IBM History of Far Eastern Languages in Computing, Part 3: IBM Japan Taking the Lead, Accomplishments through the 1990s

Kurt Hensch, Toshiaki Igi, Masumi Iwao, Akira Oda, and Toru Takeshita

This article describes the coordination of worldwide efforts in IBM that were launched in the 1970s to ensure implementation of Far Eastern language requirements with IBM products, in order that IBM would maintain its leading role in the IT industry.

Although the first emergence of computerized Kanji that IBM Japan developed in the early 1970s was a breakthrough,[1] this progress in capabilities was limited and did not satisfy the entire Asian market. At the high end, newspaper publishing systems that computerized Kanji were developed.[1] However, these required a sizable amount of equipment and software at considerable expense, affordable only by large companies.

The IBM 2245 Kanji printer, a first step for general-purpose products in 1971, could not adequately meet all market requirements for Kanji data processing. The small number of single dot-pattern type font print positions, and the cumbersome data entry via punched cards, limited the chances for wider acceptance. The cost for providing sufficient memory to accommodate more varieties of Kanji characters and sizes would have been prohibitive at that time. Further, the selected printing technology did not allow providing more than 16 characters per line.

IBM Japan, through its daily sales activities, however, was continuously under pressure from customers for Kanji data processing. The sales organization decided to ask the IBM Corporation to make Kanji capabilities a standard feature in all its future products, rather than developing them as special features as had been done with the IBM 2245 Kanji printer. Thus, IBM Japan made a formal request of IBM's product development divisions to develop a Kanji capability.

Editor's Note

This article—along with Part 1, "Requirements and Initial Phonetic Product Solutions in the 1960s," on pp. 17–26, and Part 2, "Initial Efforts for Full Kanji Solutions, 1970s," on pp. 27–37 in this issue—originated as a condensed excerpt from K. Hensch, *Research and Development in IBM, History of Far Eastern Languages in Computing*, 2nd private edition, Roehm TYPOfactory GmbH, Sindelfingen, Germany, 2004; ISBN 3-937267-03-4 (available from Amazon.com).

Steps to meet market requirements

Because of the rising market needs and new technologies in hardware and software, the IBM Corporation decided to establish a product development laboratory in Fujisawa, west of Tokyo. It started with 92 members initially, all having the competence and experience in developing custom products for the Japanese market, which included handling Far Eastern languages in computing.

Kanji linguistic study

During 1972–1973, IBM Japan's development laboratory conducted several linguistic studies on the Japanese language to find parameters that could facilitate improvements in system performance and human factor aspects. The goals of the studies were as follows:

- Analyze the character usage frequency to help minimize the access time for identification of a character on the keyboard layout during data entry.
- Define the keyboard layout to reflect the

 Published by the IEEE Computer Society 1058-6180/05/$20.00 © 2005 IEEE

most natural and most commonly used way of finding a character, whether by stroke pattern or by pronunciation with subsequent selection from groups, based on the Hiragana sequence, or alphabetical method, or any other methods.
- Determine the optimized data entry method—evaluating the trade-offs of keyboard (with or without shift keys); stylus; stroke pattern; and Katakana mnemonic input.
- Compare the efficiency of Kanji selection via pronunciation of individual characters versus lookup from an electronic dictionary. The latter often results in multiple Kanji.
- Define the minimum dot pattern matrix size to ensure sufficient resolution even for the most complex Kanji character.

Language support for future systems

Several developments took place during the early 1970s that contributed to Kanji language support. For example, during 1972 through 1974, Masumi Iwao was assigned to several IBM product development laboratories in the US to participate in the development of the IBM 3767 Katakana Communication Terminal.

A designated Kanji project was initiated in 1973 by the marketing division of IBM Japan. It was upgraded to a regional IBM Asia/Far East (A/FE) project in the following year, and staffed mainly by personnel from IBM Japan's Product Development Laboratory.

The IBM Japan marketing division also started research projects for Kanji processing jointly with a few customers in 1975. Valuable experience had been gained with the IBM 2245 Kanji printer system, enhanced by the findings of the Kanji linguistic study.

In the early 1970s, both Toru Takeshita and Toshiaki Igi were assigned to product line management at IBM World Trade Corporation Headquarters primarily to define national requirements for FS (the official code name for a next generation of future systems under development). Both men frequently visited IBM's Data Systems Division and General Products Division development laboratories to explain the hardware and software requirements for Kanji.

Programming languages

Takeshita, as Advanced Systems product manager in A/FE headquarters, provided preliminary specifications—for the inclusion of Kanji in programming languages—to projects in IBM's software development laboratory in Santa Teresa, California.

Takeshita's work led to the introduction of the Kanji data type to handle Kanji data and Kanji comments or remarks in coding. Keywords and reserved words in programming languages and file names remained in English. Katakana was no longer contemplated, because the experiment with the PL/I language keywords in the IBM System/360 era had not proved useful enough. Katakana was not even considered in the Japanese Industrial Standards (JIS) for Cobol and PL/I languages. Because subsequent programming languages—Pascal, C++, Java, and so on—were mainly used for systems and engineering programming, very few people were interested in using Katakana keywords because they did not make coding significantly faster or easier.

New product developments

In the early 1970s, a whole variety of new technologies in hardware and software had become available. This offered opportunities for developing novel products, and a full-fledged product development effort commenced in 1977 under the overall coordination of IBM Japan. All of IBM's data processing know-how and technologies throughout the world would be used. Particularly noteworthy major new technologies included:

- Virtual memory, a software concept for automatically anticipating information exchange between the limited-size main memory and large mass storage devices. Fritz-Rudolf Guentsch at Technical University in Berlin had first invented the principle in a PhD dissertation,[2] but it was not published in the scientific press of the 1950s. This concept was implemented in IBM's System/370 product line and known as the dynamic relocate function. A historical note on Guentsch's work has been recently published in the *IEEE Annals*.[3] It contains references regarding the re-invention of virtual memory,
- *Data entry by keyboard and display* linked under a processing function control.
- *High resolution, all-points-addressable (APA) technology*. Applied in displays and printers (ink jet, laser, and photo composing), this technology changed the world from the character-based approach to an addressable-points approach, which provided the solution for Kanji and Far Eastern languages representation.

As top priority for the overall development of Kanji capability, the design concept empha-

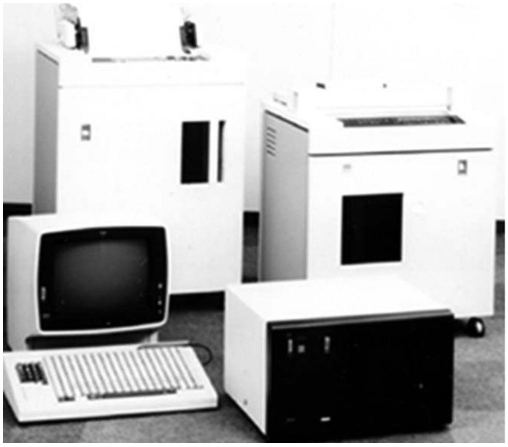


Figure 1. IBM 3270 Kanji Information Display System. (Photo by Toshiaki Igi.)[4]

sized the need for compatibility of Kanji systems with existing alphanumeric systems, inclusive of Katakana, instead of isolating any Kanji system as a dedicated and unique system.

This concept inevitably called for coexistence of new Kanji codes and existing alphanumeric codes, without interfering with each other in entirely intermixed usages. It became a fundamental requirement for all database and data communications (DB/DC) application software of general-purpose systems.

Another important design objective for IBM Japan was to ensure that Japanese customers could perform easy transitions from Katakana data to Kanji through software.

To meet these design objectives, development of each Kanji system component, both hardware and software, had to be carried out by those IBM development laboratories that were responsible for the individual system element. Thus, IBM sites located in various countries across the world became involved:

- Product development of core hardware underlying the Kanji control unit, Kanji display unit, and Kanji printer that constituted the IBM 3270 Kanji Information System was carried out by IBM Japan's Product Development Laboratory, with strong assistance from the IBM development lab at Kingston, New York.
- The IBM San Jose development laboratory developed the IBM 3800-2 Kanji printing subsystem.
- DB/DC and high-level language development, which showed the use of the Kanji data type and comments, was carried out at the IBM development laboratories in Santa Teresa, California, and Hursley, UK.
- The IBM 5924-T01 Kanji Keypunch was developed in collaboration between the IBM Japan Product Development Laboratory and the IBM Toronto, Canada, development laboratory. The IBM Raleigh Keyboard Product Engineering Laboratory developed the Kanji keyboard subassembly.
- IBM Japan's Program Development Center in Kawasaki, called the Tokyo Programming Center, developed a series of Japanese language processing software programs.

