Politics, Culture, and Software

Kenneth Keniston

Massachusetts Institute of Technology

(Published in*Economic and Political Weekly* -- Mumbai, January 17, 1998)

Of India's nine hundred and fifty million citizens, nine hundred million are currently excluded from computer use, the Internet, and the World Wide Web by the near-total absence of software in the languages which the majority of Indians speak. Restated in the jargon of the computer scientist, there has been virtually no "software localization" to any of the major vernacular languages of India.

The exclusion of almost one-sixth of the world's population from what enthusiasts term "the Information Age" raises questions about politics, culture, and software that are important not only to India, but to the entire world. I am no India expert, but it is clear that India, the world's largest democracy, is a nation that despite communal conflict has maintained a vibrant multilinguistic, multicultural society in an era of world fragmentation, and remains committed not only to economic growth but to freedom and social justice. India thus has a rare, perhaps unique, opportunity to affect the directions in which the Information Age will move.

To underline the political and cultural implications of software localization, I begin with a bad dream about the future. I then argue that whether or not software is localized at all, and if so, whether it is adapted to the cultures to which it is localized -- that both these questions are influenced by political and cultural factors. Finally, I suggest that the future social and cultural impacts of software in India, as of other aspects of the electronic age, are in no sense technologically determined, but largely depend on what Indians and Americans decide to do, and specifically on our capacity to work together to set standards for localization to non-English languages that are global without being imperialistic.

A Bad Dream: the Rule of the Digirati

First, the bad dream -- a dream that is part science fiction, part nightmare, but part sociological projection.

In the not-too-distant future, the entire world will be effectively controlled by a small group of individuals whom we can identify, for the sake of convenience, by four simple characteristics: they are all computer literate; they all have an Internet address and/or Web site; they all possess a cellular telephone (probably with direct satellite links in the future); and they all understand -- and speak and write -- English as their first, second, or third language.

This new ruling class -- we can call them the "digirati" -- will be concentrated in the nations of the so-called North, but its members will also be found in Mumbai, Bangalore, Delhi, Nairobi, Buenos Aires, Singapore, Jakarta, Kuala Lumpur, and Johannesburg. They jet from continent to continent; they communicate instantaneously in English over Internet, World Wide Web, or whatever follows. They have instantaneous access to unbelievably comprehensive networks of information; they make financial transactions in Hong Kong, Sydney, London, Lima, Singapore and Calcutta; they exchange scientific information, weather reports, business news and personal gossip at the click of a mouse.

In addition to their economic and political powers, the masters of the new "telectronic" media will be the authors, inventors, agents, actors, and controllers of a cosmopolitan, globalized, consumerist, lowest-common-denominator world culture. This new culture -- if it can be called a true culture -- will be inspired and perhaps dominated by Disney, Sony, Murdoch, MTV (suitably adapted to conditions in Delhi or Buenos Aires), McDonald's, CNN, Mitsubishi, Nike, Philips, Levi's, Nestle, Microsoft, Intel, and corporations as yet to be invented. Faced with the power of this new electronic culture, traditional, non-English-speaking, ancient, non-electronic cultures will stagger and perhaps be overwhelmed.

The remaining 99 percent of the world's population -- not computer literate, not fluent in English, without Internet-Web sites and cellular phones -- will be gently ruled by this new global telectronic ruling class, the new digirati. This other 99 percent will include the 95 percent of the people of the world who do not speak fluent English, all the world's illiterate and innumerate, as well as the underclasses of Northern Europe and North America and the vast majority of peasants, farmers, and workers in the so-called South.

The "rule" of the new telectronic class will be gentle, persuasive, and only rarely violent or coercive; it will be leveraged by the economic and cultural forces of so-called "liberalized" economies. There will be minimal physical coercion, but relentless pandering to consumer desires, a youth culture that spreads to grandparents, satellite TV in every village, World Cup Soccer witnessed by billions, universal blue jeans, T-shirts and running shoes, locally-adapted rock, and (at the "high culture" level) the Three Tenors at the Baths of Caracalla.

But reactions against this dominant, cosmopolitan, global, electronic culture will take ugly forms. Cultural, economic, and political nationalisms of a fundamentalist kind will thrive because of the neglect of local traditions, practices, values, and linguistic identities and their submerging into a single global electronic culture. These new fundamentalisms will build on imagined, recreated, and fantasized pasts. They will hearken back to ancient empires, lost languages, and imagined (though fictitious) eras of racial, ethnic, and/or cultural power and purity. They will be xenophobic and intolerant, rejecting modernization, hostile to political and cultural freedoms, antagonistic to foreigners, immigrants, neighboring nations, and communal minorities within their own borders. Ethnic, cultural, and political purity will be their goal; the exclusion of the ethnically, culturally or religiously impure will be their rule.

