energetic head of CRNet, has shown, the number of Net addresses and Web sites in Spain and Spanish-speaking Latin America, although growing rapidly, remains small as compared to the number of English-language sites in the world.34

It could be argued that the Net/Web problem is largely linguistic, and that it can be solved by continuation of the present practice of using English as the lingua franca of international telecommunications. But this proposal, however plausible, has two problems. First, it means that the vast majority of the world's citizens, who do not and who will not in the foreseeable future command the English language, will be effectively excluded from the worldwide telecommunication network and the informational resources that this network puts at their disposal. On grounds of social justice, equity, and the development of the less developed nations, this is far from an ideal solution.

Secondly, however, even the universal use of the English language does not obviate problems analogous to those of cultural localization on the Internet and the World Wide Web. These problems fall into three overlapping categories, each of which deserves detailed exploration. Here they can only be mentioned. First, there is the problem of unacceptable cultural content on the Net/Web. In America and Western Europe, as in the developing nations, the fear that obscenity, pornography, anti-religious ideas, and culturally subversive outlooks will be disseminated freely on the Internet and the Web is widespread, and not without a certain reality. In every nation where Internet and Web are realities, questions of whether and how to censor, screen, or filter culturally unacceptable content, images, and messages is debated. At the extreme lie fundamentalist religious arguments, as present among American as among Iranian fundamentalists, that would ban or exclude large areas from Net/Web exchanges as a matter of public policy. At the other extreme are libertarians who argue that any control or censorship must be a matter of individual or familial decision. Nations like Singapore and the People's Republic of China are insistent on centralized control of Net/Web communications in order to block unacceptable cultural or political content. (Many of the technical problems in selectively blocking unacceptable cultural content but at the same time permitting communication with foreign servers from whom such content might be downloaded remain to be solved.)

The second, overlapping, problem analogous to the localization problem in software is that of unacceptable political content. Again in every nation, whether highly industrialized or developing, the fear that the new telecommunications networks will be used for subversion, crime, terrorism, espionage, revolution, or counter-revolution is important. Should cookbooks for making homemade bombs continue to be available to Americans on the Internet? Many think not. Should anti-regime and subversive messages be available on the Internet in the People's Republic of China? Political authorities believe not, and in order to prevent the use of the Internet and the Web for subversive and counter-revolutionary purposes, propose to channel all messages through central servers from which they can be monitored. Should drug dealers and money-launderers be permitted free use of the Internet and Web, encrypted beyond the capacity of law enforcement officers to decrypt? The official American government policy is that they should not.

Finally, the Internet and World Wide Web raise the question of unacceptable intrusions into what should be private or privileged communications. The threat of malicious hacking, of idle browsing, of private or public monitoring of what should be protected communications, of computer theft, and the vexed question of the appropriate limits to private and public encryption all arise once electronic communication becomes widespread.

Most of these problems are, at root, political and cultural, and in this regard, analogous to those raised by cultural localization. Portrayals of women that are generally acceptable in American culture may be obscene in Arabian culture; criticisms of public authorities that are normal in West European democracies may be considered subversive, criminal, or counter-revolutionary in authoritarian regimes; encryption that seems to individuals or commercial users necessary to safeguard the privacy of their messages may appear to governments as providing unacceptable protection to criminals, drug dealers, terrorists, and saboteurs. Standards of what is acceptable culturally, politically, and by way of intrusion vary enormously within cultures and across cultures. Each of these Net/Web problems has its analogy in the simpler issues of cultural localization.

The study of cultural localization thus provides a window on some of the issues that will become ever more urgent if, as is expected, the use of telectronic images are spread ever more rapidly and to an ever wider group through the Internet and the World Wide Web.35

The Future of Localization

The task of localization or internationalization currently involves many thousand people scattered in virtually every country of the world. From the "internationalizers" in major software firms to localizers in Kenya, Argentina, Malaysia, or Pakistan, a large industry has developed which, though largely centered in the industrialized nations, promises to extend to more and more countries as computer use increases and international communications grow more accessible.