For all these widespread efforts, the marketing division of IBM Japan consistently played the role of leader and negotiator of the projects.

New Kanji system announced

Upon successful completion of these development activities, IBM announced a new Kanji system in September 1979. It consisted of the IBM 3270 Kanji Information Display System (see Figure 1), the IBM 5924-T01 Kanji Keypunch, the IBM 3800-2 Kanji Printing Subsystem, and associated programming support in form of system software and utility programs.

The system was implemented to function under the control of either an IBM System/370 processor, virtual memory functions of an IBM 303X CPU, or under an IBM 8100 Information System. It applied the design concept of Kanji system compatibility with existing alphanumeric systems, including Katakana, and was no longer a unique system dedicated to Kanji. Its basis was a well-designed code structure using two bytes per Kanji character in an extended alphanumeric, extended binary-coded data-interchange code (EBCDIC) encoding system.

IBM Japanese character set

An IBM Japanese character set was defined and published in August 1979 for the new IBM Kanji system.[5] Its defined 7,190 characters became a superset of JIS C 6226 (the Japanese Industrial Standard *C 6226-1978, Code of the Japanese Graphic Character Set for Information Interchange*). Based on a Kanji usage study conducted by the IBM Japan marketing organization, 388 characters were added to the JIS character set. The IBM Japanese character set provided 4,370 character code positions for additional characters that users could define and add for their application needs. An IBM Corporate Standard (*C-S 3 3220-024, IBM Japanese Graphic Character Set, Kanji*) eventually formalized this IBM Japanese character set.

IBM 3270 Kanji Information Display System

The IBM 3270 Kanji Information Display

System in Figure 1 consisted of the IBM 3274 Model 52C Control Unit, the IBM 3278 Model 52 Display Unit equipped with a 12-shift key Kanji keyboard, and the IBM 3283 Model 52 Printer.

The IBM 3274 Model 52C Control Unit contained a UC.5 (UC stands for Universal Controller) microprocessor and microcodes for supporting the Kanji functions. The microcodes interpreted the IBM 3270 Kanji data stream (the IBM 3270 data stream architecture was extended for Kanji data) from the host system and sent it to the display or the printer. The microcode also tracked the shift- and data key depressions on the Kanji keyboard attached to the display unit and generated the two-byte EBCDIC Kanji code for the IBM 3270 data stream.

The IBM 3278 Model 52 Display Unit displayed the Kanji characters in a 16 × 16 dot matrix, which was generated by large-scale integration (LSI) logic and the Kanji font stored in ROM, using the two-byte EBCDIC Kanji code. The display screen could display 960 Kanji (40 Kanji in a row and 24 rows), or 1,920 alphanumeric characters in a 7 × 9 dot matrix (80 characters in a row of 24). Any mixture of Kanji with alphanumeric data was done on a field by field basis.

The Kanji keyboard subsystem was common to the IBM 3278 Model 52 Display and the IBM 5924-T01 Kanji Keypunch. This keyboard differed from the Kanji keyboard of the former IBM 2245 Kanji system, but followed the similar concept of a multishift operation. It featured 12 shift keys and 254 data keys. For direct data entry, the keyboard layout included a mixture of 2,567 alphanumeric, Katakana, and most commonly used Kanji characters. In addition, this keyboard provided the input capability for those characters that were not in the layout, by entering the character reference number. The conventional alphanumeric typewriter keyboard layout was integrated into the center of the keyboard for a single-byte alphanumeric data entry operation.

The IBM 3283 Model 52 Ink Jet Printer composed each Kanji character in a matrix of 28 × 28 dots. This product was developed by the IBM Japan development laboratory, using the electrostatic ink jet technology developed by the IBM Office Products Division in Lexington, Kentucky (which later became the independent Lexmark Company). This technology had a 240-dots-per-inch resolution, producing high Kanji print quality.

In summary, the IBM 3270 Kanji Information Display System facilitated interactive operations between a host system and a display unit—specifically, between Information Management System (IMS) and Customer Information Control System (CICS) Kanji applications and a display unit with a Kanji keyboard. The IMS and CICS were enhanced to support the IBM 3270 Kanji data stream.

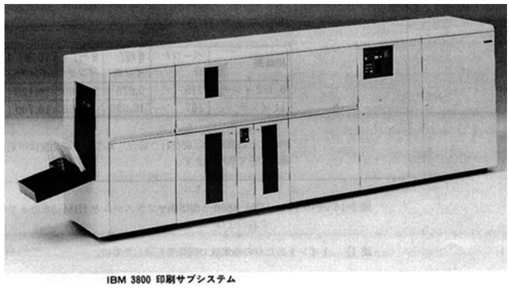


Figure 2. IBM 3800 Printing Subsystem.[6]

IBM 5924-T01 Kanji Keypunch

The IBM 5924-T01 Kanji Keypunch was based on the IBM 029 and 129 card punch units and used the same 12-shift-key Kanji keyboard subsystem as the IBM 3278 Model 52 Display Unit. The keyboard subsystem was connected to the IBM 029 or 129 card punch via LSI logic and generated two-column punched holes on IBM 80-column cards, using ROM tables for code conversion.

IBM 3800-2 Kanji Printing Subsystem

The IBM 3800-2 Kanji Printing Subsystem was the Kanji version of the IBM 3800 Printing Subsystem based on photocomposing and developed at the IBM San Jose, California, development laboratory. It was enormous, almost the size of a trolley car, featuring an ultrahigh printing speed of 10,000 alphanumeric lines per minute (see Figure 2). This printer utilized an electrophotographic drum, the same technology as in most recent plain-paper copiers, printing on continuous paper forms.

Utility and system programming support

Software support expansion for Kanji data type handling was made available by IBM mainframes' Operating System/Virtual Storage (OS/VS), Disk Operating System/Virtual Storage Extended (DOS/VSE), the IBM 8100 Information System's and Distributed Processing Programming Executive (DPPX) software. Kanji data type handling support was provided for database and data communication by IMS/VS and CICS/VS software.

The programming languages PL/I and Cobol were enhanced to handle Kanji characters.

IBM Japan developed several utility pro-

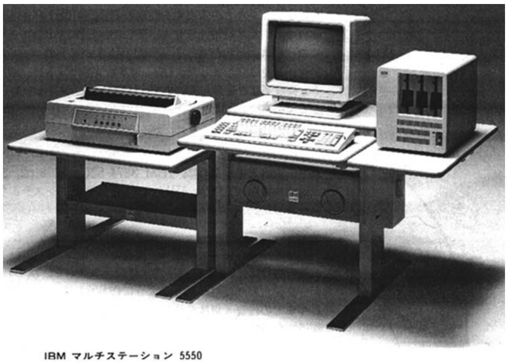


Figure 3. IBM 5550 Kanji multistation.[6]

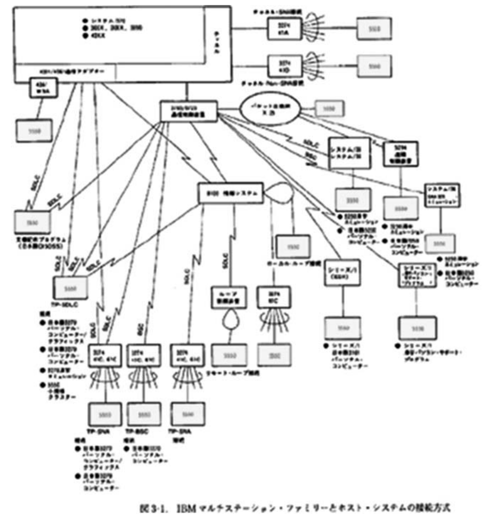


Figure 4. Large system with IBM 5550 workstations.[6]

grams such as one for printing data sets containing Kanji characters, a code conversion program between JIS and IBM Kanji codes, and another conversion program from Katakana data to Kanji.

IBM Japan's Tokyo Programming Center provided a very flexible sort/merge program that covered four sorting methods (basic Kanji type, the Japanese dictionary method, and character look-up by both pronunciations, *on-yomi* and *kun-yomi*).

IBM Kanji system enhancements

An IBM 3200 Kanji printer was announced in June 1982 as a smaller and more economic printer than the IBM 3800-2. This printer used the same electrophotographic drum as the 3800-2 printer. However, because IBM could not produce it to its customers' timing and price constraints satisfaction, this was an OEM product from Hitachi.