If I have painted a dark picture, it is not because I believe that this future is an inevitable consequence of the information revolution. On the contrary, it is precisely because I think we have a chance, through actions we could begin taking now, to avoid the negative cultural and political consequences of a particular kind of information age.

What does this have to do with software localization? Localization is, after all, that highly technical process by which computer programs written in one language by members of one culture are translated into another language for use by members of another culture.

Currently, the major packaged software firms, almost all of which are located in the United States , prepare for localization by setting apart the irreducible source code of major programming languages, operating systems, and applications from the linguistically and culturally specific elements which need to be changed for special local markets. This process is called the "internationalization" of the program code. The list of elements that need to be set apart so as to be "localized" is long: not just obvious text translations, but character sets, scrolling patterns, page geometries, dictionaries, search engines, colors, numbers, box sizes, names, dates, and icons. (As one observer has noted, there is no gesture of the human hand that is not obscene in some culture!)

The complex technical features of software localization are well understood and often written about by specialists. But two other aspects of localization, both of which have significant cultural and political implications for India, are sometimes mentioned but seldom studied. They are: first, whether or not localized versions of major programs exist at all; and second, the embedded cultural content of even technically well-localized programs.

Whether Localization Occurs at All

Let me start with the first question, whether localized versions of English-language software already exist. At present, about 80 percent of the world market in packaged software is produced by American firms, a percentage that is currently growing each year. With few exceptions, localization therefore means whether or not software written originally for an English-speaking audience by American programmers is or can be adapted to other languages and cultures (often with the help of colleagues abroad). What factors determine whether these English-language programming languages, operating systems, and applications are made available to non-English speakers -- that is, are localized?

Consider some curious facts. The Windows NT platform is currently localized, we learn, not only for major European languages with large computer-user populations -- e.g., French, Spanish, German, Norwegian -- but "enabled" (a lesser step than localization) for Catalan, Rhaeto-Roman, Bahasa, and Icelandic. Or, in the case of the MacIntosh operating system, localization is available not only for the major European languages, but for the language of the tiny Faeroe Islands in the North Atlantic south of Iceland (population 38,000), for Kazakh, for Uzbek, etc. But with the exception of English, none of the major languages of India including Hindi (spoken by almost as many people as English or Spanish) is included in either list. The population of the Faeroe Islands have a MacIntosh localization and the inhabitants of Norway have a localized version of Windows NT, but the populations of Uttar Pradesh and Bihar have neither, nor do the populations of Bengal, Tamil Nadu, Gujerat, Kerala, Karnataka, or Maharastra. Unless they speak English fluently, the peoples of India have no access to these major computer operating systems.

How do we explain these omissions? The most common explanation is economic, and I will state it as persuasively as I can. A software company's decision to localize software -- a costly undertaking -- most obviously responds to its perception of potential market demand. Where a large population uses computers, and -- an important qualification -- where piracy rates are low enough that software producers can sell their products rather than have them stolen, then companies are more likely to invest in localization. For this reason, we have French, Spanish, German, Finnish, and Swedish versions of major programs by international software firms like IBM, Microsoft, Digital, Oracle, SAP, etc. In India, I have often been told, the absence of a significant domestic market for localized software means that dynamic Indian software firms, now primarily dedicated to overseas collaborations and the sale of software services, lack any economic incentive to produce software in languages other than English. In any event, the need is limited because, it is said, India possesses the second or third largest English-speaking population in the world. After 50 years of independence, English remains the lingua franca for inter-State communication; and members of Indian elites, whatever their mother tongues, generally have a superb command of English as a second or third language -- therefore, there ostensibly is no market and no need for localization.

This economic explanation is quite plausible. In a country where the annual income (parity purchasing power) of the average Indian is less than half the cost of a well-equipped computer, where almost half the population is illiterate, where almost a third of the population lives at or below the official level of subsistence, and where the cost of an Internet connection may exceed the cost of food for a month, computers -- and therefore localization to Indian languages -- are today beyond the means of any but a minority of Indians.

But before ending this discussion with an economic explanation, we need to ask further questions. Do economic factors alone really explain the existence of localized programs for Iceland, for the Faeroe Islands, and for the Norwegians who speak the dialect of their northern mountains? Why have large commitments to localization been made by American software firms in the People's Republic of China, when there, too, piracy rates are said to exceed 90 percent, when there are deep differences between the political philosophies of the People's Republic and the United States, and when doing business in the PRC is generally unprofitable and often involves, it is said, very large hidden costs? Is economics alone an adequate explanation?

