# Global Networks: Computers and International Communication

edited by Linda M Harasim

The MIT Press

From The MIT Press

MITCogNet

This book was set in Stone Serif and Stone Sans by .eps Electronic Publishing Services and was printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Global networks : computers and international communication / edited by
Linda M. Harasim
p. cm.
Includes bibliographical references and index.
ISBN 0-262-08222-5
1. Wide area networks (Computer networks)—Social aspects. 2. Telematics.
3. Computers and civilization. I. Harasim, Linda M. (Linda Marie), 1949– .
TK5105.87.G57 1993
303.48'33—dc20 92-47473
CIP

# 15 Islands in the (Data)Stream: Language, Character Codes, and Electronic Isolation in Japan

*Jeffrey Shapard*

Once upon a time long ago, Japan was naturally secluded from most of the world by vast distances of land and sea. The only way in, or out, was to sail treacherous seas. The flow of trade and ideas and information and communication, what there was of it, was a difficult process. Then came the Portuguese and the Dutch in the early seventeenth century with their missionaries and their merchants. So destabilizing was their influence that the powers that were, the Tokugawa Shogunate, limited the access of these westerners by allowing them to sail only into the port of Nagasaki and to reside only on a small artificial island called Dejima, constructed for this purpose and linked to the rest of Japan by a narrow bridge. From 1641 until 1856, when Admiral Perry and his fleet of warships put an end to this policy of national seclusion in a chain of events that toppled the Tokugawa Shogunate, opened Japan to the world, and opened the world to Japan, tiny Dejima was the only window between Japan and the world. Its residents could not wander off the island to mingle with the rest of Japan, and only a small handful of Japanese were able to go to Dejima (Kodansha, 1983).

In a way that resonates strangely with history, the Japanese language and the prevalence of character codes incompatible with international standards results in a new kind of natural seclusion today for the individuals on the *pasocom tsuushin* (personal computer communications) islands of Japan, as well as for the researchers on the JUNET/WIDE archipelago. This chapter gives an overview of language and character codes in electronic networking in Japan, discusses a case illustrating problems faced and solutions taken, and proposes directions for further research and development. The Dejima Syndrome in the Japan context is used as an example.

Differences in languages and the character codes needed to support them are important issues of global scope that must be addressed more widely, and which must result in broader thinking throughout the field of computer networking. Narrow vision, one-byte seven-bit ASCII biases, the assumptions about character coding that arise from them, inadequate international standards, and local solutions that disregard what international standards there are and that pay no heed to the ramifications for others—all these are serious related problems that inhibit, rather than enhance, increased connectivity and communications.

### *Setting Sail: The Japanese Writing System*

The origins of modern Japanese remain a matter of some academic controversy, pulled between linguistic reconstruction through analysis (Miller, 1972) and desires for mystical uniqueness as an isolate (Kindaichi, 1979). Whatever the real story of its origins, the Japanese language does not come from China, although its writing system does.

The anthropological linguist Edward Sapir (Sapir, 1921; Mandelbaum, 1970) noted that language and culture are two sides of the same coin. The Japanese language illustrates the dynamic borrowing by Japanese of useful aspects of foreign cultures, from the Chinese *han* ideograph character writing system itself, which came to Japan with wandering Chinese and Korean traders and Buddhist priests a couple millennia ago, and the new lexical items (words) it allowed to be created, to the variety of localized lexical items from other languages borrowed along with the concepts and things they describe. *Kanji* (the Japanese form of han ideographs) and the supplemental scripts derived from them have become an integral aspect of Japanese culture.