The IBM 3273 Model 53 Kanji Printer was announced in October 1981 as another printer for the IBM 3270 Kanji Information Display System. It was an impact printer, capable of printing on multiple-copy print forms and employed a 24 × 24 wire dot matrix print head from the Oki Electric Company.

In July 1980, IBM announced a Japanese Language Processing System software program. Its key elements were these Kanji application programs:

- Kanji document input/editing program for the Kanji keyboard
- Kanji document input/editing program for the Katakana keyboard with Kana to Kanji conversion capabilities
- Kanji document composition program
- Print program for the IBM 3800-2 Kanji Printing Subsystem and IBM 3283 Kanji Printer

These application programs were supported for IBM's Multiple Virtual Storage (MVS) system and Virtual Machine (VM). IBM Japan's Tokyo Programming Center in Kawasaki performed the development.

IBM Kanji System/34

An IBM Kanji System/34 was announced in the fourth quarter of 1979, developed by the IBM Product Development Laboratory in Rochester, Minnesota. This system provided Kanji interactive display capability similar to the IBM 3270 Kanji Information Display System. Its 12-shift Kanji keyboard subassembly was the same as in the IBM 3270 Kanji Information Display System.

Workstation business unit

A workstation business unit was organized in 1982 to develop the double-byte character sets (DBCS) for the IBM PC, collect customer requirements, make requests to IBM development divisions, and market workstation products. The Kanji character set for a PC double-byte code page called Shift JIS was introduced in the Japanese PC industry.

IBM 5550 Kanji Multistation

In 1983, the IBM 5550 Kanji Multistation (see Figure 3) was announced. It consisted of a desktop CPU with up to three drives for 5.25-inch 720-Kbyte floppy disks, a keyboard, and a monitor. A wire dot matrix printer was also offered, providing single-style, single-size Kanji fonts in a 24 × 24 dot matrix. The IBM 5550 found wide use as multiple online workstations in large computer systems (see Figure 4). However, it was still rather expensive for the general public as a stand-alone PC. A character set and code reference manual was published.[7]

The IBM 5550 came also with a Japanese word processor, which was based on the IBM Displaywriter operating system. Later, for the Japanese PC DOS of the IBM 5550, a non-IBM—and very popular—Japanese word processor called Ichi-Taro and developed by the Justsystem Company was also made available.

Data entry for Kanji was done by a Kanji PC DOS keyboard input front-end processor, which provided a Kana (user-selectable Katakana or Hiragana) to Kanji (pronunciation to ideograph) function. The conversion was accomplished through a software dictionary and conversion algorithm on a single Kanji character basis, or on a Kanji word basis.

Double-byte Technical Coordination Office

To better support the implementation of the national language requirements in East Asian countries, Toshiaki Igi established the Double-byte-character-set Technical Coordination Office (DTCO) in 1982. Located at the IBM Japan Product Development Laboratory, a group of dedicated engineers from IBM Japan was made responsible for planning, guiding, and coordinating the development of all East Asian language features at all IBM laboratories worldwide.

National language support

The DTCO developed an IBM strategy and implementation proposal for national language support (NLS) by establishing the technical definition of DBCS programming support for Japan and Asia. These definitions documented the required support based on the application need of each software component. IBM product development headquarters at Harrison, New York (at that time), recognized the need for initiating an organized approach to provide systemwide support for DBCS, and it decided to conduct a corporate-wide task force with major software development laboratory participation. The task force developed modifications of IBM MVS systems software based on the DBCS support level definition.

The DTCO presented the NLS implementation proposal to IBM Corporation top management in May 1984. In a favorable response, IBM set aside a significant amount of money as an approved development fund for five years of work, to modify IBM S/370 MVS software for DBCS support.

Further, to ensure automatic implementation of NLS in any future product, an appropriate IBM corporate policy for NLS was documented and issued internally as an IBM Corporate Instruction, declaring this support to be a fundamental policy.

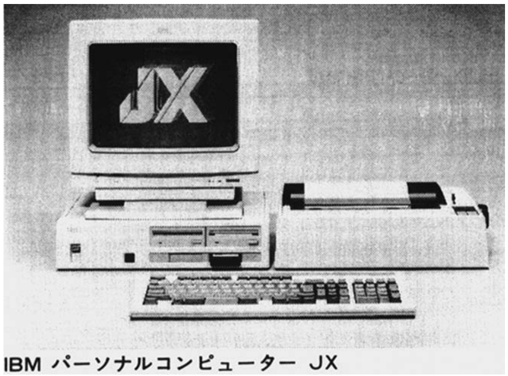


Figure 5. IBM Personal Computer JX.[6]

So that IBM programmers worldwide could develop DBCS-enabled software products, the DTCO developed and published various English guidebooks for design, testing and application.[8,9] DBCS competency centers were organized in five locations at IBM programming laboratories to provide on-site technical support, and DTCO sent assignees to these locations.

IBM Personal Computer JX

Not too long after the appearance of the personal computer in the US, IBM Japan followed suit in 1984 with the introduction of its IBM Personal Computer JX (see Figure 5). Different models came with a tabletop CPU that had one or two 3.5-inch floppy disk drives, or even a hard drive. The IBM Personal Computer JX was one of the first PCs in Japan that introduced 3.5-inch floppy drives.

The JX Word Japanese word processor was using ATOK, a phonetic input to Kanji/Kana text conversion program and a key part of Ichi-Taro. A plug-in circuit card housed the fonts.

The customary keyboard cable was the means of basic connection to the system unit. Wireless infrared technology was used as a supplemental connection for convenience, offered as an advantage. This advantage, however, turned into a handicap if the keyboard was not properly pointed toward the CPU.

The printer offered a new thermal technology with electrode print wires thinner than those in mechanical devices. The requirement for special thermal-sensitive paper was a drawback. Printing was later replaced by a wire dot matrix. Again, Kanji was available only in single-style, single-size fonts.

Large advertisement posters from IBM Japan, announcing the first IBM Kanji Personal Computer, showed a student holding the CPU under his arm, and the keyboard sticking out of his backpack! (A precursor of things to come?)

Twenty years after our work on a Kanji solution had started, our dream had come true. It was exciting to buy a JX in Japan and be able to do Japanese word processing in any part of the world.

Operating systems supporting Kanji

On the software side, the IBM 5550 Multistation and the IBM JX PC ran on a Japanese-unique PC DOS called Kanji PC DOS. Although the file system was compatible with English PC DOS, the application compatibility between the two systems was not very good, primarily because of the conflict between Shift JIS encoding and the English encoding scheme. To solve the problem, IBM Japan developed DOS/V (the "V" stood for Video Graphics Array). This epoch-making contribution eliminated the need for providing a Kanji character font generator in output devices (displays and printers). Competitors (Fujitsu, NEC, Toshiba, and others) followed and adopted this method.

Kanji requirements for IBM sites outside Japan

The need for Kanji processing outside Japan emerged first at the IBM Corporate Standards office where Kurt Hensch had worked since 1977 as program director of standards. One of his responsibilities was the publication of IBM internal standards. Previously, these documents were contracted out to graphic designers and publishers for printing. He changed this to an internal task, from camera-ready masters to making them electronically available over the IBM internal network.

For the development of future Kanji products, all IBM sites worldwide had to be provided with IBM standards on coded character sets of Asian languages. Consequently, Hensch designed a layout that would fit on both the international standard A4 paper size as well as on the US standard size of 8 1/2 by 11 inches. This was a relatively easy task, but getting documents with good quality Kanji into the hands of all IBM product developers was not.

As a first step, Hensch ordered one of the IBM 5550 Workstations for his office and used it until retirement. However, the quality of dot matrix single-style, single-size Kanji fonts was inadequate.

Because the photo composer developed by IBM's Systems Development Division in Boulder, Colorado, had just been installed at a few selected IBM sites, Hensch started using one of them to produce IBM standards documents in-house. However, this method meant that the Corporate Standards office would have to distribute all these documents by mail, which was not acceptable. Print quality was excellent, but the host system had no Kanji fonts. Further, the production of camera-ready masters still required the additional step of chemically processing the documents at the site of the photo composer.

Here a new IBM 4250 high-resolution printer (600 dots/inch) offered a less complicated solution. This machine, developed during the 1970s by IBM Germany's development laboratory, could produce brilliant, high-quality camera-ready masters, using black paper coated with aluminum powder. Hair-thin electrodes eroded tiny dots and exposed the black paper to compose the letter in a dot matrix so fine that it could be recognized only under a magnification glass. Unlike the photo composer, this technology did not require any additional treatment of the printed output.

