B 194

# 電子古籍中的缺字問題

## 1. Preface

Hanzi have been used for thousands of years. For example, a rather modern font 隸書 is more than 2300 years old. Just like 北魏江式 said：『世易風移文字改變』is unavoidable。《顏氏家訓 · 雜藝》wrote：『晉宋以來……不無俗字，非為大損。……大同之末，訛替滋生。蕭子雲改易字體，邵陵王頗行偽字……朝野翕然，以為楷式。畫虎不成，多所傷敗。……爾後墳籍，略不可看。北朝喪亂之際，書蹟鄙陋，加以專輒造字，猥拙甚於江南。乃以百念為憂，言反為變，不用為罷，追來為歸，更生為蘇，先人為老：如此非一遍滿經傳……』。From these statements, it is obvious that usually one character may have many variants。Other evidences can be found from dictionaries of different time in history。As an example,《干祿字書》differentiate characters into three categories, 通(general)、俗(popular)、正(normal)。There were also dictionaries especially collecting variants, such as《龍龕手鑑》。Although variants complicated Hanzi processing, their effects are not totally negative. Variants provide usefully hints for identifying versions and verifying contents of ancient documents. So, we can not just neglect or bypass them.

### (1)The Missing Character Problem

Wittern and App stated "In East Asia, the problem of missing character is ubiquitous, from individuals unable to type their own name to universities, companies, government agencies. In Japan, these missing characters are called " gaiji "...... It is clear from our work on electronic Chinese Buddhist texts that even Unicode ( ISO 10646 ) will not significantly reduce this problem "【註一】。

Facing missing character, a popular treatment is to create the missing typeface in the user-defined-area in a code space. This treatment can display the missing character in that computer screen, but it is not a cure for the problem and the price paid is too high to be acceptable. The major drawbacks of this treatment are:

**A. Increase the data entry load manifold.**
**B. It is difficult to manage thousands of missing characters.**
**C. There not enough spaces for missing characters in any existing Interchange Code.**

【註一】 Christian Wittern and Urs App.〈IRIZ Kanji Base : A New Strategy for Dealing with Missing Chinese Characters 〉世界電子佛典會議(EBTI)台北, 1996 年 4 月

**D. Most of the missing characters are variants, and there is not way to handle variants.**

**E. Create many information processing problems, such as matching, sorting, merging of Hanzi.**

**F. Cause texts not sharable.**

### (2).Principles of solving the missing character problem

The central theme of solving the missing character problem is to represent more knowledge of Hanzi into computer so that the computer can use related knowledge to do what we want it to do. Besides, the following principles must be carried out:

**A. Do not sacrifice text sharing property for solving missing character problem.**

**B. The solution should be Fair to all Hanzi from different countries and regions.**

**C. Establish the capability to represent, to input, to search, to share and to manage missing characters.**

**D. Give formal working definitions to character, glyph, font, typeface, etc. so that hey are acceptable to linguistics and Wen-zi-xue. Formalize their relations.**

**E. Establish a database for the attributes of Hanzi.**

**F. Use ISO 8879 SGML to describe Hanzi text files for text sharing.**

**G. Expandability and flexibility of the system must be considered for traditional Wen-zi-xue.**

**H. Keep the working environment within two-byte character code.**

# 2. The Representation of Hanzi Knowledge

## (1) Previous Works

**表一《中文電腦基本用字集》匯集的十一種字彙**

１.莊澤宜,《基本字彙》,廣州中山大學教育學研究所,1930
２.胡顏立,《小學初級分級暫用字彙》教育部,1935
３.教育部,《注音漢字》商務印書館,1935 初版,1961 台一版
４.蔡樂生,《常用字選》英文中國郵報社,1946
５.台灣省國語推行委員會,《國音標彙編》,開明書局,1947 初版,1971 台二版
６.王清波,《國民小學現行國語課本國字初現課次、重現次數之分析研究》, 高雄市政府,1963
７.國立編譯館,《國民小學常用字彙研究》中華書局,1967
８.台灣電信局,《電碼新編》,1967 增訂版
９.星華打字儀器行,《中文打字機新版文字排列表》,台北,1969
１０.世界中文報業協會,《新聞常用字彙》,1970
１１.中南鑄字廠,《常用字表》,台北,1971

## 表二《中文電腦基本用字集》異體字的整理原則

1. 就已有字彙選取,不另創新字。
2. 一字數形,取其簡便者。而不計其本體抑俗體,古字抑今字。古字簡便者從古,如取「[illegible]」不取「禮」,今之簡便者從今,如取「[illegible]」不取「繡」。
3. 一字數形,取其結構適合電腦設計者。如取「略」不取「[illegible]」,取「裡」不取「裏」。
4. 一字數形,取其通用者,如取「拿」不取「拏」。
5. 在世俗上已通行一體,而原字還有其他意義的,則兩者並存。如「尿」、「溺」。