The future prospects of localization currently fall somewhere between two equally unlikely extremes. At one extreme is the possibility that, over time, English will become the lingua franca of all international communications, that all computer users throughout the world will learn English, and that the approximately eight percent of the world's population that now speaks English will grow to 100 percent of the computer users in the world.

In the other unlikely extreme is the possibility that every one of the world's thousands of vernaculars will become the basis for computer use for those who speak that vernacular, requiring localization of major operating systems and programs into thousands of distinct languages, character sets, scrolling patterns, etc.36

Clearly, the actual future will lie somewhere between these two extremes. But a projection of current trends suggests that, in the years and decades ahead, localization will become more, not less, important.

The pattern of diffusion of new technologies is such that they gradually move from their countries of origin to the middle classes of other countries of comparable outlook and growing wealth, achieving saturation first in the nation of origin, then approaching saturation in other countries, and finally moving into other less wealthy countries via growing middle classes. This pattern is observable with railroads, automobiles, refrigerators, radios, television, and a variety of other technologies. Many have first achieved saturation in countries like the United States (automobiles, radios, color television, etc.); or remain to be extended to most of the world's population. On January 6, 1997, the New York Times (page 1) reminded us that 50 percent of the world's population has never made a telephone call. But at present in the United States, 40 percent of households have a computer; whereas in the wealthier countries of Europe and Japan, the figure hovers between 10 and 30 percent depending on the nation. In developing nations, computer ownership is either largely limited to officialdom, in the case of centrally controlled economies, or to corporations, financial institutions, major academic centers, and a few wealthy or middle class citizens in the case of market oriented economies.

In January 1997, a base-level computer capable of running most available packaged software required a fast Pentium processor, 16 megabytes or more of RAM, a one-plus gigabyte hard drive, a 28.8 modem, and a 6x+ CD-ROM drive. In the United States, with careful shopping, such a machine can be purchased for plus or minus $2,000, complete with 15-inch monitor. It is thus within the financial reach of a majority of families in America and the wealthier countries of Europe, Canada, Australia, New Zealand, Singapore, Hong Kong, and Japan. But it is not within the reach of most households in the world, including most of Africa, India, large parts of Latin America, and the People's Republic of China.

In addition to their financial cost, computers have a high cost in learning, despite efforts by programmers to create "user-friendly" interfaces. Typing instructions on a keyboard, or using a mouse, pointer, or joy stick to click on desired options, is not a "natural" human activity; it requires literacy, which by no means can be assumed in many nations; it also requires a grasp of how a computer works, the ability to type or enter instructions from a keyboard, the ability to create and locate files, etc., etc. As compared, say, with learning to drive a car, learning to operate a computer is more complex, a task that is probably best learned gradually, and in childhood, much like basic literacy. Thus, even among those who can (or could) afford the financial cost of a computer, and no matter how low that cost may fall in the future, there will be a significant percentage who cannot or do not choose to pay the price in learning how to use the computer.

With all these caveats, the future possible spread of computer use, especially in nations where currently only a tiny fraction of the population has access to computers, depends most importantly on the economic development of the countries involved. Given growth rates like those that have characterized Japan, South Korea, Hong Kong, and Singapore in the last decades, other nations can expect a rapidly growing ability to purchase computers, starting with the growing middle class and extending deeper in the society.

Thus, given an optimistic scenario about increases in the real purchasing power of citizens -- or at least of the middle class -- in the developing nations, we can anticipate that computer use is likely to spread to ever larger numbers of people in the developing, as in the developed nations.

As this happens, it is unlikely to be accompanied both by facility in the use of computers and facility in knowledge of English. Consider again India, with a 1997 population of approximately 940 million people. Of these, perhaps 20 or 30 million speak good English and therefore could have easy access to unlocalized English-language programs, Internet, and the World Wide Web. But if we assume that India's enormous and growing middle class, whose size is now estimated at 100-200 million people, moves slowly toward computer ownership, it is almost inconceivable that they could all acquire the ability to operate computers effectively in English. India is a nation with 18 officially recognized languages, and 1600 minor languages and dialects listed in the 1991 census. The national language, Hindi, is spoken as a vernacular by 20-40 percent of the population, and the official language, English, is used in the upper civil service, courts, national legislature, and to some degree as a lingua franca, but is spoken well by only 20 or 30 million. Widespread computer use in India would require localization to at least 18 distinct languages, each largely unrelated to the others and, in many cases, with distinct character sets.

