A Multi-lingual Coding System Based on CCCII

S.S. Tseng, T.C. Gau, C.C. Hsieh,
C.C.Yang, C.T. Chang, Jack K.T. Huang

Abstract

In this paper, a graphic symbol coding system for multi-lingual software environment will be presented. The structute of this system is based on CCCII (Chinese Character Code for Information Interchange) developed in 1980. The first version of this system was implemented in 1981 and a revision of this system, version 2, was completed in 1983. They are implemented on a Data General S/250 computer.

This system is flexible to adopt any existing coding systems that based on ISO-646 and -2022. At this moment, it has space for accommondating more than 90 7-bit byte standards. Therefore, most of the published 7-bit byte standards may be integrated into this system as wished without system modification. For oriental languages, the character sets collected include those used in ROC, Japan, Korea and mainland China. Besides, the variations for the mentioned characters collected so far are beyond 11 thousand.

The structure of CCCII provides a convenient mechanism for cross mappings among different character sets. Used in conjunction with the attribute data base of characters, the CCDB, this system provides attribute cross-reference access features among different standards. For example, an user can access Janpanese Kanji material through the mandarin pronouncition of corresponding Kanji.

At present, the system structure is fixed but its symbol set is still open. We expect that in a couple of years the characters collected will exceed 60 thousand. This system is not only good for international information interchange, but also valuable for developing multi-lingual software in tomorrow's computer systems.

NEEDED FOR MULTI-LINGUAL CODING SYSTEM

Multi-lingual coding system is needed for international information processing, such as library cataloguing systems. Different language uses different graphic character set, each has their own standard coding system, some of them use single byte, some of them need multi-bytes. Chinese for instance has more than fifty thousands graphic characters, it needs two or three bytes for coding them. Most oriental languages use a lot of chinese characters or Chinese likely characters, the coding systems of them are obviously beyond a single byte. A library store books of different languages, the cataloguing system is then very complicated if we wish to access the different text with different coding systems. What we want is essentially a multi-lingual coding system, it can provide multi-language access capability but without ESC, SI, and SO commands, and still keep the individual standard coding system of them unchanged. The Chinese Charater Data Base (CCDB) developed based on the structure of CCCII has all the capabilities as discussed above.

WHAT IS CCCII

Chinese Character Code for Information Interchange (CCCII) is a coding system for Chinese character issued in 1980 and revised in 1983. It is a three 7-bit-bytes code based on ISO-646 and -2022. The graphic area of a 7-bit-byte only have 94 positions, so CCCII has 94X94X94, i.e. 830,584 coding positions. The first byte (B_1) denotes the position within a section, the second byte (B_2) denotes the section of a plane and the third byte (B_3) denotes the planes.

CCCII is organized such that every six consecutive planes is grouped into a layer. There are sixteen layers all together, but the 16th layer has four planes only, as shown in figure 1.

The front 15 sections of the first plane of each layer, except the last layer, is reserved as the user data area. In each of the rest planes, the first section is used for user control area. By such an arrangement, the total coding positions are divided into several groups as follows:

1. User data area has 1,410 positions on each layer and 21,150 positions in total.
2. User control area has 470 positions on each layer and 7,050 positions in total.
3. Graphic coding area has 6,956 positions on each layer and 139,714 positions in total.

IMPLEMENTATION OF CCCII

Although Chinese has almost 53,000 characters, one third of them are variant forms. A Chinese character may associate with several characters that have different shape but with the same meaning and same pronunciation. Those characters are called the variant form of the original one, and the original character is called the regular or normal form. Not all the regular form characters have variants. Some regular form may have more than one variant forms. An informal statistics tells that there are about thirty five thousands regular form Chinese characters. In CCCII, The regular form characters are grouped into three sets according to their usage frequencies. The most frequently used characters are collected in CB1, the next frequently used characters are collected in CB2, and the rarely used characters are collected in CB3. All the regular forms are allocated within layer 1. The variant forms are allocated from layer 2 through layer 13 right beneath their corresponding form characters. So the codes of variant forms have the same byte 1 and byte 2 as its regular form code, but byte 3 has a fixed displacement, in the value of six or multiplication of six,