One reason that economic factors, although vastly important, do not provide a sufficient explanation for the absence of localization to Indian languages has to do with the long-term planning cycle of software firms. I know the saying that the most successful software firms plan a full six hours ahead, because the market and the technology change so fast! But in fact, all major software firms, abroad and in India, have long-term planning cycles as well. American firms, among them Microsoft, place long-term bets on future markets, bets which may not pay off for a decade or more. Along with the capacity for quick adaptation, then, leadership in the software industry also requires the ability to look far ahead. American software firms' investment in R&D in China is a case in point -- a way of establishing a foothold in a potentially vast market in the distant future, though rarely a source of current or near-term profit.

With regard to India, even if the corporate, business, and personal demand for vernacular software is limited today, it takes little imagination to foresee a day when it might be large, indeed vast. To foreigners like myself, the vitality of the Indian economy in recent years is impressive. India already is said to have the largest middle class in the world. National growth rates overall may appear modest because of the moderate growth of the vast agricultural sector. But industrial growth rates in recent years, especially in the southern Indian states, have been in the double digits. And it does not, as they say, take a rocket scientist to predict that if these rates of growth continue, more and more firms, businesses, and individuals -- banks, warehouses, merchants, shippers, shopkeepers, libraries, post-offices, bus lines, private and eventually public schools and parents -- will little by little constitute a growing, and eventually a vast, market for software in Indian languages. Moreover, piracy rates in India have been dropping due to a concerted effort to bring India's rates closer to European/North American rates of 20-40 percent. Thus, from the point of view of software manufacturers in India and overseas, it would seem a reasonable economic gamble to anticipate the emergence of a substantial demand -- i.e., a profitable market -- for software in the major Indian languages. Indeed, not to anticipate this day would seem economically irrational.

Why then is localization to Indian languages not yet a tangible reality? We are compelled to move beyond an exclusively economic perspective toward one that also takes account of politics and culture. As a way of approaching this topic, let us consider the role of culture and politics in localization to Chinese.

The technical problems of localizing from English to Chinese are formidable. Chinese is an ideographic written language with tens of thousands of ideographs (only 7000 of which currently exist in Unicode), no phonetic alphabet, and no single agreed-upon way of using the Roman (qwerty) keyboard to enter ideographs. Moreover written Chinese is linked to complexly tonal spoken languages which vary dramatically (and often unintelligibly) from region to region. These problems are staggering. Yet localized Chinese versions of many major programs already exist.

Why? The reasons are partly cultural and partly political. The Chinese written language is everywhere the same, even though spoken dialects often are mutually unintelligible. Moreover, the present Chinese government is authoritarian and highly centralized. So it is possible to negotiate with a single ministry in Beijing and make, at least on paper, binding agreements about standards of localization for all 1.3 billion citizens of the People's Republic. The official outcry when Microsoft initially failed to localize to the simplified Chinese character set used in the PRC illustrates, paradoxically enough, the role of central authority in determining software standards. Not a single public voice applauded Microsoft for using the traditional, pre-Revolutionary character set still used in Taiwan and Hong Kong. On the contrary, Microsoft was universally denounced; the offending software was removed from the market; and eventually new software using the "correct," simplified characters was introduced. Windows NT now exists in both the old Mandarin form for Taiwan and the new simplified PRC character set. The official standard was and is clear.

Another fact that makes localization convenient is the uniformity of the written Chinese language. Although there are many dialects of spoken Chinese, the written Chinese required for computers is essentially the same from

Canton to Beijing, and again, standards are set in Beijing. It is thus politically, culturally, and linguistically possible to make binding arrangements with central authorities to localize to a single language. China, to be sure, like India remains a relatively poor nation where the average individual family or small business still cannot afford to purchase a computer, even if political authorities allowed it. But centralized political authority and uniform localization standards make it reasonable to place a long-term bet on the eventual development of a profitable Chinese market.

In India the situation is obviously more complex. India is a democratic, federal nation with an admirable tradition of multilinguistic and multi-cultural pluralism. No ministry, no individual, no party can presume to speak for, or set standards for, all of India. There is no majority language; eighteen languages are officially recognized; and many more are spoken. The prevailing policy has been liberalism and tolerance with regard to the use of local languages.

But this virtue creates a problem. I recently was in contact with the director of internationalization of a large American software firm. He said, "We would like to do some localizing in India, but we don't know how. Which Indian languages?"

On the face of it, this question is easily answered: "Start with Hindi because Hindi has the highest use and most nearly national status." Depending on how its boundaries are defined, Hindi is spoken by about four hundred million Indians, whether as a first or a second language. It is among the five most widely spoken languages in the world.