Modern Japanese writing typically involves a combination of the following:

- Kanji (Japanese form of han ideographs) for most of the core lexical items, with several readings possible for many of them, *on-yomi* if based on original Chinese readings, of which there may be several, and *kun-yomi* if based on native Japanese lexicon
- *Hiragana* (cursive syllabic characters) for grammatical functions and inflections, and for native Japanese words done more easily than in kanji

- *Katakana* (block syllabic characters) for sounds and for words borrowed from other languages
- Little hiragana and katakana as *furigana* alongide kanji to help the reader figure out which readings are being used
- *Romaji* (roman characters) sprinkled about for effect
- Arabic numerals (0–9) and various other symbols for punctuation, footnoting, listing, currency symbols, and so on

Despite the "Nihongo boom" (an increased interest in the Japanese language) over the last ten years, most likely as a result of Japan's emergence as a world economic power, perhaps no more than a million people outside Japan speak or, especially, read and write Japanese, and fewer than 1 percent of these are Japanese expatriates or emigrants. At the end of 1991, there were 620,000 Japanese living abroad, most of them in North America (258,300), South America (130,600), Western Europe (109,700), Asia (83,900), and sparsely populated Oceania (21,400) (KKC, 1991). Japanese is, therefore, a very big local language.

## *Treacherous Waters: Platforms, Standards, and Other Characters*

Kanji and the complex nature of the Japanese writing system have been a major challenge in Japanese computing and computer networking and have slowed down development in ways not faced by those who can get by just fine with the smaller set of 94 printable characters of the total 128 characters defined as part of the seven bits of the one-byte American Standard Code for Information Interchange (ASCII) (Lung and Nakamura, 1991). Even by using the full eight bits of a one-byte character code, the resulting 256 possible combinations still fall far short of the 7,000 or so needed for modern Japanese. But ASCII seven-bit biases, especially in English-speaking countries where so much research and development in computers and networking have come from, are in the very kernel of most computer operating system environments.

By 1983, the Japanese Industrial Standards bureau (JIS) had defined a Japanese character set that contained 6,877 two-byte characters: 6,353 kanji in two levels (2,965 kanji arranged by pronunciation in level one and 3,388 kanji arranged by radical in level two), 86 katakana, 83 hiragana, 10 numerals, 52 English characters, 147 symbols, 66 Russian characters, 48 Greek characters, and 32 line elements (Lunde, 1990). Each byte contains seven bits, with the high bit undefined. The original JIS C6226 code was expanded in 1985 as JIS X0208 and JIS X0202 (equivalent to ISO2022) to include more kanji and to be closer to the ISO (International Standards Organization) standard. Since ASCII is often used along with kanji, JIS X0202 defines escape sequences used for shifting between one-byte seven-bit ASCII and two-byte seven-bit JIS X0208 kanji characters. As a result of some of the problems faced by mainframe operating systems where two-byte codes were too difficult to implement, there is also JIS X0201, which defines one-byte codes for katakana (Murai and Kato, 1987). In late 1990, JIS defined a supplemental character set called JIS X0212 to include another 5,801 kanji, 21 symbols and diacritics, and 245 diacritic-marked roman characters, bringing the total defined kanji character codes to 12,156, although computer manufacturers have yet to implement them (Lunde, 1992).

However, while the JIS character codes gave a useful common reference, computer makers and software developers in Japan had their own needs and created their own variations. For example, Nihon DEC (Digital Equipment Corporation Japan) developed DEC kanji code, based on JIS C6226 in organization but using two-byte, eight-bit codes rather than two-byte seven-bit codes. Later, a group of UNIX systems developers got together in a task force and advisory committee convened by AT&T International (now UNIX Systems Laboratories) to develop an enhanced UNIX code (EUC) for kanji. They ended up accepting a proposal based on DEC kanji code, that is, a two-byte, eight-bit approach rather than the JIS two-byte, seven-bit standard (Burkley, 1989a). Meanwhile, as the *pasocom* (personal computer) revolution hit, ASCII Corporation in collaboration with Microsoft developed yet another kanji code: Shift-JIS or MS-Kanji, with a combination of a one-byte 8-bit code compatible with JIS X0201 and a two-byte 8-bit kanji code compatible with nothing else, differing in basic organization from both JIS and EUC. Kanji character code development for personal computers was influenced by processing speed considerations, and it seems to have been overlooked that these underpowered little toys would ever need to be connected and have to talk to real computers either as "intelligent terminals" or as peers on a network. To top it all off, Shift-JIS is also the system kanji code on the millions of inexpensive portable Japanese language *waapuro* (word processors) that have flooded the market in recent years, combining with personal computers to make the most incompatible character code also the most widespread in Japan.