from its regular form code. There are only 11 positions for variant forms, exclude the simplified form characters used in mainland China. Thus the number of variant form of a certain character can not exceed eleven, if it did occurred, we choose the most frequently used eleven characters and discard the rest. Actually, right now, we only have six variant forms as a maximum, so there is a lot of free space available for more variant form characters. The simplified forms are allocated in layer 2. Layer 14 and 15 are reserved for other use, and layer 16 is reserved for other languages that can be coded according to ISO-646 standard. In summary, the structure of CCCII has the capability to do the multi-lingual coding by using the reserved space and the allocation method as just described.

INTRODUCTION TO THE USAGE OF CCDB

Chinese Character Data Base (CCDB) is a data base together with necessary software developed base on the structure of CCCII in order to make the application of CCCII easy and efficient. Except for layer 1, the density of the rest layers are very loose, it is not efficient for storage space is concerned. The CCDB did two things to remedy it. Firstly, it condensed the three bytes code to a two bytes code named R94, which is a 16-bit binary code. The coversion formula is listed as follows.

$$R94 = [(B_3 - 33) \text{ MOD}(6)] \times 94^2 + (B_2 - 33) \times 94 + (B_1 - 33)$$

where B_3, B_2, and B_1 are the three bytes of CCCII code,
MOD is an arithmetic function to get the remainder after the operation of $(B_3 - 33) / 6$.

The second thing which CCDB did is that it maintains a number of index files and supplies a certain number of utilities to control those files. The structure of CCDB is shown in fig. 2. and the index mechanism is show in fig. 3.

CCDB also provides attribute files for CCCII, and the most important thing is that it provides a pratical mechanism to implement the multi-lingual coding system. The coding of any language which is based on ISO-646 and -2022 can put in any reserved section or plane of CCCII without change its original code, only simply add on one or two bytes precede it depending on that the original code is two bytes or single byte. CCDB provides a mechanism to do the character searching and code conversion by simply treat them as CCCII itself. How it could be done? Let us illustrated as follows.

JISC-6226 for instance is an ISO standard two 7-bit-bytes coding system issued in 1978. It contains special characters, Latin letters, digits, hiragana, katagana, Greek letters, Russian alphabet and Kanji. There are 6,798 graphic characters all togehter. Among those characters, Kanji is the major part, it contains 6,798 graphic symbols. There are 5,712 Kanji which are exact Chinese characters. The rest 422 Kanjis are more or less the same or likely the same pattern of variant forms of Chinese characters. So we treat the Kanji of JIS 6226 as the variant forms of Chinese characters and put it in layer 13 just beneath their regular forms, the other symbols put in the user data area of layer 13. In the mean time JIS 6226 is also placed in a plane of layer 16 with its code unchanged. But a third byte is added precede the JIS code to extend the code to three bytes and treat them as CCCII. Then the CCDB creates a directory of layer 13 to point to an index file, and all the characters within layer 13 are stored in an indexed sequential file for saving the storing spaces. The structure of the index file of layer 13 contains the stroke number of the character, the

searching code of CCCII, JIS code, and the pronunciation in gana. By using this index file, JIS characters can be searched by its stroke number, the asociated search code of CCCII, or the pronunciation of gana, and the code may be mapped to JIS or CCCII. In general, by use only the stroke count as the search key is not enough, it may produce too many characters as a result of the search. Also, only use the pronunciation for search may produce many results. However, by using both the stroke count and the pronunciation as a combined search key will largely reduce the number of characters as the result. CCDB provides a good deal of software to do this. As another example, the single byte ASCII code is placed in the user area of layer 1.

CCCII provides a good structure for multi-lingual coding system, and CCDB is a realization of the idea. The work is still far from complete, we just step on and hope to reach a more fruitful result in the future.