Hensch installed one of the IBM 4250 printers in his office and produced a large number of documents for topics other than Kanji. Kanji was still a problem, but a new one dwarfed it, although it may have been an unjustified cause for alarm: When Hensch heard that deposits of aluminum were found in the brains of deceased Alzheimer patients, he immediately stopped using the IBM 4250 and moved to another office. More practical reasons for discontinuing the use of this machine, however, were the requirement for the special paper, and the fact that it was not available at all IBM facilities. In any event, marketing of this printer soon ceased when the superior laser printers became available.

The solution for the online retrieving of IBM standards on Kanji at all IBM sites eventually came with the IBM 3820 Laser Printer. This machine was installed in every IBM site at multiple locations in large numbers. The lack of Kanji fonts on IBM systems outside Japan was overcome by implementing a unique method with embedded graphics.

The first IBM standard for a Japanese Kanji character set was ready to be published in 1979, and the English text was ready. However, there was no Kanji font library on any IBM host system except in Japan. The solution was a page segment (PSEG) feature of IBM's internally used text-processing software ISIL (later announced as an IBM product under the name BookMaster). It allowed inserting such page segments (graphics) between texts.

The PSEG feature meant that we needed to produce one page segment for each of the thousands of Kanji characters, which appeared to be a monstrous task. However, two brilliant young programmers of IBM Japan, Takashi Ogura and Akio Kido, accomplished it. They devised a programmed semi-manual method to convert the Kanji fonts called FONT3820 into PSEGs and inserted those into the ISIL source file of the document. It worked, although formatting of the file brought the host system almost to its knees.

Year	Product
1966	IBM System/360 and IBM 1403 Printer
1974	IBM System/3270, IBM 3277 Terminals, IBM 3287 Printer
1977	IBM System/32
1978	IBM System/34
1981	IBM System/38
1983	IBM System/36

Figure 6. IBM products with Thai language capability.[10]

Other East Asian countries make progress

From the mid-1960s to the mid-1980s, IBM development capabilities mean that China, Korea, and Thailand also experienced progress in computerizing their respective languages.

China

During the two decades beginning in the mid-1960s, a few products with Chinese language capability were marketed by IBM, but only in Taiwan. Specific information proved to be very difficult for us to obtain for purposes of this article, because the present generation of local IBM employees has only a vague recollection of DBCS products marketed in Taiwan before 1984.

Likely products that were involved are the IBM 3283 C52 Ink Jet Printer, with the IBM 3278 Display Unit and a large multishift data entry keyboard, adapted from the IBM Japan Kanji system. The data entry operator had to remember the positions of all characters precisely; presumably, there must have been a certain arrangement pattern on the key tops.

Because the IBM PS/55 dot matrix printer could print Japanese Kanji, it is likely that it could have been adapted to print Hanzi. Likewise, IBM CICS were enabled to accept, process, and output 3270 DBCS data strings.

Korea

Hensch's proposed implementation for Hangul on the IBM System/360 was documented in the IBM Special Engineering Catalog of RPQ offers (Request for Price Quotation; a simplified method IBM used to expedite the development of special products). Therefore, it is conceivable that some products with Korean language capability were marketed by IBM between the mid-1960s to the mid-1980s. However, we were unable to obtain relevant documentation for this article.

Thailand

Supported by Thai RPQ designs of IBM 026/029 keypunches and IBM 056/059 verifiers for data entry, a number of products with Thai language capability were marketed by IBM Thailand (see Figure 6). Because a Thai National Standard for a coded character set was not published until 1986, IBM Thailand used the internally defined Thai EBCDIC encoding.

In 1984, IBM Thailand established a national language development (NLD) team. Its initial task was to develop Thai language support for controllers, printers, display terminals, and specifically required software such as terminal emulation. Later, through the early 1990s, the team assisted numerous IBM development laboratories in implementing the Thai language requirements on IBM products. The NLD team has been providing not only development coding and testing support but also educational material, test procedures, translation, and localization services.

East Asia DBCS standardization

A number of factors, beginning with IBM corporate standards, played a role in the standardization of numerous Far East character sets.

Start of Kanji standardization efforts

During the development phase of the new IBM 3270 Kanji Information Display System (see Figure 1), which was announced in September 1979, the standardization of Kanji coded character sets became urgent. A standing procedure at IBM was that the announcement of any new product required the official sign-off from designated key functions at IBM corporate headquarters. One of those was the IBM corporate standards function. Consequently, the IBM new Kanji system required the preparation and publication of an IBM corporate standard for the IBM Japanese coded character set.

This was only the beginning of a still-ongoing effort, which has stretched over decades. The IBM DTCO was given the responsibility to prepare IBM standards for countries in the Far East. The work had to be based on previously

Date	IBM Reference Manuals and Standards	Japanese Industrial Standards (JIS)
1972-04	IBM 2245 Kanji character and code set (7525 characters) Reference Manual N:GA18-1018	
1978-01		JIS C 6226 defined (6,802 characters) (later renumbered as X 0208)
1979-07	IBM C-S 3 3220-024 (7,190 characters) published for Host-code (glyph image based on Seirei *1 and "Shinjigen" *2) (28 symbols and 360 Kanji more than JIS) Reference Manual N:GC18-0611	
1983-03		JIS C 6226 revised (6,877 characters) — 71 symbols and 4 Kanji characters added — 22 pairs of Kanji code point swapped — 244 characters glyph image changed
1986-08	IBM C-H 3-3220-024 revised (7263 characters) — PC code definition published *3 — 69 symbols and 4 Kanji added *4 — 11 characters glyph image changed *5 Reference Manual N:GC18-0785	
1987-03		JIS C 6226 renumbered to JIS X 0208
1990-09		JIS X 0208 revised (6,879 characters) — 2 Kanji added — Heisei font used for standard publishing
1990-10		JIS X 0212 defined, Japanese Supplementary, (6,067 characters)
1992-11	IBM C-H 3-3220-024 revised (7,265 characters) — 2 Kanji added — 14 characters glyph image changed *5	
1993-03	IBM C-H 3-3220-127 published, extended UNIX Code (EUC)	
1995-01		JIS X 0221, UCS (Japanese version of ISO 10646)
1996-08	IBM C-H 3-3220-133 published for DBCS-PC (NewJIS Sequence) added	
1997-01		JIS X 0208 revised, Shift-JIS and Internet encoding added
1999-03	IBM C-H 3-3220-024 revised, Host code (14,823 characters) revised to accommodate Unicode Japanese Subset	

*1 The glyph image is defined by Government order
*2 One of popular Japanese Kanji dictionaries
*3 IBM PC implemented the code earlier as internal process code
*4 Two characters were already in IBM 1979 definition
*5 Adjusted to follow "Seirei" glyph image

Figure 7. DBCS standards, Japan.[11]

established ground rules for character encoding, available in four fundamental IBM standards documents:

- *IBM Corporate Standard C-S 3-3220-002, Extended BCD Interchange Code EBCDIC*
- *IBM Corporate Standard C-S 3-3220-019, Coded Character Sets Implementation*
- *IBM Corporate Standard C-H 3-3220-050, Registry, Graphic Character Sets and Code Pages*
- *IBM Corporate Standard C-S 3-3220-102, Double-Byte Character Set (DBCS) Terminology and Code Scheme*

IBM and national documentation

Although the character sets in Western languages are of finite numbers, East Asian languages could be labeled open ended. National authorities in East Asian countries define their sets in groups of usage. While most characters find their way into a group that will be encoded for computer applications, there are still many others of lesser usage that remain unencoded, perhaps forever. Revisions are made all the time, which must be reflected in IBM standards.

Figure 7 shows the list of IBM standards documents, which needed to be published on coding topics for Japan, and that reach into the present. Each of these documents underwent at least one, but usually more, revisions to accommodate changes. Figure 8 shows IBM standards documentation activities for China and Korea. Both figures are an indication of how often the standards required revision.

Japanese coded character sets, IBM

Within the large number of code points of the 16-bit structure (meaning double bytes), IBM defined many code pages on a global scale. To be responsive to all national requirements in the future, it was necessary to cover all languages and special characters, plus controls.

Relying on the most comprehensive Japanese dictionary available, IBM Japan defined an EBCDIC code page, assigning two 8-bit bytes for each Kanji character. At this time, unlike the arbitrary code assignments that were used for the IBM 2245 Kanji system, IBM considered important logical aspects for the new set.