So, today, in addition to the two-byte, seven-bit JIS C6226 and JIS X0208/X0202 kanji codes, various two-byte, eight-bit proprietary variants in the mainframe and minicomputer environment, and the two-byte, eight-bit EUC in the UNIX environment, there is also the mixed-byte, eight-bit Shift-JIS on millions of pasocom and waapuro.

And these are treacherous waters for network sailors.

### *Archipelagos and Islands: Isolation and Parallel Realms*

There is a contrast between the significant international connectivity of the Japanese academic network realm of the JUNET/WIDE archipelago and the nearly absolute isolationism of the islands in the pasocom tsuushin realm. They are in parallel, but each seems to have little interest in or awareness of the other. While JUNET/WIDE researchers continue to pioneer the application of international standards in the Japanese context, pasocom tsuushin system developers continue to reinvent the wheel in nonstandard ways and develop local solutions incompatible with international standards.

The most significant networks in Japan in terms of size, connectivity, and global networking are JUNET (Japan UNIX Network) and WIDE. They are closely related. JUNET, which began in 1984 with dialup UUCP (UNIX-to-UNIX Copy) test links between two public universities and one private university (Murai, 1990), now connects over three thousand computers in more than three hundred organizations (Frey and Adams, 1990) and is growing rapidly. The WIDE (Widely Integrated Distributed Network) Internet is an infrastructure network that provides IP (Internet Protocol) connections to sites at member organizations, operates the JUNET backbone, and provides Japan's primary IP connectivity with the rest of the research world. WIDE evolved out of a project team assembled in late 1986 "to design the future JUNET," and is oriented toward internetworking, open systems, and global connectivity (Murai, 1989; 1990).

The main function of JUNET is communications. Member organizations connect to backbone sites to exchange electronic mail and NetNews. Electronic mail addresses and headers strictly follow de facto international Internet standards. JUNET NetNews includes over 120 Japanese language newsgroups in the fj.* category (Lunde, 1992), as well as over a thousand of the worldwide USENET newsgroups in English. The official kanji code of JUNET is JIS X0202 (equivalent to ISO2022), which is based on JIS X0208. As a two-byte, seven-bit character code it can pass through the narrow straits of the seven-bit ASCII world most of the time. Although the total number of people outside Japan who read and write Japanese may be minuscule in comparison to the domestic population, the designers of JUNET wanted connectivity both in Japanese as well as in English and other languages. Global connectivity is a basic operating principle of JUNET/WIDE.

When JUNET began, international communications had to be in English or romanized Japanese. This played a role in Japan's reputation as a "black hole" of information, with so much flowing in and so little flowing out. Jun Murai has described how network design and development are not just a matter of link protocols and addressing but an approach to the "total computing environment," which in the case of JUNET also involved kanji support in a windowed user interface to the messaging systems. Once Japanese researchers could communicate easily with each other in Japanese through JUNET, then the amount of public traffic, as well as JUNET membership, also increased dramatically and has continued to grow. A non-Japanese participant complained at a conference that the increased public NetNews traffic in the fj.* category was all in Japanese. Jun Murai replied, "Ah, but we have done our part in making it accessible. Learning Japanese is still up to you" (Murai, 1989). Fair enough.

Until 1984, individuals who wanted to use their personal computers for telecommunications had to go to systems abroad, such as the information utilities CompuServe and The Source (now defunct) in the United States. The only way to get to these systems, other than direct international dialup, was through KDD's (Kokusai Denshin Denwa, an international telecommunications monopoly in Japan) Venus-P public X.25 data network. Nevertheless, despite the high costs for individuals, and despite the fact that telecommunicating with others through U.S. systems meant telecommunicating in English rather than in Japanese, pasocom tsuushin had captured the imagination and enthusiasm of a segment of the growing market of personal computer users, and it began to take off.