But to non-Indian observers from afar, it may seem that there are difficult issues with Hindi. Whose Hindi is to be chosen as the standard language? Nehru is said to have complained that he could not understand All-India Radio's reports of his own speeches in the highly Sanskritized version of Hindi then in use. Nor, he complained, could he read the Indian Constitution in the variant of Hindi in which it was then written -- although he was, of course, a Hindi speaker as well as a master writer and stylist in English. Studies of the linguistic patterns of Northern India indicate wide regional variations in spoken Hindi-Hindustani. The division of the spoken Hindustani of the peoples of the Northern plains into two distinct languages -- a Persianized Urdu, and a Sanskritized Hindi -- has produced two written languages and to a lesser degree two spoken languages where once, at least in some regions, there had been only one. Finally, film and television, in particular satellite television, are said to be causing a further blurring of regional linguistic distinctions, and, it is claimed, may even be creating a new language, "Hinglish", heard on some Indian-produced films and television.

In addition, the introduction of localized software raises the complex issue of the politics of language in India. At one level, the multi-lingualism of India as a nation, the acceptance of eighteen official languages, the coexistence of many linguistic groups in all major Indian cities, and the fact that many Indians speak one or more languages in addition to their mother tongues -- these facts of multilingual tolerance and pluralism first strike the eye of the foreign observer. Language, it is sometimes said, is not a political issue in India.

But at the same time, one needs to ask what forces are held in check by today's permissive policies. If the director of internationalization mentioned above not only started but stopped with Hindi, the repercussions would be unfortunate to say the least. Any localization effort which concentrated exclusively on Hindi would arouse the wrath of a majority of Indians, and in particular of southern Indians who speak languages of Dravidian origins. As Das Gupta and Brass have documented, and as Rushdie has narrated in an episode of his great work, *Midnight's Children*, linguistic issues have on occasion aroused powerful, even lethal, passions. Privileging any one language inevitably means privileging those who speak it, just as today, privileging English bestows special benefits on those whose English is adequate for computer use. Thus, localization only to one Indian language (e.g. Hindi), or indeed only to two or three, could well arouse the quiescent passions that are now kept latent by the prevailing policy of linguistic pluralism.

But from the point of view of a software firm, localizing into all eighteen of the official languages of India may seem an inordinately complex, expensive, and difficult task. No matter where and by whom decisions about languages for localization are made -- whether by an all-Indian body, by a multinational corporation, or ideally by a consortium of Indians and multinationals -- the possibility of stirring up ardent linguistic nationalism needs to be taken into account.

In short, in addition to real economic factors, cultural and political factors help explain the absence of localized software in the vernacular languages spoken by most of the peoples of India.

Embedded Cultural Content

I now turn to a second related topic, namely the way culture and politics inter-penetrate the very heart of software. My first theme was the role of culture and politics in determining whether localized versions of English language software are developed. My second theme is the role played by culture and politics in the content of localized (i.e., non-English-language) software.

Writers of manuals for software internationalization invariably pay lip service to cultural factors in software (e.g., they note that the meaning of the color red differs from one culture to another). But they almost never, I think, examine thoroughly enough the built-in or embedded views about the nature of reality, the nature of users, and the social world contained in software and hardware. To deal with this subject adequately is beyond my ability, so. my comments here will be suggestive and programmatic.

For those who are engaged in advanced scientific work, who live in the so-called "modern world", or who are actual or potential members of the "digirati," the telectronic ruling class -- for us, a whole set of assumptions about time, human nature, and society may have come to seem "natural". These root assumptions are in essence the rational, analytic, reductive, scientific assumptions that were incorporated, in Europe about 500 years ago, into what we now define as "modern" views of time, matter, nature, and human nature. Today, they constitute the ideological bedrock on which science and technology -- including electrical engineering and computer science -- are built. But we need to emphasize the fact that they bring to software (and indeed to the hardware on which it runs) an inescapable commitment to a world view that was unknown throughout most of world history, and that remains alien to much of the world's population today.

Consider only the "modern" understanding of time. As historians like Lewis Mumford, E.P. Thompson, and David Landes have shown, during most of world history, time has been organized around shared experiences like the seasons, day and night, the life cycle, the duration of common tasks, the phases of the moon, or great religious and historical cycles. In many traditional cultures, modern assumptions about time are reversed or inverted: people are assumed literally to face the past, and thus to walk backwards into the future (an eminently sensible view since we can always visualize where we have been, but are never sure where we will be tomorrow); time may be seen not to move forward, but in circles or backwards; historical cycles may be viewed as periods of decline or as recurrent circles of generation and regeneration rather than of forward movement or "progress".

The concept of time embedded in the so-called "modern world view" is of course radically different. It emerges in China and Europe in the Middle Ages with the effort to divide time into discrete units, and it is disseminated with the widespread use of mechanical clocks, which in turn, became governors of social and individual life, including especially work life, in the new industrial world that was first built in Northern Europe and America in the eighteenth century. Time -- once circular, discrete, idiosyncratic, seasonal, or organized around tasks -- became linear and directional, moving like an unerring arrow in a straight line from past to future.