IBM standards for Japanese coded character sets emerged in two versions, one for host systems, and the other for PCs. Newly defined JIS

	Korean	Traditional Chinese	Simplified Chinese
IBM Standard	1985-03 Defined: Korean DBCS-Host code 1989-10 C-H 3-3220-124 published DBCS-Host and DBCS-PC 1992-09 C-H 3-3220-125 published DBCS-Host and DBCS-PC revised 1993-11 C-H 3-3220-128 published 1997-09 C-H 3-3220-030 revised DBCS-PC, industry standard code 1999-04 C-H 3-3220-030 revised Euro symbol added	1985-01 Defined: Traditional-Chinese DBCS-Host code 1989-10 C-H 3-3220-126 published DBCS-Host and DBCS-PC 1992-01 C-H 3-3220-126 revised DBCS-Host and DBCS-PC 1993-11 C-H 3-3220-129 published 1994-01 C-H 3-3220-131 published BIG 5 Code (industry standard PC) 1999-04 C-H 3-3220-126 revised Euro symbol added	1985-01 Defined: Simplified-Chinese DBCS-Host code 1992-11 C-H 3-3220-130 published DBCS-Host and DBCS-PC 1993-11 C-H 3-3220-130 revised DBCS-Host and DBCS-PC 1994-06 C-H 3-3220-132 published 1996-02 C-H 3-3220-020 published for GBK Code
National	KS X 1001:1992, KS X 1002:1991, KS X 1003:1993; KPS 9566-97 (North Korea)	Big Five, Big Five Plus, CNS 11643-1992, CCCII (75,684 characters)	GB-Series (11 documents) see Lunde's book (Ref. 2) page 72

Note: The bottom row provides only a lead to Korean and Chinese national standards documents.
For any detailed information, consult with Ken Lunde's book (Ref. 4), and also the books on "Unicode" (Ref. 5, 6).
Japanese readers can obtain more details

Figure 8. DBCS standards, Korea and China.[11]

character sets were included as subsets. Several subsequent revisions to the IBM standards became necessary to recognize changes or additions to Japanese industrial standards, and those too came in different versions.

Japanese Industrial Standard

There was no Japanese Industrial Standard in 1971 for a Kanji character set when IBM announced the IBM 2245 Kanji printer. IBM filled this void by pioneering the definition and publication of its *Kanji Character Set and Code Reference Manual*.[12]

On a national scale for Japan, a JIS standards committee began to develop a standard Kanji coded character set for the country in 1976. Since then, Kanji JISs have continuously been revised and expanded. Together with representatives from industry, universities, publishing companies, and other experts in this area, qualified and designated IBM employees from the Kanji development team at the Fujisawa lab, and the DTCO continuously and actively participated in the JIS committee work. The goal of this committee was to allocate a unique pair of 8-bit bytes to every Kanji character on designated code pages.

Toshiaki Igi was a member of the first JIS Kanji Industry Standard committee, which published JIS C 6226-1978 (subsequently changed to JIS X 0208, Japanese Industrial Standard *C 6226-1978, Code of the Japanese Graphic Character Set for Information Interchange*). His experience from having assisted in defining the IBM 2245 Kanji printer character set greatly benefited the JIS committee. Other key contributions from IBM were made later through Tsuneo Oda, secretary of JIS X 0208:1997, Japanese Industrial Standard *X 0208:1997 7- and 8-bit Double Byte Coded Kanji Sets for Information Interchange*, and Akira Oda, group leader for defining the Shift JIS encoding from the de facto standard into a JIS standard in JIS X 0208:1997. These people and unnamed others deserve high praise for their efforts in the development of JIS standards for Kanji coded character sets. Readers interested in more detail may wish to consult additional sources.[13–17]

Development of industry standards was tedious and difficult, because the IBM committee representative would frequently be in conflict between the committee objective and IBM's product direction and computer architecture. The task required considerable work and patience. The following examples illustrate the complexities of some of the problems encountered over the years:

- The JIS committee defined the standards for information interchange primarily for communication line use, based on the 7- and 8-bit International Standard Code for Information Interchange scheme. JIS standards for coded Kanji character sets, and the code extension technique of mixing one-byte and double-byte code, could not be implemented in the internal code for operating system processing in a mainframe, Unix machine, and PC. This internal code implementation required a code conversion from the JIS standard by code mapping or table translation.
- In JIS C 6226:1983, Japanese Industrial Standard *C 6226-1983* (changed to *X 0208-1983* in 1987), *Code of the Japanese Graphic Character Set for Information Interchange*, the code points for 22 pairs of Kanji characters were swapped, affecting 44 Kanji characters. This created a code point compatibility problem. IBM could support both JIS C 6226-1978 and JIS C 6226:1983 for information interchange since a table translation method was used. IBM kept consistency with the EBCDIC Kanji internal code and

customer database code integrity. Some other mainframe manufacturers could not support the JIS C 6226-1983 standard because they implemented direct code mapping from the JIS 1978 standard to their internal Kanji EBCDIC code.

- For PC internal code, Japanese Control Programs for Microcomputers (CP/M) and Japanese Microsoft DOS introduced the Shift JIS method in early 1980. IBM was one of the first companies to adapt Shift JIS, and implemented it for the IBM 5550 in 1983. IBM assigned code page 932 for Kanji Shift JIS in the *IBM Standard for Graphic Character Set ID*. Shift JIS for the PC became the de facto standard until it was included in JIS X 0208 in 1997.[15]

The DTCO team followed the JIS activities very closely, to ensure that future IBM products would support all of the nationally standardized Kanji. To continuously meet this objective throughout the following years, they produced and maintained an internal IBM document titled *IBM Coded Character Set for Japan*.[5,7,12,18] Derived from IBM's Kanji usage study for data processing, this document always included additional Kanji characters over and above the JIS Kanji set. IBM declared that compatibility would be maintained with the JIS committee, by always making the IBM character set a superset of the JIS standard when the JIS standard changed. See Figure 7 for an overview of the history.

Eventually, as a result of these activities, an official JIS Kanji dictionary emerged.[19] It lists all Kanji characters with their pronunciations, along with their double-byte code assignments in all relevant code schemes. A designated special section provides the means with which any Kanji can be found by its basic stroke pattern. This book is a masterpiece example of years of hard, tedious, and detailed work.

Chinese coded character sets

For China, the DTCO prepared two IBM standards for Chinese coded character sets. Assisted by native employees from local branches in Taiwan and Hong Kong, one document addressed Traditional Chinese (the center column of Figure 8). A second effort, undertaken with native employees from mainland China, resulted in an IBM standard for Simplified Chinese (rightmost column of Figure 8). Every Chinese Hanzi was allocated to a unique pair of 8-bit bytes on designated code pages. Subsequent revisions to the Chinese standards were required, resulting from changes made by national standards committees.

According to author Ken Lunde's book, *CJKV Information Processing*,[20] the Kanji characters used in Hong Kong and Vietnam are based on Traditional Chinese, but they have their own unique standards.

Korean coded character sets

The standardization of Korean character sets was comparably complex. The leftmost column of Figure 8 summarizes IBM and national documentation. The Kanji characters used in Korea are based on Traditional Chinese. North Korea has its own standard, which differs from the South Korean document. Each of the Jamo, Hanja, and the large number of Hangul permutations were given a unique, two 8-bit byte assignment on designated code pages.

Thailand coded character sets

After IBM Thailand had established the NLD function in 1984, its representatives began working with the Thai Industrial Standards Institute (TISI) and other vendors to define national standards. This work resulted in the issuance of TIS 620-1986 on the *Thai Coded Character Set Standard*. A revision was published in 1990.

IBM internal standards for Thai were developed in compliance with national and international standards. They are registered as code pages number 874 (PC-ASCII) and number 838 (EBCDIC), as part of *IBM Standard C-H 3-3220-050, Registry Graphic Character Sets and Code Pages*.

Unlike IBM Korea, IBM Thailand decided to standardize only code pages for the set of components in a single 8-bit byte structure, but not the symbols themselves. IBM Thailand wanted to stay in single-byte format, because the DBCS PC at that time required special hardware like display adapter cards and fonts. IBM Thailand developed a small rendering engine (a print program routine) to display composed symbols from variable-byte encodings, which simulated a single-byte data stream from the viewpoint of application programs. This approach enabled them to use many English application programs without modification.

Business size was relatively small in Thailand, and the first requirement was to be able to handle Thai data. Special efforts for translating all of the programs were to be avoided.