By the end of 1984, there were more than a half dozen bulletin board services (BBSs) operated by individuals and hobbyist clubs. All but a couple were in English (Shapard, 1991). In the years since, grass-roots BBSs in Japan have boomed. Today there are well over a thousand public access personal computer BBSs in Japan, and all but a dozen or so are in Japanese (MBM, 1991). Most personal BBSs in Japan are isolated islands in the datastream, where the only access is direct access, but a few networked exceptions exist, such as the small but active FidoNet-Japan group (Yamada, 1990) and other efforts based on unique locally developed automatic "porting" mechanisms (e.g., Electronic Networking Forum, 1991). The kanji character code of the little islands dotting the seas of the Japan BBS realm is Shift-JIS, the kanji character code of Japanese personal computers.

From 1985, multiuser commercial subscription or usage fee systems began to emerge in the electronic seas of the large Tokyo and Kanto plain market, where 30 percent of the total population of Japan lives. Some of the earliest, largest, and still active, are ASCIInet (1985), operated by the people who brought us Nihongo MS-DOS and Shift-JIS kanji, JALNET (1985) of Japan Air Lines, EYE-Net (1985) of the Fujimic media group, PC-VAN (1986) of NEC, and Nifty-Serve (1987) of Fujitsu. PC-VAN and Nifty-Serve each claim user memberships in excess of 300,000 and compete for market dominance. Meanwhile, outside of Tokyo and in collaboration with local government and local business, regional online systems such as COARA (1985) in Oita Prefecture (Kyushu) have emerged to serve local communities. All of these larger systems have active online communities and extensive public data network connections, with members coming in from other parts of Japan and the world. But the only access is direct access, and even this can be a problem as a result of the seven-bit biases of not just a few so-called international public data networks in the world beyond Japan.

Off in the research lab, with a high-resolution graphics terminal on your desk, a powerful UNIX workstation nearby, and the rest of the world just a TELNET command away (for direct access) through high-speed IP links, it is easy to lose track of the very different context and environment of users of personal computers and 2400bps modems. Likewise, people who are just learning how to turn their machines on and off (and are in near total befuddlement with the "techno-esoterotica" of modem commands and serial parameters that confront them before they can even get to a local BBS or commercial information utility) are busier looking for something simple that works than taking much interest in solutions that adhere to international standards for greater connectivity. They are still trying to get their oars in the water and get that most basic connectivity of direct access.

## *Isolation in Tokyo Bay: The Case of TWICS*

TWICS started as one of the first half-dozen pioneer BBSs in Japan in 1984, evolved into one of the first public multiuser systems in 1985, and became one of the first commercial operations in 1986 as an electronic mail and conferencing service (Shapard, 1986; Shapard, 1990; Quarterman, 1990). TWICS has been a settled island in the electronic seas of Japan for a long time, globally accessible through international public data networks since 1986, and networked through intersystem mail since 1987. But despite a certain international notoriety as one of the few systems in Japan so accessible, TWICS still has only a small population of a little over 700 members who remain isolated from most of their Japanese neighbors while at the same time enjoying connectivity with much of the rest of the world online. Language and character codes have played a role.

### *The Nature of the Community and Its Communication Needs*

Most of the early members of TWICS were also active members of systems abroad, which they accessed directly through international public data networks. So, while they lived and worked in Japan, they were also members of what was then a fairly small community of people using pasocom tsuushin internationally. TWICS has always had a multicultural population, from the system operators to the members who used the system. There is no cultural majority, although the largest segment of the population is Japanese (Shapard, 1990).

The early TWICS members perceived themselves as part of a larger international community of those people who had to spend much time going out through international public data networks to various systems abroad. Eventually, this changed into a perception more like that of the JUNET members, where they were able to share information and ideas with people elsewhere in the world from the convenience of a local "home" system as part of the world of electronic mail networks.

JUNET originally served researchers in the same field but in different organizations in Japan and gave them a channel to professional colleagues elsewhere. The initial focus was international as much as it was domestic.