The Indian example suggests that if programs originally written in the United States or Western Europe are introduced into non-European societies, problems of localization are likely to become more prominent, and at the same time more difficult to solve both technically and culturally. There is, after all, a unity sometimes referred to as that of "western civilization" that joins nations as disparate as France, Germany, Australia, the U.S., Spain, Argentina, and Costa Rica. Basic linguistic structures and constructions of reality in one culture have analogs or parallels in the others. But as we move to countries with radically different languages, thought patterns, traditions, and cultures -- e.g., Japan, China, the diverse cultures of India, the thousands of languages and cultures of Africa, the many cultures of central and northern Asia, etc. -- the taken-for-granted similarities of Indo-European languages and cultures disappear. It is therefore predictable that cultural localization, as an aspect of the general problem of internationalization, will become more important.

This argument supposes that (obviously) the entire world is not about to learn fluent English, and that if and when computer use spreads to populations where it is now non-existent or rare, the demand for technical localization and the importance of cultural localization will increase. Linguistic localization alone will require an army of localizers to keep abreast of. Dealing with the cultural issues discussed here will become more important, although initially likely to take a second place to the more immediate and easily definable problems of translation. For the foreseeable future, then, localization is likely to remain a growth industry.

Research Orientations

Given concerns over cultural aspects of localization expressed by everyone from members of the European Commission to representatives of developing nations, it is surprising that virtually no research on the topic has been conducted. Especially lacking are cross-cultural studies that might clarify some of the questions discussed above.

Given the relative absence of research on this topic, quantitative or survey-type research would seem premature. Instead, ethnographic-anthropological-observational and case studies are likely to be more productive. For example, a study of three topics seems likely to produce a useful result.

1. Localization in Western and Non-Western Contexts. This study would entail an in-depth study of localization in two countries, one in the "Western" world and the other in the "non-Western" world, to examine the ways localized software is used, perceived, and reacted to and against. In both countries, discussions would take place with indigenous localizers, with purchasers of localized software, and with users. Specific questions would entail perceptions of cultural conflict or incongruity in software, the degree of user involvement and the structure of decision-making with regard to software purchases, the way local needs and realities are balanced with international considerations in software decisions, the role played by the English language in decisions of purchasers and users, and present plans with regard to Internet and Web use. One possibility might be in Argentina to focus on Mendoza, and in India to focus on Karnataka, as regions in both countries where "computerization" is relatively advanced.

2. Localization of a New Program, e.g., studied by establishing an alliance with a major American software producer like Lotus, Corel, Microsoft, etc.37 The strategy would be to study how American designers and programmers go about developing products for international sale, examining their relationships and interactions with indigenous localizers, studying the relationship of indigenous localizers to actual and potential clients, and exploring how local perceptions are fed back to and incorporated (or not incorporated) into the work of the designer team in the United States.

3. Literature Search and Researchers' Network. Preliminary exploration has demonstrated the existence of a significant number of researchers and professionals, some in the commercial world and others in the academic world, who express an interest in cultural localization and believe that its study is important. At present, no network for communication between these interested parties exists. Nor has there been an adequate bibliographic search for relevant literature, narratives, and empirical or case studies.

Two steps would therefore be useful. First, establish via Internet and eventually Web an intercommunicating list of key researchers and commercial actors in the study of cultural localization. Some should be academic researchers, chiefly anthropologists and psychologists, others, professionals involved with localization in software firms or in the branches of firms dedicated to internationalization. Second, an annotated bibliography of the most useful works, studies, and cases needs to be compiled and made available to interested researchers.

Outcomes

Several outcomes might be foreseen from research in this area. The first would be to develop models of cultural localization, defining more precisely the problems that arise with regard to cultural differences in software, exploring solutions and efforts at solutions that have been attempted, that have worked, and that have failed. Especially important in the long run will be studies of efforts to establish local centers of software development to counter the economic and cultural influence of software that emanates from the United States and Western Europe. When are such projects likely to be undertaken? When are they likely to succeed? How and in what areas can nations with relatively few resources in this area compete effectively with American software producers?