International code page system

National character set standards for all languages had to be prepared in concert with an international effort conducted by the International Organization for Standardization

(ISO) and the International Electrotechnical Commission (IEC) in a joint technical committee, ISO/IEC JTC1/SC2, for a global structure of code pages documented in ISO/IEC 10646. The key standard on Far Eastern languages, *ISO 10646 Information Technology—Universal Multiple-Octet Coded Character Set,* was not published until 1993.

IBM Japan code experts closely coordinated the Kanji coding with IBM's globalization department, which had worldwide responsibility for coded character sets standardization. Careful consideration was essential, so that national code page assignments would not conflict with each other. Thus, only certain areas of code points were selected for Kanji. This was the beginning of an ongoing international effort.

ISO/IEC 10646 is a fundamental standard, potentially affecting all parts of information technology. However, it specifies only coded character sets, not a complete system for text representation. It provides the basis for internationalization, but does not give a perfect solution to the problems in this field.

Outlook universal code

First published in 1993, the Universal Multiple-Octet Coded Character Set System—Unicode—is a character set specified by a consortium of major US computer manufacturers. Its primary objective was to overcome the chaos of different character sets in use with multilingual programs, and facilitate the development of software for international applications.

Unicode assigns a unique number to each character in each of the world's major languages to eliminate any conflicts or incompatibilities during information interchange. It started on a 16-bit basis (UCS-2), but a later version was based on 31 bits (UCS-4). With a clever system of mapping onto different 16-bit planes, it has the potential to cope with more than one million characters.

The international code page system is complex. For detailed information, consult Lunde.[20] Other recommended sources are Fowels[21] and the Unicode standard.[22] The latter is a voluminous book of more than 1,200 pages. Other details are available at http://unicode.org.

Major changes through new technologies

As explained in Part 2,[1] suitable technologies did not exist in the early 1970s to encode, store, process, and print the enormously large numbers of complex characters of Asian languages, with a performance compatible to languages with limited character sets. Several major breakthroughs occurred during the 1980s, which threw the gates wide open for opportunities in the area of computer-related technology. IBM was not alone, because many companies took advantage of this, and development work ensued. A whole new industry emerged, competing with innovative hardware products and related software support. The problems faced in the 1970s could now easily be coped with. New PC designs started to become available at short intervals. With these, development of affordable, high-quality Kanji systems became a reality. Millions of people in China, Japan, Korea, and Thailand were finally able to utilize the Internet and e-mail like everyone in alphabet-based countries.

Main memory

Integrated circuits on chips had long replaced magnetic core memory. RAM capacity grew ever larger, while processor speed became faster. Interestingly, PC prices steadily drop even while continuously offering increasing power.

Disk storage

Progress in storage technologies virtually exploded. It started with the 720-Kbyte low-density floppy disks, soon to be doubled and quadrupled. Hard disks have become smaller, cheaper, faster, and of greater capacity than anyone could have imagined 25 or 30 years ago.

Printing

Laser technology, fast and noiseless, approached a stage of perfection, which facilitated building machines that outclassed all previous impact printers. The IBM 3820 printer was one of IBM's early, successful laser printer products. Laser printing was soon followed by the technology of ink jet printing. Both allowed printing at much finer resolutions than mechanical wire dot printers could ever offer. High-quality printing, particularly of Kanji characters, became possible, accompanied by a large variety of fonts and at low costs. Ink jet printers, with rapidly dropping prices, especially turned out to be attractive for PC users. These evolutions provoked an abrupt end to impact printing in the mid-1980s.

Same monitor for Latin and for Kanji

Monitors of the early years had their own code generators for Latin character sets, and text-processing applications mushroomed. For Kanji, however, in particular for small PC users,

Figure 9. IBM G40 ThinkPad Japanese keyboard. (Photographed by Kurt Hensch.)[23]

all-points-addressable monitors were needed as well as exceedingly fast access storage to grab letters from fast font libraries and dictionaries.

Good-bye, IBM punched cards

Lower cost monitors connected to processors resulted in the obsolescence of IBM punched cards. Direct data encoding with input into CPUs and memories via computer terminals called workstations became a reality in the late 1970s. Hard-wired and dial-up connections via acoustic couplers and modems emerged, which were the beginning of pre-Internet local, and wide, area networks (LANs and WANs).

Compact discs, memory sticks, and smart cards

Although punched cards are surely a thing of the past, CDs might soon suffer the same fate with the appearance of high-capacity memory sticks and smart cards, usable with both PCs and digital cameras. Attachable to newer PCs via USB connections, they might become a superior competitor for CD drives in the not too distant future.

Data entry method for Japanese

With punched cards having fallen into disuse, data entry methods settled on the use of keyboard layouts that were very similar to the physical arrangement of keys on English-language keyboards. To permit the entry of Japanese text, however, the key tops of workstations, desktop PCs, and laptop computers carried Hiragana symbols in addition to the familiar English QWERTY arrangement. Figure 9 is an example of such a keyboard, the IBM ThinkPad Model G40 that was used for writing this article. Users can choose to enter Japanese text by pronunciation in either Hiragana or English alphabet mode. Interestingly, even many Japanese people use Latin to key in the pronunciation of Kanji, with subsequent selection of the character they are looking for.

In addition to key tops with graphics, keyboards such as the G40's have several additional control keys not seen on western keyboards. For example, a special key switches full-size characters (double-byte) to half size (single-byte) and back. Latin characters, numbers, special characters, and Katakana are encoded in both single and double byte and require this feature.

These enhanced, versatile keyboards need powerful software support to fully handle double-byte-coded character languages. The Justsystem Company was leading this development with its program called ATOK. It uses not only a pure Japanese dictionary (similar to an English thesaurus), but also certain syntax and grammar rules to find the most suitable Kanji character first. (Microsoft implemented this method later in its software called MS IME (IME stands for "Input Method Editor").

The major breakthroughs in Kanji data entry became possible only through the availability of low-cost storage, such as hard disks, combined with fast processing speed. Whole dictionaries can be stored with the advantage of keying in the pronunciation of words that consist of more than one Kanji character and added Hiragana as needed for grammatical reasons.

Thus, the selection of Kanji by pronunciation on a single character basis effectively came to an end. As a consequence, other previously ventured approaches were abandoned, such as selecting a Kanji character from a huge template, either by a stylus or by multishift keyboards.

Zhuyin	Hanyu Pinyin	Tongyong Pinyin
注音	漢語拼音	通用拼音
ㄔㄨㄢ	chuan	chuan
ㄔㄨㄤ	chuang	chuang
ㄔㄨㄟ	chui	chuei
ㄔㄨㄣ	chun	chun
ㄔㄨㄛ	chuo	chuo
ㄘ	ci	cih
ㄘㄨㄥ	cong	cong
ㄘㄡ	cou	cou
ㄘㄨ	cu	cu
ㄘㄨㄢ	cuan	cuan
ㄘㄨㄟ	cui	cuei
ㄘㄨㄣ	cun	cun

Figure 10. Alphabetized Zhuyin sound list (partial).[24]

Tone Type	Key to be Pressed
flat tone	space bar
raised tone	number 6 key
tone lowered and raised	number 3 key
tone lowered	number 4 key

Figure 11. Zhuyin tone selection criteria.[24]

Data entry method for traditional Chinese

Data entry for traditional Chinese is a multistep process. Unlike Japanese with its ambiguities of several pronunciations for the same Kanji, Chinese Hanzi basically have only one, which typically remains unchanged even when combined with another Hanzi. Compared to Japanese, this seems like an advantage, but a superimposed tone (flat, up, down and up, or down) will change the pronunciation and the meaning and, therefore, result in a different Hanzi.

The Zhuyin system includes a set of 37 sound symbols (see Figure 10) plus four tone identifiers on the keyboard. It takes into account the melodic Chinese pronunciation for the desired Hanzi. One of the four levels shown in Figure 11 must be selected by pressing the corresponding key.[25]

The entire process of getting one Hanzi on the screen will take from two to five keystrokes. Once the desired Hanzi is found, it is tagged with its designated double-byte code for further processing, display, printing, and storing. A further step of sophistication is in the software that sometimes displays a second Hanzi for a likely combination.

Several keyboard layouts and systems emerged, all based on the English QWERTY physical arrangement. Some use the pronunciation and tone approach, while others rely on stroke-pattern input. Individual keyboards with corresponding key top arrangements are available.

IBM came out with a universal combination of the QWERTY layout, featuring the Latin characters together with three additional Chinese symbols on each key top (see Figure 12). Software support allows the user to toggle between several keyboard layouts to activate and display the desired one on the screen. This is called a soft keyboard, which facilitates data entry but without a physical keyboard.