However, like other pasocom tsuushin system users in Japan, TWICS members primarily come online as individuals, rather than as part of a site or organizational membership. Unlike JUNET, the host computer functioning as the community or "home" mailbox server was not located within the domain of an organization that most of the members shared while not online.

The first domestic applications of pasocom tsuushin in Japan were to provide locally that which had only been available overseas, that is, communications facilities and databases. And, having this locally meant having it in Japanese, the local language. Since almost everyone who could read and write Japanese, and who had personal computers and software that supported kanji, were Japanese living in Japan, the initial focus was absolutely local and domestic. TWICS, with its use of English and its multicultural community, has been a notable exception.

The social and communications needs of the TWICS community members are as various as the membership is eclectic. Some members use the system primarily for professional purposes, especially for communications with people abroad. In this they are like the users of JUNET, of which TWICS (twics.co.jp) is also a member. The researchers on JUNET are members of the larger world academic and research community, and their communications are basically professional in nature, transcending national boundaries. Some members of TWICS join primarily as a result of personal social needs, like those of other pasocom tsuushin systems in Japan. Their purpose is to pursue personal interests and meet new people. Public access BBSs and commercial online services provide them with a wider scope and new ways to meet people they would otherwise not have the chance to meet.

In the middle are the open community areas, the public topics and conferences open to all TWICS members. A lot of crossover happens as those with primarily professional needs benefit as well from the local relationships they develop and those who came online for personal social needs benefit from increased professional relationships as the world online opens up to them.

Attempts to get Japanese language communications started in public areas on TWICS have largely failed, mainly because the Japanese members prefer to use English, or, rather, prefer not to be "segregated" according to language. The most extensive Japanese language communications on TWICS have been through the use of romaji, which can be done with ASCII characters and therefore supported by any terminal. Writing and reading Japanese through romaji, however, is "unnatural" to Japanese speakers, something akin to English speakers writing and reading in phonetic script. One TWICS member, a mathematician, told me about some statistics he had run during his graduate school days for a linguist doing a study (unpublished) on the readability of Japanese with different scripts (Yoneda, 1986). The use of only kana or romaji, rather than the usual kanji/kana combination, resulted in readability rates not all that much different from those of Japanese reading English, a foreign language.

Language segmentation leads to isolation, and communities require a shared common language. In the case of TWICS, with its multicultural community and their international communication needs, this shared common language has been English. TWICS is at the same time a part of the larger global community of people interconnected through the matrix, as well as a tiny English language island isolated from the communities of the nearby islands of the Japanese language pasocom tsuushin seas.

### *The Influences of Language, Character Codes, and Environment*

JUNET researchers required international connectivity from the beginning. While they may be using English internationally and kanji domestically, it all has to travel through the same channels. Japanese researchers and others abroad need kanji as well, even if they are located in some seven-bit ASCII environment. So, JUNET adopted kanji character code standards compatible with international standards. The developers of JUNET had the need as well as the means to get past the original limitations of their computing environment and the local-solution directions of their domestic computer industry.

Personal computer users, on the other hand, also wanted kanji in their interfaces and online communications, but they were stuck with Shift-JIS. As with so much else in the personal computer field, Japanese

BBSs and then larger commercial systems evolved without paying much attention to developments for other kinds of machines and environments. When the host systems were home-rolled BBSs running on personal computers, the total Japanese language environment was based on Shift-JIS, the only game in town as far as they were concerned. This has caused more than a few problems along the way, as the larger commercial and regional systems run not on MS-DOS personal computers but on larger multiuser systems. So, while their own internal system kanji codes were EUC or other JIS variations, their interfaces had to be Shift-JIS. And they have often continued this right down into the heart of their messaging systems and databases, thereby rendering themselves into character code isolation.

Although TWICS has from the beginning supported Japanese in various forms, global access requires the lowest common denominator in terms of assumptions about the kinds of character codes supported by the personal computers and terminal software of the member users, and this means one-byte, seven-bit ASCII and the use of English, the most widespread global language, as the system default language. An early design principle was "global access with local flavor." Global access meant access with no assumptions about the user equipment, and therefore a reliance on international standards such as English and ASCII. Local flavor meant Japanese place names and style. The paradox is that, despite the names, if things are in ASCII or English, then they are not Japanese.