Moreover, time has become a quantity that can be measured exactly, divided into infinitely smaller components -- even nanoseconds. And as a quantity, time has also been defined as a commodity that can be spent, saved, splurged, invested, wasted, or, as Benjamin Franklin put it, thought of as money. To the 14th century peasant in France or the 20th century agricultural worker in Uttar Pradesh, such concepts of time were, and are, alien. But today, the power of time over people like us is such that we even define our vacations as a way of "getting away from the clock" -- or the digital timepiece that replaces it. But try to imagine a computer not built on this conception of time! Try to imagine a computer without an internal clock! Try to write software in which the only measure of time is the phases of the moon!

Assumptions about how to live -- about human values -- are also deeply if implicitly embedded in our understanding of computers. In the "modern" world we commonly assume that saving time is good, that slow is bad and fast is better. Thus, Moore's Law -- that the number of computer operations per unit cost doubles every eighteen months -- is commonly taken to be a universal indication of progress. The rationale for ever-faster computers and ever-more complex programs is that they enable us to do more work in less time, and we take it for granted -- as "natural" -- that this is a Good Thing.

This concept of linear, clock time, and the virtue of speed is bound up with other notions even more directly centered on the life of human beings: i.e., notions of the value of efficiency (doing the most work in the least time), and of productivity (producing the most in the least time with the least investment, including investment of time). We humans are "naturally" assumed to seek optimum productivity and efficiency: computers are considered valuable because they aid us in this "natural" quest.

India is a nation with 5,000 years of history, which retains powerful traditional sectors. Indians are therefore more likely to realize that concepts of linear time, of optimizing speed, of maximizing efficiency and of increasing productivity are not, as they are widely assumed to be in the Northern countries, simply "natural", inevitable, "rational", universal assumptions about people and the world. Rather, they are products of specifically "modern" historical conditions, and they appear "natural" only to specific groups at specific historical periods. Hence they also are unknown and profoundly alien to many, perhaps most, of the world's peoples. I think, for example, of M.N. Srinivas' marvelous account of a South Indian community in his book, *The Remembered Village*. Clock time, speed, saving (or wasting) time, efficiency, productivity -- these attitudes are completely absent in a human and social world that coheres around other traditional values and concepts of time.

Cultural Content in Software: Culturally-Specific Assumptions

Much more should be said about the cultural assumptions embedded in the hardware and the software of every computer in the world. But I now want to mention briefly a second, different kind of cultural element in software, namely, the incorporation into software of assumptions about the human and social world that are specific to particular nations or economic systems.

In a well-known work, Hofstede studied the outlooks of workers and managers in more than twenty countries. For example, he found that out of all the countries studied, the workers and managers in United States prized most highly the value he called "individualism", which he contrasted with a more "collective" orientation toward life, achievement, and social relations. Thus, it was not surprising, on a recent trip to Argentina, to hear a rural Argentine primary school teacher complain that the well-translated, well-localized American educational software used in her school presupposed, as she put it, "solitary individuals sitting at a keyboard solving problems as rapidly as possible". The translations to Spanish were excellent, she said, but somehow these values were "not Argentine". Indeed, she wondered whether if one took this U.S. educational software, excellently localized to Spanish, as a model of life, it would not transform young Argentine children into "little North-Americans".

I doubt that children can be transformed into "little North-Americans" by a single educational computer program. But nonetheless the teacher's perception is crucial. Is it not clear that software not only can solve problems, but can convey a set of implicit and culturally-specific assumptions about the world? Could the hegemony of American packaged software be one small aspect of a larger pattern in which "American" -- or more precisely, global technetronic -- culture spreads across the world at the expense of local diversity?

To take another example, Per Lind, a distinguished Swedish scholar, has noted that American and European management information software assumes a certain kind of "rational", i.e., market-based, world. Components ordered are assumed to be delivered on time, items warehoused last week are still there this week, prices are determined by costs plus benefits, prices quoted are prices paid, and there is no space in the software for entering payments to facilitating intermediaries, much less for subtracting the expenses of tips or bribes. This vision of the world of commerce may accurately characterize Germany, France, and Sweden, but does it characterize Eygpt, Kenya, Gabon, Rwanda, or Zaire? When Western business software is introduced into firms or government agencies in those countries, observers sometimes wonder why it is ineffective or little used. Lind's answer is clear: the software presupposes, takes for granted, a particular social "reality" and economic "rationality" -- in this case a complex system of predictable capitalistic market economic relationships -- which do not correspond to the quite different realities and rationalities in other parts of the world.