Second, building on the already-existing network of localizers organized through LISA, the project could hope to develop a more selective yet professionally broader network of professionals and researchers interested in cultural localization. Such a network could confront more directly issues of cultural and economic hegemony, problems of the selection and use of localized software in developing and developed nations, differences between cosmopolitan and local groups, etc. The issue of cultural localization should be put on the agenda of professional groups dealing with international software and the future of Internet and the World Wide Web.

Third, the project should stimulate publications both in scholarly journals and in trade journals. Access to trade journals is especially important, for the burden of localization falls not on the academic community, but on the software industry, and specifically on those in any software firm charged with internationalizing actual and future software products. Just as manuals currently exist to guide technical localization, so, in the future, guidelines for effective cultural localization might be developed and made available to the software industry.

Acknowledgements

Among many email correspondents who have been helpful, I especially want to thank Jan Freijser of Stream International; Claude Henri Pesquet of DEC; Eric Hyett formerly of Microsoft; Diana Forsythe of the University of California/San Francisco; Steven Lerman of the Center for Educational Computing Initiatives at MIT; Donald Day and his associates at the University of New South Wales; Bonnie Kaplan of Yale University; and Michael Anobile of LISA in Geneva, who suggested the names of several localizers with an interest in cultural aspects of internationalization.

In Argentina, I am especially indebted to Dora Balada, Director of the Center for Informatica Educativa in Mendoza; Maria Puerta de Gonzalez, former Minister of Culture, Science and Technology of Mendoza; Armando Bertranou, Rector of the National University of Cuyo; and Horacio Reggini of Buenos Aires.

In Costa Rica, I have profited from conversations with Guy de Teramond, Director of CRNet; Jeanina Umana Aguiar, Executive Director of the Omar Dengo Foundation; Clotilde Fonseca Quesada, Deputy Minister of Public Education; and Eleonora Badilla Saxe, Executive Director, Ministry of Public Education.

Although we have not corresponded, I have found Subhash Bhatnagar's editorials in the Newsletter of IFIP Working Group 9.4 especially thoughtful on the issue of computers in developing societies.

My friend Leo Marx has as always been helpful in his comments and criticisms of this draft. I am grateful to Michael Fischer and Michael Dertouzos for their encouragement to pursue this exploration.

NOTES

[1] Erran Carmel, "American Hegemony in Packaged Software Trade and the 'Culture of Software'", to appear in The Information Society, (1996) 12(4). Carmel estimates that the U.S. supplies about 75% of the world's packaged software, and accounts for about 45 percent of world demand.

[2] David Brake, "The U.S. Wide Web," Salon (Apple on-line), Issue 30 (September 3-6, 1996).

[3] On the relative failure of European firms to produce internationally marketable software, see "Europe's Software Debacle," The Economist (November 12, 1994), pp. 77-78.

[4] Nadine Kano, Developing International Software for Windows 95 and Windows NT (Redmond, Washington: Microsoft Press, 1995). See also Tuoc V. Luong, James S.H. Lok, David J. Taylor, and Kevin Driscoll, Internationalization: Developing Software for Global Markets (New York: John Wiley & Sons, Inc., 1995), among many other works on localization and internationalization.

[5] It goes without saying that localization, if well done, is a lengthy and expensive process, involving for U.S.-written programs repeated, regular interactions between source code English-language programmers and "localizers", whether in another country, directly attached to the core programming team, or both. These costs of localization may make it economically unrealistic for small or niche programming firms to contemplate localization of their software.