The computing environment, and the influence on kanji character codes resulting from the design philosophies of their makers, has also had a strong influence on the use of kanji on TWICS, or the lack thereof, as the system evolved through several platforms. Short of getting all the Japanese computer makers and software developers to agree on a single, common, kanji character code, and one that is compatible with international standards, which in turn must be designed to recognize the needs of languages like Japanese, the short-term solutions involve filters between the different kanji codes. There are three ways to go about doing this.

***Kanji Filter Solution 1—on the Personal Computer*** One approach to the filter solution is to use terminal software on personal computers that handles the interactive filtering at that end of the connection.

This is the most efficient solution, as most personal computers are underutilized in the first place when used as mere terminals. However, it makes assumptions about the terminal equipment on the user side and introduces unfair biases for Japanese language users. Many of the more popular Japanese terminal programs, especially those for the ubiquitous NEC PC-9801 series, only support a limited number of kanji character filters, and for some reason, EUC/DEC Kanji and JIS X0208/X0202 are often not supported. Also, a lot of people use convenient and low-cost Japanese language waapuro for telecommunications, which have even fewer options.

***Kanji Filter Solution 2—on the Host Computer*** The second solution is to put interactive filter mechanisms on the host side, generally in the terminal drivers or some other interactive filter.

This can be done fairly easily with UNIX terminal drivers, and most UNIX systems in Japan now support interactive kanji filtering in their terminal drivers, although the manufacturers do it in different ways. It is a different story with operating systems like VMS.

The first time TWICS went to Nihon DEC with this problem in 1986, their reply was to offer the source code and a license to change it. In 1988, TWICS went back to Nihon DEC again to argue that their support of interactive kanji character code filtering in the VMS terminal driver would help increase their market share in business use and to propose a solution (Rikitake, 1988). The reply was that such solutions did not fit into DEC's own global strategies, regardless of how big the lucrative Japanese market was with all those people sailing Japanese pasocom and waapuro in their local and incompatible seas of Shift-JIS.

***Kanji Filter Solution 3—File-to-File*** The third solution is to forget about interactive kanji filtering altogether and use kanji file filters instead. In addition to the kanji character code file filters provided by Nihon DEC for Nihon VMS, for example, there are several others available as shareware to solve this problem (Lunde, 1992). They are generally written in such a way that they can be compiled and run on various platforms, including those that do not have other Japanese language support, and thereby allow any host system to handle kanji messages and file exchange.

If someone with a personal computer wants to send a message in Japanese and their own machine and software only support Shift-JIS, they use an editor or word-processing program to write the message and save it in a file. Then, they connect it to the host system or server and transfer the file through a file transfer protocol like Kermit or XMODEM or ZMODEM. After that, they run it through the filter utility on the host system to convert it to a kanji code that will work. For example, they convert it to JIS X0208/X0202 if it is going through a seven-bit data path, or to someone on JUNET, or in a Japanese language newsgroup on USENET. Finally, they send it or post it. It can be a little bit easier to display kanji messages, as the filters can direct their output to the terminal screen, but it still requires a couple extra steps.

In addition to the "native" EUC/DEC Kanji code of the VMS system, TWICS also has EUC/DEC and Shift-JIS in jCaucus, over 120 Japanese language USENET newsgroups with JIS X0208/X0202, and links with others systems where either EUC/DEC or JIS X0208/X0202 must be used, depending on the site. It is small wonder the networkers in Japan stick to the most familiar kanji codes of their own favorite electronic island. The more is not the merrier.