Let me only mention one other set of absolutely critical but implicit assumptions about software users -- namely, that users will be "numerate" (i.e., that they will have a reasonable command of basic arithmetic, if not of higher mathematics), that they will be literate (that they will be able to read instructions on the screen and use a keyboard in whatever language the keyboard is designed for), that they will be accurate (that they will not misspell addresses or computer commands), and finally that they will be capable of working in a microworld where all choices are binary (yes/no, up/down, delete/retain, go back/go forward). These assumptions presuppose the presence of a certain kind of person on one end of the computer. But what if he or she cannot use numbers? What if she or he cannot read the instructions on the screen? What if he or she cannot write, or cannot type? What percent of the world's population today satisfies all of these elementary requirements of computer use, when half of the people in the world have never made a telephone call?

My general point is simple and surely obvious to anyone who has worked with computers. It is that software localization, while it is importantly a matter of technology transfer and economics, is also a practice with decisive cultural and political parameters. The content of localized software is determined not only by the language chosen for localization, but by deep, underlying, usually implicit and unacknowledged (because thought to be "natural") assumptions inherent in the software itself. And software carries with it a view of the world, of people, of reality, of time, and of the capabilities of users, which may or may not be compatible with any given cultural and social context.

A Good Dream: Consortia for Software Localization

Let me now recast my initial bad dream. Some commentators argue that the electronic-communications revolution, far from improving the condition of the Southern nations and of the poor in the Northern nations, will inevitably enlarge the gaps that exist. Some claim that this is an inevitable consequence of any new technology that is accessible only to an elite.

To me, in contrast, it seems that the consequences of the new telectronics will not be determined by the technologies as such, but rather by the ways we use them, by the contexts within which we choose to deploy them, by the wisdom and values that guide our actions in using them.

To return to the issue of localization to Indian languages: whether, how, or when this is accomplished, and for whom, are obviously crucial factors in determining whether the information age widens the gaps that currently exist in India -- and everywhere else -- between rich and poor, powerful and powerless. If English were to remain the only easily available Indian language for computer use, and if we make the reasonable assumption that access to computers (and to computer-based electronic communications) is empowering, then 95% of the Indian people who do not speak good-enough-English for computer use will automatically be disempowered. Existing gaps will grow.

Whether this happens is above all a matter that Indians in collaboration with international software companies have the power to determine. On the one hand, there is of course the possibility of consolidating the existing privileges of a gifted, educated, cosmopolitan, English-speaking elite. If this happens, it is likely -- in India as in any country -- that fundamentalist reactions against the growing power of the globalized English-language electronic culture and an English-speaking elite will mount, and that these reactions might overturn, as they have done elsewhere, India's traditions of multicultural tolerance, democracy, diversity and human rights.

There is, however, another possibility -- if you will, a happy dream. It is a dream of Indian and international cooperation to make computers accessible to the vast majority of Indian people who are not fluent in English. It is, in short, a dream of localization to Indian languages.

Let me give two recent examples of the kinds of collaborative standard-setting that could serve as models of pathways to make the new telectronics accessible to new groups of Indians.

This past spring, there was an unprecedented meeting of representatives of the large software firms in the United States to discuss developing common standards for software internationalization. Although these firms compete tooth and nail for American and international markets, they are nonetheless trying to develop, over time, uniform standards for internationalizing new programs. Developing common internationalization standards will be a complex, technical, difficult job. These standards will require each company to change existing procedures. But they will also make it far easier for localizers in, let us say, Mysore, Mumbai, Calcutta, Ahmedabad, Delhi and Madras to develop vernacular versions of the English-language software and applications written by these companies. As the work proceeds, we could even move towards a day when all major new software programs have, as it were, a common "plug-and-play" localization interface. If that day arrives, the cost of localization from English to other languages, including Indian languages, will decrease (and the probability of doing so will increase).

A second example originates in my own institution, MIT. As the World Wide Web grows in importance, as bandwidth increases, as traffic multiplies, as problems of encryption become more complex, as commercial uses expand, as use of the Web for telephony and digital video burgeons, the development of new worldwide standards becomes necessary. At MIT a consortium subscribed to by dozens of companies worldwide was recently created. It aims to establish common standards to insure that the worldwide digital communication networks and technologies developed by distinct firms in different nations will, in the decades ahead, be compatible with each other and indeed, compatible with all major languages, including those of India. In this process, each participant has had to relinquish sovereignty, to modify existing procedures, to disclose corporate secrets, and in some cases to abandon technologies in which they had deeply invested. The important points, however, are that working groups have been established, standards hotly debated, and progress made.

In short, it is possible for people from divergent backgrounds, with divergent and often conflicting commercial and intellectual interests, to cooperate in facilitating and widening access to the electronic world. This is the good dream -- or one of them.