[6] Kano's excellent work is typical in this regard: cultural localization is mentioned as an afterthought, with an exhortation that it needs to be considered. The literature on "user-friendliness" is especially striking for the absence of studies of "users", and for its focus on technical problems of programming. See, for example, Scott P. Robertson, Gary M. Olson, and Judith S. Olson (eds.), Human Factors in Computing Systems: Reaching through Technology -- CHI '91 Conference Proceedings, New Orleans (New York: Association for Computing Machinery, 1991). In a work of more than 500 pages, the only actual study of "users" occurs in a discussion of British and North American responses to the "Scandinavian challenge", an article that underlines the extreme infrequency, outside of

Scandinavia, of implementing the Scandinavian proposal that users actively participate in defining software design goals. The article mentions problems in "mutual accommodation between participatory design procedures/institutional culture", and questions the feasibility of "democratic design procedures in apparently 'hostile' environments". ("Participatory Design in Britain and North America: Responses to the 'Scandinavian Challenge'", op cit., pp. 389-392. An important and very useful exception to the usual neglect of user characteristics and culture is found in Elisa M. del Galdo and Jakob Nielsen (eds.), International User Interfaces (New York: Wiley, 1996), especially chapters 3,4,5 and 6, all of which focus on cultural aspects of software.

There are few social science studies of the cultural aspects of localization. But for important exceptions, see the work of Bonnie Kaplan e.g., "The Computer Prescription: Medical Computing, Public Policy, and Views of History", Science, Technology & Human Values (Winter 1995) 20(1): 5-38; Diana E. Forsythe, e.g., "Engineering Knowledge: The Construction of Knowledge in Artificial Intelligence", Social Studies of Science (1993) 23(3): 445-477, or "Blaming the User in Medical Informatics: The Cultural Nature of Scientific Practice", Knowledge and Society (1992) 9: 95-111; Lucy Suchman, "Do Categories have Politics?", Computer- Supported Cooperative Work (1994) 2: 177-190. International studies are especially rare; but see Kumiyo Nakakoji, "Beyond Language Translation: Crossing the Cultural Divide," IEEE Software (November, 1996), 42-46, "Crossing the Cultural Boundary," Byte (June, 1994), 107-109; chapter 6 in del Galdo and Nielsen, op cit.; and the project being initiated by Donald L. Day and colleagues at the University of South Wales, Australia: see "Cultural Bases and Interface Acceptance: Foundations" in N.A. Sasse, R.J. Cunningham, and R.L. Winder (eds.), People and Computers XI: Proceedings of HCI'96 (London: Springer-Verlag, August 20-23, 1996), 35-47, and Vanessa Evers and Donald Day, "The Role of Culture in Interface Acceptance," paper submitted to the INTERACT '97 Conference, Sidney, July 1997.

7. In a personal communication, Claude Pesquet contrasts management support systems with general ledger systems, and argues persuasively that the issue is whether or not there is a "correct answer". Throughout, Pesquet's comments have been invaluable. (Personal communication, 2 October, 1996.)

8. For an early account of this project, see Clotilde Fonseca Quesada, Computadoras en la Escuela Publica Costarricense: La Puesta en Marcha de una Decision (San Jose, Costa Rica: Fundacion Omar Dengo, 1991).

9. Subhash Bhatnagar, "Use of Emerging Information Technologies in Developing Countries: Key Issues," Newsletter of IFIP Working Group 9.4, October 1996, 6;4. Panel participants were of course not entirely negative, and also stressed the positive potentials of interactive distance learning, overcoming barriers of physical distance, etc. The tone of the panel as reported, however, is largely cautious.

10. "African Perspectives of the Global Information Superhighway", (no author listed), in Newsletter of IFIP Working Group 9.4, October 1996.

11. From Microsoft's point of view, it is of course essential to maintain control over localized versions of programs if later corrections and upgrades are to be incorporated into localized versions of Microsoft programs. Maintaining some reasonable balance between the demands of localization and the need to maintain the integrity and upgradeability of the program is perhaps the thorniest problem facing software firms that have an international, multi-lingual client group.

12. "Microsoft's Long March", Business Week, June 24, 1996, pp. 52-54.

13. South China Morning Post, (October XX, 1996).

14. Alireza Asadi-Khomami, "Reviewing Research in the Area of Software Localisation" (Computer Science Postgraduate Project 3), Department of Computer Science, University of Queensland, Australia, November 3, 1995.

15. Nikkei Weekly, July 22, 1996, p. 20.

16. Alex Inkeles and Daniel Levinson, "National Character," Handbook of Social Sciences [check reference]

17. G. Hofstede, Culture's Consequences: International Differences in Work-Related Values (Beverly Hills, CA: Sage Publications, 1980), and G. Hofstede, "Cultural Constraints in Management Theories," Academy of Management Executive (1993) 7(1): 73-94.