### Global Directions: Building Bridges for the Future

The issues raised in this chapter regarding multibyte character codes are not unique to Japan, but are shared by other people in East Asia who have writing systems derived from Chinese han ideographs. Chinese speakers use *hanzi* (Chinese form of han ideographs) of various sorts in the People's Republic of China (PRC), Taiwan, Hong Kong, and elsewhere. Korean speakers use a combination of some *hanja* (Korean form of han ideographs) along with their own phonetic alphabet *hangul,* derived from hanja in shape but similar to Arabic or Hebrew in the method of indicating syllables. And Japanese speakers use a combination of kanji (Japanese form of han ideographs) along with their own syllabic kana scripts, katakana and hiragana (described earlier), as well as some romaji (roman characters like the English alphabet) and Arabic numbers.

The classical Chinese han character set, with is origins over 4,000 years ago and its often complicated fonts for up to 50,000 ideographs, has evolved into several variations today for writing and computing (Burkley, 1989b):

### *People's Republic of China*

- Up to 7,500 hanzi (with simplified 6,763 hanzi)
- GB 2312 two-byte code
- Input through phonetic conversion and radical composition

### *Taiwan*

- Up to 15,000 hanzi (PRC simplification not recognized)
- Proposed three-byte code
- Input through phonetic conversion and radical composition

### *Korea*

- Up to 4,000 hanja (for formal writing)
- Up to 40 hangul
- Ministry of Education two-byte code
- Input through hangul composition and hangul to hanja conversion

### *Japan*

- Up to 7,500 kanji (with 1,945 taught in school)
- Up to 100 kana in addition to romaji and Arabic numbers
- Various two-byte codes
- Input through kana to kanji conversion or romaji tokana/kanji conversion

One-byte character codes are clearly inadequate to accommodate the needs of the speakers of East Asian and most other languages with writing systems derived from origins other than those of European languages today. The 128 basic one-byte, seven-bit ASCII codes are only useful for English and languages that use the same character set, with no additional diacritics such as accent marks and so on (see Mason, chap. 12, regarding problems in networking with French and German character sets). Enhancements that add support for another 128 characters through the use of the full eight-bits of a one-byte code remain local, not international, solutions, as they will display differently in other contexts.

Two-byte character code solutions, such as those proposed by ANSI (Unicode), ISO (ISO2022), and various computers makers like DEC, are also the subject of great controversy and debate (Burkley, 1989b; Sheldon, 1991), even with their potential room for 65,536 character codes. The problems are both technical as well as political.

The primary technical problem with two-byte character codes is that up to 40 percent of the 65,536 possible spaces are lost by setting aside all codes that include C0 and C1 sequences that could result in a character being interpreted as a control code (Sheldon, 1991), a problem illustrated by Japanese Shift-JIS. There is just not enough space then left to include all the various hanji of Chinese, the hanja and hangul of Korean, and the kanji and kana of Japanese as separate language character code areas. People trying to use these languages together would need to switch in and out of different language character code modes, and additional codes would be needed to indicate the language mode of a series of text. One proposed solution to this problem is to unify the han characters into one basic set where they overlap, and then support the local variations in separate areas. Other solutions involve using more bytes in the basic character code, such as the three-byte code proposed in Taiwan, or a four-byte code such as ISO10646, which would support over four billion spaces. And this is where the politics come in.

The Chinese agree to a two-byte international standard with a unified set of han characters, but the Koreans and the Japanese have problems with it (Sheldon, 1991). Language and culture are closely related, and writing systems that have been in use for nearly two thousand years can take on a rather mystical quality regardless of the origin. Despite the ancient Chinese origins of Japanese kanji, the Japanese consider their kanji to be Japanese, and not just a subset of han characters. And they resent the assumption that they should continue to work with compromises forced onto them for compatibility with the short-sighted vision and narrow assumptions of a field that is only now beginning to recognize that ASCII is not enough for the world.

## Conclusion

In conclusion, as we sail the electronic seas and explore, settle, and develop the virtual world online, we face many of the same issues that our ancestors have faced in the past as their cultures collided with those of others, and as they discovered whole civilizations built upon vastly different assumptions. We can learn from parallels in history and lessons gained through research in various fields and disciplines as we design the new environments in which we work and communicate. Or else we can stumble along in the blindness of our own narrow biases and wonder why this technology leaves us isolated from others rather than living up to its promise of greater connectivity.