Earlier I noted the confusion of American software firms, usually inadequately informed about India, about how to proceed in this complex nation. The leaders of many Indian software firms also have expressed parallel uncertainty about how or whether to proceed with non-English Indian languages. This is the confusion of software producers. On the side of desired social goals, however, I also have been struck with the almost universal hope of Indians to move toward a time when village stores, shops, banks, post offices, warehouses, schools, and eventually homes can be interconnected on Internet and the World Wide Web; when Indians of all this nation's rich cultures could have access to computers in their own languages, and when the potentialities of digital technologies and multimedia for recording, storing, deepening, and accessing the riches of Indian cultures could help to strengthen, rather than vitiate, the variety of this nation.

For that to happen, especially in a nation so complexly multilingual and multicultural, will require an unusual degree of cooperation and visionary leadership from Indians in both private and public sectors. They will need both vision and determination in order to develop (surely in collaboration with international software firms) common standards for localization to Indian languages. Enormous creativity already is going into plans for developing standards for Indian languages. At the National Center for Software Technology, CDAC, Kondan Railways, NASSCOM, the Bhaba Center for Atomic Research, and elsewhere, a variety of ingenious methods for entering Indian languages and scripts into computers have been developed and continue to be developed. Moreover, major international firms have announced plans to develop versions of their current operating systems in Indian languages.

The imaginativeness and diversity of all these efforts is a promise for the future, but also a difficult challenge. The stage now seems to be set for a final act in which the key players come together to produce a grand finale -- coherent and agreed-upon standards for localization. Such standards could provide the "plug-ins" that enable Indian languages to dovetail with the work of the American consortium in developing common standards for localization. Without such coordination between foreign and Indian developers of standards for internationalization, the outcome is likely to be a Tower of Babel.

The alternative to Babel is that there will develop consortia of Indians and multinationals to develop standards for localization to the major Indian languages, perhaps beginning with Hindi but surely including other languages, especially of the south of India. These consortia could bring together major participants in the public software sector with major firms in the Indian private software sector, and with them, representatives of foreign software firms. Their goal would be to establish mutually agreed-upon standards for such matters as keyboard entry, scripting and fonts, standardization of languages, and the uniform translation of critical computer terms. Accomplishing this will not be simple, either technically or (in a broad sense) politically: too many creative people have devoted too many hundreds of hours to differing solutions, not all of which can prevail.

The stakes are very high. For unless Indians come together to develop common standards for localization, there are only two alternatives. One is that such standards will never develop, and real localization will not be implemented. The other is that if localization to Indian languages is accomplished, it will be defined by default in Redmond, Washington, rather than in Delhi, Bombay, and Bangalore, and the results could too easily be inappropriate to India.

We are left with questions rather than solutions. How and by whom should the basic languages of India be defined? How should the keyboard be used to enter each language into the computer? What are the possibilities of voice recognition? How should fonts and scripting be defined? Which languages should have priority for localization? How can the costs of developing standards be apportioned between Indian and foreign, private and public agencies? How can the creative work already done at NCST, CDAC, Tata, IIT Kanpur, CICT, Kondan Railways, and other groups be optimally incorporated into the final standards? And above all, how can the results be responsive to the interests of the mass of Indian people?

None of these questions will have an easy answer. It is no easier for Indians than for Americans to speak with a single voice. But if these questions are not answered, the result is the likely exclusion of most non-English-speaking Indians from the electronic world. To suffer that defeat without a major effort to avoid it would be a great pity, especially when there is so much obvious desire in India to use the electronic revolution to close rather than widen the gaps in this society.

I trust that I have not overstepped the bounds of propriety in discussing an Indian situation about which I know too little. In this anniversary year of a half century of Indian Independence, Indians often point to what they consider the incompleteness of the Indian transformation. But no one can visit India today without also recognizing India's extraordinary achievements: its maintenance, through Partition and communal conflict, of a sturdy democratic tradition, of openness and mutual respect in most areas of life, its extraordinary diversity and cultural pluralism, the recent unleashing of the vast economic potential of this nation, and finally, its continuing commitment to the difficult task of including the excluded in India's future. The goal in software, as elsewhere, is to see that the information revolution assists in that task.

Acknowledgments

I am grateful for the help of many colleagues. Among them, I would single out Venkatesh Hariharan, formerly of *Computer Express*; Dr. Navin Singhi of the Tata Institute for Fundamental Research in Mumbai, who organized the seminar at the Tata Institute where this paper was originally given; Professor M.N. Srinivas of the Indian Academy of Science, and Professor Michael Dertouzos of the MIT Laboratory for Computer Science, for their help and encouragement. I am especially indebted to Professor Pat Hall of the Open University (U.K.) and the University of Kathmandu for his generous assistance. His comprehensive paper, "Vernacular Software in South Asia: What Happens Now and What is Needed," provides an essential account of the state of the art with regard to software localization in India as of December, 1997, and his help has been invaluable. Funds for this study were provided by Nippon Electric Corporation through a grant administered by the Provost's Fund at the Massachusetts Institute of Technology.

Bibliographic Notes.