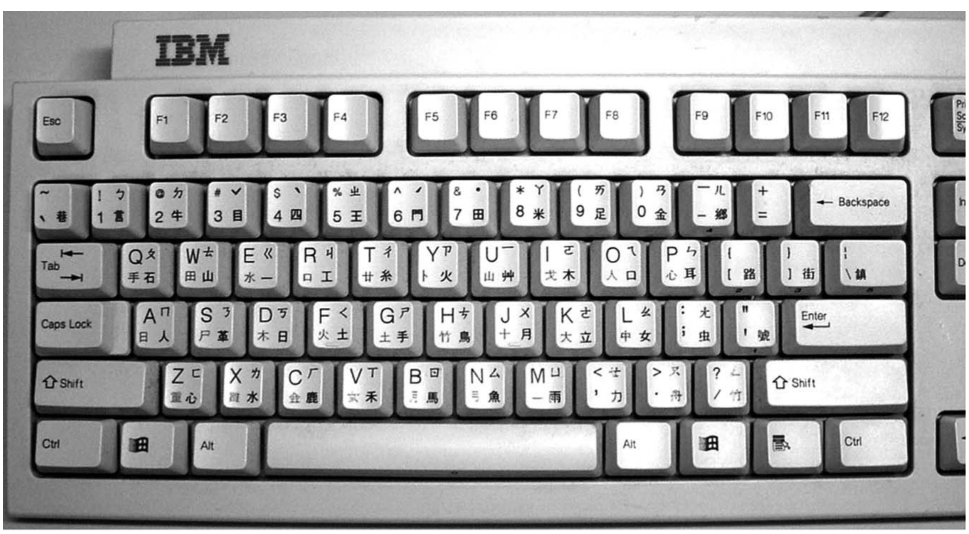


Figure 12. IBM keyboard for Traditional Chinese keyboard.[24] (Photo provided by Satoru Kimura.)

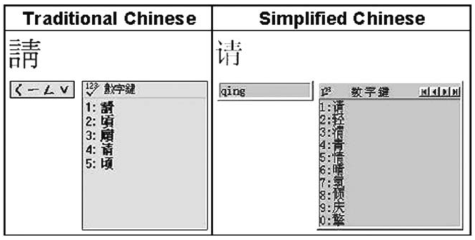


Figure 13. Hanzi data entry screen displays.[24]

請在此頁下面查找你的投票站地址
Check Your Polling Place Address Below

Figure 14. Sample Chinese sentence.[26]

Data entry method for simplified Chinese

Simplified Chinese data entry itself has been greatly simplified, by using only the regular English QWERTY keyboard. The Pinyin list is used directly, but may require as many as six keystrokes. A tone mark selection is not used, however, which results in a larger number of displayed Hanzi candidates (see Figure 13, right box). The upper right buttons permit scrolling through additional selections.

Traditional vs. simplified Chinese

Both boxes of Figure 13 show the Hanzi 請 which means “Please” and is the first character of the Chinese sample sentence in Figure 14.

In the left box, the three designated Traditional Chinese Zhuyin pronunciation symbols of the initially pressed keys appear below the Hanzi, followed by the tone mark.

The right box shows the corresponding simplified Hanzi, with the previously keyed in pronunciation using the alphabetical letters *qinq*.

The large shaded boxes demonstrate that the number of Hanzi candidates has narrowed down to five for Traditional Chinese, reflecting the tone mark entry. However, many more Hanzi must be inspected in the case of Simplified Chinese.

Science fiction in sight

Recent research raises hope for an increase in efficiency for Chinese data entry. The approach is to have the computer know where its user is looking, or gazing. A paper in the *Communications of the ACM*[27] reports an experimental eye-tracking system that could be used for selecting a Hanzi character simply by gazing at it long enough among a displayed group, and then pressing the space bar.

This could eliminate the presently needed last step of character selection, whereby the user must take the eyes away from the screen to press the designated number key.

Data entry method for Korean

The keys in Figure 15, an IBM keyboard for Korean users, show basic phonetic symbols called Jamo, which are consonants and vowels. The Jamo set is a bit larger than the available keys, forcing some Jamo symbols onto the upper shift location in the QWERTY row. When entering two to four Jamo symbols from the keyboard, those are combined under certain rules and arranged into one character print or display position. Once found, the Hangul character is tagged with its designated double-byte code for further processing, display, printing, and storage.

Data entry method for Thai

Unlike the Japanese, Chinese, and Korean keyboards, the keys on the Thai keyboard carry no English alphabet, but only the 87 Thai symbol components. These are the consonants, vowels and tone marks that are used to compose a complete Thai symbol. Numbers are shown in Thai, and only some are placed on the same keys as on the QWERTY layout (see Figure 16).

Figure 15. IBM Korean keyboard.[24] (Photo provided by Satoru Kimura.)

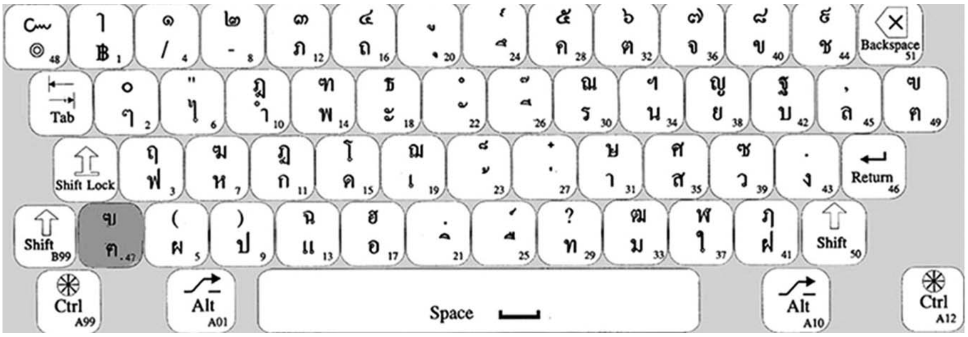


Figure 16. Thai keyboard layout.[10] (Image provided by Ranat Thopunya.)

Unlike Japanese, Chinese, and Korean, the composed Thai symbols (more than 2,000) are not double-byte encoded. Therefore, the inherent programming support as well as translation work is not required. The penalty for this advantage, however, is that English must be used with many application programs. Automatic (program controlled) composing of Thai characters was introduced during 1987–1989.

IBM Thai keyboards are registered under the numbers 190, 191, and 192 in the IBM keyboards standards document. For data entry of mixed Thai and English, the user must switch between the Thai and the English keyboard. Most operating systems allow such language switching by a certain key combination. Arabic numerals are available on the English layout.

Multifont Kanji printers

Since converting its Lexington facility into the independent company Lexmark, IBM stopped making printers under contract for several years. Therefore, IBM PCs were forced to use products from other manufacturers, such as Lexmark, Hewlett-Packard, and Epson. Because these use either laser or ink jet technology, printing fine dot pattern matrices, they can all handle Kanji without any special features or modifications.

Dictionaries and font sets usually reside on hard disks, together with programming routines that handle pronunciation group displays.

Thus, a large selection of type fonts emerged for practically every written language to accommodate not only thousands of characters, but also a multitude of different fonts for each character, thanks to the breakthrough progress in micromemory circuit chips.

Kanji fonts

By exploiting significant advances in high density, low-cost memory technology, the use of high-quality and multiple Kanji fonts became the way of life even for PCs. High resolution ink jet printers attached to PCs can now produce almost the same quality Kanji printing as was possible only by expensive Kanji publishing systems like JPS. Kanji fonts evolved from dot matrix representations of only 16×16 or 24×24 in bitmap format to a range from 8×8 to 96×96 in both bitmap and outline format.[25]

Because the Kanji character set is infinite, a font design editor called Gaiji (meaning outside character) Editor to create unsupported Kanji fonts became mandatory. This capability had to be provided from the advent of Kanji implementation in computing and was part of the IBM 2245 Kanji printer system. Also, today's Windows operating system comes with the Gaiji Editor.

PC software

As an operating system, IBM used OS/2 initially, but later largely replaced it by Microsoft Windows on most IBM and IBM compatibles (clone) PCs.

For Japanese text processing, IBM products can use a variety of non-IBM software products, such as Ichi-Taro and EGWord, both local products purely in Japanese, as well as Microsoft Office with Japanese language support.

Considerable effort went into making Japanese word processing possible on US English–based PCs. The intent was to serve Japanese users in the US so they could avoid the need for buying a second (Japanese version) PC. One such product was a Kanji kit, requiring that the hard drive be partitioned, thus simulating two computers in one PC.