18. Carmel, op cit.

19. Another hacker characteristic, not discussed by Carmel, may be especially relevant to the question of cultural localization. As Sherry Turkle (The Second Self: Computers and the Human Spirit (New York: Simon & Schuster, 1984)) and others have noted, "hackers" as a group tend to be by temperament and training insensitive to affective and "cultural" cues. Inclined and trained to deal with technical problems that have more or less unequivocal solutions, they are less comfortable with, and tend to avoid, issues of affect, culture, values, and myths which do not lend themselves to unambiguous analyses and clear solutions. Thus, one reason why cultural localization has gone so relatively unstudied may have to do with a temperamental and subcultural conflict between hacker culture and the qualities requisite for sensitivity to "cultural" issues. When I proposed to a young computer scientist that a study of the cultural assumptions of software might be useful, for example, he replied emphatically, "Once you start examining your navel, there's no end to it. You might as well stop working altogether."

20. Benjamin R. Barber, Jihad versus McWorld (New York: Times Books, 1995). For a summary of his dissenting view, see Samuel Huntington, "The West: Unique, Not Universal," Foreign Affairs (November-December, 1996) 75(6): 28-46. On general issue of the internationalization or "convergence" of both economies and cultures, a useful overview is provided in a special issue of Sciences Humaines, "Vers la convergence des sociétés?" (September-October 1996), 14. That issue adds many qualifications to the unqualified claims of total economic multinationalization of the economy and global Americanization of culture.

21. In 1996, Singapore and the People's Republic of China announced their intention of having electronic communications passed through central servers from which obscene, culturally inappropriate, or anti-regime communications could be removed. Some of the same objections, of course, can be made to telephonic communications, but the latter are more easily monitored without recourse to complex electronic decoding devices.

Furthermore, Internet and Web communication, like fax communication, permits one individual to send the same message or image to hundreds or thousands of computers at a keystroke. Finally, it is often possible to circumvent Net/Web censorship by downloading forbidden messages or images from "acceptable" sites to which they have been sent.

22. Unless, of course, the software is pirated. Estimates of the percentage of software that is pirated are hard to obtain, but the percentage is often extremely high. In India, for example, it was estimated that in 1986, 98% of all software was pirated, a figure that was thought to have dropped to 76% in 1994, 68% in 1995, and 60% in late 1996. See Richard Heeks, India's Software Industry (New Delhi: Sage, 1996), 133-135, and Financial Times (January 15, 1997). In the PRC and Pakistan, piracy rates are higher than 90%; in the U.S. and U.K., 25-30%. Despite widespread pirating, Carmel, op cit., estimates U.S. software exports in 199[X] as $[XX] billion.

23. Barber, op cit., pp. 298-300.

24. For Japanese/American software and cultural differences, see Masao Ito and Kumiyo Nakakoji, "Impact of Culture on User Interface Design," pp. 105-126 in del Galdo and Nielsen, op cit.

25. New York Times, Monday, December 9, 1996, p. 1.

26. See the useful analysis of variants of Spanish by Javier Garcia Alvarez, "The Spanish Language: Dream or Nightmare," Localisation Industry Standards Association Newsletter (September, 1996) [www.LISA.unige.ch/news.html], pp. 9-19.

27. See Heeks, op cit. and Hans-Peter Brunner, Closing the Technology Gap: Technological Change in India's Computer Industry (New Delhi: Sage, 1995).

28. For updated statistics on international Internet and Web sites, consult [http://risknet.com/scc/whitepaper/growth.htm] or [http://www.mit.edu/people/mkgray/net/web-growth--summary.html].

29. In the magazine section of the neighborhood grocery store near my home in Boston, there are currently six different computer magazines for sale each month. Each magazine has approximately 200 advertisements. In a major bookstore within a mile of my home, I recently counted 27 distinct computer magazines, each aimed at potential purchasers and supported primarily by advertisements. Readers in France, Italy, Argentina, Germany, etc., can tell similar stories.