On world culture and fundamentalism, see Benjamin R. Barber, *Jihad versus McWorld* (New York: Times Books, 1995) and the four volumes of the Fundamentalism Project, edited by Martin E. Marty and R. Scott Appleby, (Chicago: University of Chicago Press, 1991-1993 and 1994). See also Manuel Castells, *The Information Age: Economy, Society, and Culture. Volume 1: The Rise of the Network Society* (Malden, MA: Blackwell, 1996).

There are many useful works on software internationalization. See, for example, Nadine Kano, *Developing International Software for Windows 95 and Windows NT* (Redmond, Washington: Microsoft Press, 1995); Elisa M. del Galdo and Jakob Nielsen (eds.),*International User Interfaces* (New York: Wiley, 1996); Tuoc V. Luong, James S.H. Lok, David J. Taylor, and Kevin Driscoll, *Internationalization: Developing Software for Global Markets* (New York: John Wiley & Sons, 1995); and P.A.V. Hall and R. Hudson (eds.), *Software without Frontiers. A Multi-platform, Multi-cultural, Multi-nation Approach* (Chichester: John Wiley and Sons, 1997).

My account of languages in which "localization platforms" are available is taken from Appendix A, "Current Platform Support for Internationalization" in Hall and Hudson, *op. cit.* (1997). Hall's article, "Vernacular Software in South Asia: What Happens Now and What is Needed" indicates that since the publication of the aforementioned work, several new localization packages, developer kits, etc., have been released or are planned in India, though none are widely used. On September 15, 1997, for example, Tata IBM announced launching a Hindi version of PC-DOS; see, "Hindi DOS will boost PC Demand", *Hindustan Times* (September 23, 1997). As is well known, however, DOS is an old operating system which is not currently used for modern computer applications.

On Microsoft's early experiences in China, see "Microsoft's Long March", *Business Week*, (June 24, 1996), pps. 52-54.

Geert Hofstede's best known work is *Culture's Consequences: International Differences in Work-Related Values* (Beverly Hills, CA: Sage Publications, 1980). See also Alex Inkeles, et al. *National Character: A Psycho-Social Perspective* (New Brunswick, NJ: Transaction Publishers, 1997).

The question of who speaks what languages in India, and the definition of what is meant by "speaks", like the problem of linguistic politics, deserves a separate, detailed treatment. The number of Indians said to speak English is highly variable: I have heard numbers ranging from twenty to one hundred million, although five percent (or fifty million) is a widely accepted figure. For computer use, of course, the critical question is how much English is necessary in order to master (unlocalized) English-language programs, a question that has never (to my knowledge) been studied.

On the general issue of Indian languages, I am indebted to Harsh Kumar for an initial orientation. The best detailed analysis of Indian language use is to be found on the World Wide Web at <www.sil.org./ethnologue/countries/inda.html>. But this lengthy document often relies on out-of-date figures (e.g. 1961 figures for English in India). Using more current figures, it indicates one hundred and eighty million primary mother tongue speakers of Hindi (in 1991) and three hundred and forty six million total including second language users in India (in 1994.)

On the issue of linguistic politics, the early works of Jyotirindra Das Gupta, *Language Conflict and National Development: Group Politics and National Language Policy in India* (Berkeley: University of California Press, 1970) and Paul R. Brass, *Language, Religion and Politics in North India* (Cambridge: Cambridge University Press, 1974) are essential, as is Brass's more recent work, *The Politics of India Since Independence* (2nd edition. Cambridge: Cambridge University Press, 1994). See also Joshua Fishman, Charles Ferguson, and Jyotirindra Das Gupta (eds.), *Language Problems of Developing Nations* (New York: John Wiley and Sons, 1968). On language, politics, and nation-building, David Laitin's, *Language Repertoires and State Construction in Africa* (New York: Cambridge University Press, 1992) uses the Indian case as a model for language patterns in other developing nations.

On the "modern" conception of time, see David S. Landes, *Revolution in Time: Clocks and the Making of the Modern World* (Cambridge, MA: Harvard University Press, 1983); Lewis Mumford, "The Monastery and the Clock" in Mumford, *Technics and Civilization* (New York: Harcourt, Brace and World, 1934); and E.P. Thompson, "Time, Work Discipline, and Industrial Capitalism," *Past and Present* (December 1967) 38: 56-97.

Per Lind's work in Egypt is described in his *Computerization in Developing Countries: Model and Reality* (London; New York: Routledge, 1991). Lind's work is especially valuable because he rejects the European notion of a single economic "rationality" and posits instead "multiple economic rationalities" in different societies.

The MIT-based consortium is described in Michael L. Dertouzos, *What Will Be: How the New World of Information Will Change Our Lives* (San Francisco: Harper, 1997). To my knowledge, an account of the discussion of internationalization standards by major software producers in 1997 has not been published.