Voice entry of Japanese words and text, voice server for transcription products, bilingual translation, optical character recognition, and Internet versions all emerged after the time frame (1960s through 1990s) of this article.[25]

Continuation of developments

Developments are by no means coming to a halt now. Considerable work is still required to make products ever more user friendly. Specific problem areas to focus on are multilingual applications in PCs, error-free code page structure control, multilingual operating systems, flawless processing of e-mail across different languages, and multilingual Internet applications.[25] Kiyokane and Suehiro[28] provide guidelines for internationalization to Japanese programmers.

Acknowledgments

Karl Ganzhorn, founder and director of the IBM Germany Development Laboratory, inspired Hensch to document the IBM history of Far Eastern languages in computing and provided guidance and invaluable assistance.

Satoru Kimura is a brilliant, young staff engineer with Product Assurance, PC Development, Asia Pacific Technical Operations, IBM Japan Yamato Lab. Assisted by Ken Han, on assignment from IBM China, Satoru was of tremendous help not only to make one understand the complexities of Traditional and Simplified Chinese, but also their data entry implementation. Most of the figures in the Chinese data entry section of this article were skillfully captured on the fly by Satoru during a demonstration.

Ranat Thopunya, manager of the National Language Development and Translation Services Center of IBM Thailand, supplied the input for describing the Thai language implementation on IBM products.

References and notes

1. K. Hensch et al., "IBM History of Far Eastern Languages in Computing, Part 2: Initial Efforts for Full Kanji Solutions, Early 1970s," *IEEE Annals of the History of Computing*, vol. 27, no. 1, Jan.–Mar. 2005, pp. 27–37.
2. F.R. Guentsch, *Logischer Entwurf eines digitalen Rechnergerätes mit mehreren asynchron laufenden Trommeln und automatischem Schnellspeicherbetrieb* (*Logical Design of a Digital Computer with Multiple Asynchronous Rotating Drums and Automatic High Speed Memory Operation*), internal report, Computing Center of Technische Universität Berlin, 1956, and doctoral dissertation, Technische Universität Berlin, D 83, 1957 (in German).
3. E. Jessen: "Origin of the Virtual Memory Concept," *IEEE Annals of the History of Computing*, vol. 26, no. 4, Oct.–Dec. 2004, pp. 71-72.
4. T. Igi, personal documents archives.
5. *Kanji Character Set and Code Reference Manual for IBM Kanji Systems,* N:GC18-0611, IBM Japan Ltd., Aug. 1979 (in Japanese).
6. *IBM Japanese Information Processing System/General Description*, N:GA18-1160-0, IBM Japan Ltd., 1987 (in Japanese).
7. *Kanji Character Set and Code Reference Manual for IBM 5550 Kanji Work Station*, N:GC18-2040, IBM Japan Ltd., 1983 (in Japanese).
8. *IBM System/370—DBCS Application Primer (Enable programs for Chinese/Japanese/Korean)*, GC18-9059, IBM Japan Ltd., Nov. 1987.
9. *DBCS Design Guide—IBM System/370 Software*, C18-9095, IBM Japan Ltd., June 1988.
10. Prepared by the IBM Thailand Globalization Competency Center. IBM Thailand has one of several such centers that IBM has established in selected major markets. These centers are responsible to identify and analyze national requirements that are to be addressed in the development of IBM products.
11. Prepared by the IBM Japan Globalization Competency Center. IBM Japan has one of several such centers that IBM has established in selected major markets. These centers are responsible to identify and analyze national requirements that are to be addressed in the development of IBM products.
12. *Kanji Character Set and Code Reference Manual for IBM 2245 Kanji Printer System*, N:GA18-1018, IBM Japan Ltd., 1973, (in Japanese).
13. Y. Mikami, *The History of Character Codes—Asia*, Kyouritsu Shuppan (Kyouritsu Publishing Co.), 2002 (in Japanese).
14. Japan Industrial Standard, *JIS X 0201:1997, 7- and 8-bit Information Interchange Coded Character Set*, Japan Standards Assoc., 1997 (in Japanese).
15. Japan Industrial Standard, *JIS X 0208:1997, 7- and 8-bit Information Interchange Coded Kanji Set*, JSA, 1997 (in Japanese).
16. Japan Industrial Standard, *JIS X 0212:1999, Information Interchange Kanji Code—Additional Kanji*, JSA, 1999 (in Japanese).
17. Japan Industrial Standard, *JIS X 0213:2000, 7- and 8-bit Information Interchange Coded Expanded Kanji Set*, JSA, 2000 (in Japanese).
18. *Kanji Character Set and Code Reference Manual for IBM Kanji Systems,* N:GC18-0785, IBM Japan Ltd., 1986 (in Japanese).
19. K. Shibano, ed., *JIS Kanji Dictionary*, Japan Standards Assoc., 1997 (in Japanese).
20. K. Lunde, *CJKV Information Processing*: *Chinese, Japanese, Korean, and Vietnamese Computing*, O'Reilly & Associates, 1998.
21. K. Fowels, "Unicode Evolves," *BYTE*, Mar. 1997, pp. 105-110.
22. The Unicode Consortium, *The Unicode Standard,* v. 3.0, Addison-Wesley Longman, Apr. 2000.
23. K. Hensch, personal documents archives.

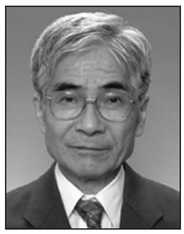
24. Prepared by IBM Japan Product Assurance Department, at Yamato, Japan.
25. K. Hensch, *Research and Development in IBM, History of Far Eastern Languages in Computing*, 2nd private edition, Roehm TYPOfactory GmbH, Sindelfingen, Germany, 2004.
26. Instructions for Chinese voters. This was a flyer that was dropped into everybody's mailbox in San Francisco, California, prior to a local election in 2002.
27. S. Zhai, "What's in the Eyes for Attentive Input," *Comm. ACM*, vol. 46, no. 3, Mar. 2003, pp. 34-39.
28. Y. Kiyokane and Y. Suehiro, eds., *Internationalization Programming—I18N Handbook*, Kyouritsu Shuppan, 1998 (in Japanese).

Kurt Hensch joined the Product Development Laboratory of IBM Germany in 1957. His international career began in 1961 with assignment to IBM World Trade Corporation headquarters, New York. Successive assignments followed at IBM Asia Pacific headquarters, IBM World Trade Corporation headquarters, and ultimately at IBM Corporate headquarters, until he retired as program director of standards in 1992. Hensch holds an engineering degree in precision mechanics from the Ingenieurschule Gauss [School for Precision Mechanics and Communications Engineering] in Berlin, Germany.

Toshiaki Igi joined Masumi Iwao's small IBM Japan Engineering Liaison group in 1966, and which he later took over. He spearheaded the efforts to resolve the Far Eastern language requirements in IBM products and was the first manager of the double-byte character set Technical Coordination Office (DTCO) when it was established in 1982. Igi is now retired from a high-level management position at IBM Japan's Yamato Product Development Lab.

Masumi Iwao, who holds an engineering degree in precision mechanics, worked with Canon Company for several years before joining IBM Japan in 1961. He developed the Katakana feature for the IBM 1440 printer and became the lead engineer of the small Engineering Liaison group in 1964, assisting Hensch in addressing the language problem. "Massi" also played a major role in the development of the first IBM Kanji data processing system.

Akira Oda holds a master's degree in electric engineering from Kyoto University. He and his team at the Globalization Center of Competency, Yamato Software Lab, IBM Japan, made outstanding contributions on double-byte character set (DBCS) standardization. He also provided the summary information on the development of all relevant documents on coded character sets.



Toru Takeshita, a graduate of Kyoto University's mathematics department, joined IBM Japan in 1957 and participated in the development of the first IBM Cobol compiler from 1960 to 1962. He led a programming team to develop the IBM Tokyo Olympic Information System from 1962 to 1964, and worked on software requirement, development, and marketing and research areas. After retirement from IBM in 1991, he was a professor of information science at Chubu University in Japan until March 2004. He is a member of the IEEE Computer Society, and an honorary member of the Information Processing Society of Japan.

Contact the authors about this article at Kurt Hensch kurt@kdd.net; Toshiaki Igi, igit@ny.airnet.ne.jp; Masumi Iwao, miwao@f-h.nir.jp; Akira Oda, akiraoda@jp.ibm.com; and Toru Takeshita, wckbb710@ybb.ne.jp.