30. In centrally planned and centrally controlled economies, as indeed in most "free market" economies, most computers are not "personal" but parts of proprietary networks and systems. Even in the United States, banking systems, reservations systems, supermarket checkout and inventory systems, corporate networks operating from central servers, etc. still provide a large percentage of computer terminals. Initially, the line between mainframe (e.g., corporate, governmental) computer systems on the one hand and freestanding personal computers on the other was fairly clear. Increasingly, however, with the growing power and lowering cost of personal computers, and the ease of networking both locally and internationally, the lines between mainframe and "personal" computers are becoming hopelessly blurred. As this happens, the market for "packaged software" increases because, for example, packaged network software, which can be fairly easily customized to specific institutional needs, is increasingly replacing the ground-up proprietary systems of the past.

31. Attitudes toward official, national, and vernacular languages are, as David Laitin points out (Language Repertoires and State Construction in Africa (New York: Cambridge University Press, 1992)), extraordinarily complicated. For example, Laitin reports that in India, passionate advocates of requiring Hindi as a national language known by all Indians may nonetheless send their children to schools where English is the language of instruction. In such cases, nationalist values justify one language choice; practical considerations and ambitions lead to another. There is good reason to believe that this Indian experience is more typical than not.

32. Laitin, op cit. Laitin's work is extraordinarily useful in broadening the understanding of localization. To his basic distinction between vernacular, national, and official languages, he adds a number of other crucial concepts like lingua francas, pidgins, and creoles, argots, code-switching, loan words, prestige languages, etc. A thorough discussion of the future of localization would need to include all these distinctions, and to deal directly with Laitin's prediction that the pattern of "three plus or minus one" is likely to characterize African -- and by implication -- other developing nations in the future. Laitin does not dwell on the fact that the official (usually colonial) language is commonly spoken by only a small elite minority of the population; and his book was written before computers, Internet, or the Web had any impact on language choices. As I note throughout these comments, one of many factors determining language learning and language choice may be whether the language gives access to available computer software and/or to Internet/Web information resources and possibilities of communication with those of other nations.

Another point of Laitin's deserves special attention: namely, his comment that women are not only more likely to be illiterate than men, but less likely to know the official and/or national languages. It follows that localization to the vernacular, which women indeed know, would tend to favor women, whereas localization to national and official languages would tend to exclude women disproportionately in many societies. But in some of the countries and regions discussed by Laitin (e.g., northern India, much of sub-Saharan Africa) the majority of women are illiterate, and by that fact alone effectively are excluded from access to computers.

33. As Garcia Alvarez argues (op cit.), Spanish is probably the most widely used and useful language for localization, with almost as many vernacular Spanish speakers in Latin America and Spain as English speakers in the world. French occupies a parallel position as a lingua franca, but not as a vernacular, in what is often but misleadingly called "francophone" Africa, namely, former French colonies in West Africa where French continues as, in many cases, the official language. (See Laitin, op cit., pp. 129-130, on Senegal.) But despite the possible future role of Spanish, French, or in the more distant future possibly Chinese as lingua francas of international communication, at present and for the foreseeable future, the dominance of English seems overwhelming.

34. These figures are available at [http://www.cr/latstat/], compiled by Guy de Teramond.

35. This is not to claim that the problems of the Net and the Web are simply compoundings of problems of software localization; Net/Web problems are in many respects sui generis. For example, the added complexity of multiple inter-lingual and inter-cultural communications is so great that it becomes qualitatively different from the usual A to B problem of software localization. Furthermore, localizing a software package is ordinarily a protracted project which begins, in large software firms, with the writing of the basic source code in an internationalizable way, and continues through multiple localizations, beta testing, back translation, etc. over a period of months or years. With Net/Web communications, in contrast, any localization must be instantaneous. Nevertheless, some of the issues of cultural acceptability, sub-cultures, elites, etc. are parallel, and a better understanding of the more manageable topic of cultural localization could help in anticipating Net/Web problems.

36. Laitin (op cit.) argues, for example, that there are a thousand distinct languages in the Sub-Saharan West African region alone.

37. I am indebted to Professor Donald Day of the University of New South Wales for this suggestion.