按上列原則所選出之形體,當作「字形」，其他各形體則列為「參照字形」。

## 表三《中文電腦基本用字集》前 500 字之累積使用頻率表

| 累積字數 | 累積頻率 | 累積字數 | 累積頻率 | 累積字數 | 累積頻率 |
|---|---|---|---|---|---|
| 5 | 9.24% | 50 | 32.39% | 372 | +70% |
| 10 | 14.20% | 60 | 35.22% | 472 | +75% |
| 15 | 17.79% | 80 | 39.92% | 500 | 76.27% |
| 20 | 20.54% | 100 | 43.74% | 1000 | 89.38% |
| 30 | 25.13% | 141 | +50% | | |
| 40 | 29.02% | 232 | +60% | | |

註：“＋”號表示「正好超越」，如＋50%表示前 141 字的累積使用頻率剛剛超過 50%

## 表四　A formal representation of Hanzi Glyph in Bakcus Normal Form

〈漢字集〉 ：：＝ 〈漢字〉／〈符號〉

〈符　號〉 ：：＝ 合標點符號、注音符號、兩文字母、阿拉伯數目字及
其他專業符號等, 多少不拘。

〈漢　字〉 ：：＝ 〈字根〉／〈部件〉／〈漢字〉〈定位符號〉〈漢字〉

〈部　件〉 ：：＝ 〈字根〉／〈部件〉〈定位符號〉〈部件〉

〈定位符號〉 ：：＝ 橫連符號、直連符號、包含符號

〈字　根〉 ：：＝ 496 個, 詳見〔表五〕

## 表五　交大中文字根表

圖一:灣字的構成

|  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|
| 1 | 輸入碼(倉頡) | 水廿中人 | 十子 | 人卜一口 | 竹山心 | 手一女 | 土廿手 | 輸入系統 |
| 2 | 中文碼(Big5) | BA7E | A672 | AB48 | AEA7 | AAED | B946 |  |
| 3 | 標準字樣(楷書) | 漢 | 字 | 信 | 息 | 表 | 達 | 交換碼 |
| 4 | 隸書 | 漢 | 字 | 信 | 息 | 表 | 達 |  |
| 5 | 仿宋體 | 漢 | 字 | 信 | 息 | 表 | 達 |  |
| 6 | 明體 | 漢 | 字 | 信 | 息 | 表 | 達 | 字體庫 |
| 7 | 細圓體 | 漢 | 字 | 信 | 息 | 表 | 達 |  |
| 8 |  |  |  |  |  |  |  |  |

圖二、時下電腦系統中漢字相關信息的示意圖

## (2). Definitions

Character is an abstract concept. They are differentiated only by the meanings they carried. For example, a normal form and its corresponding simplified form are considered to be the same character. Characters are identified only by the code assigned to each of them.

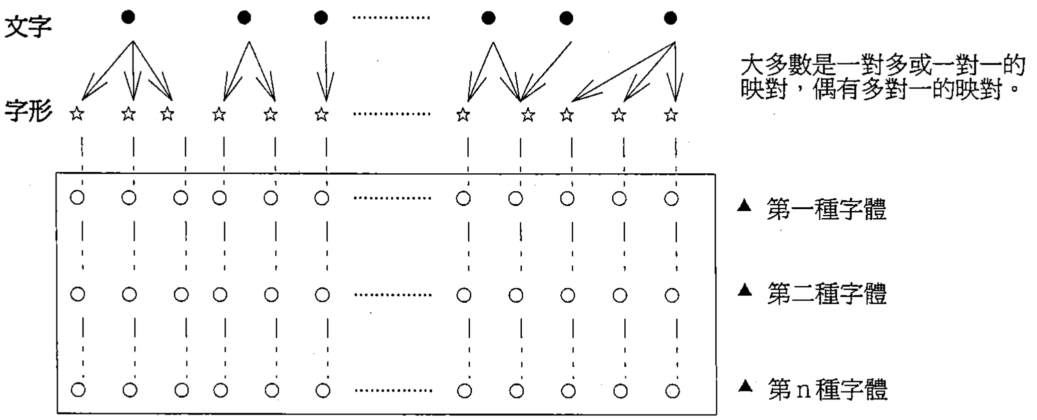
A character may have many glyphs. Glyph is also an abstract concept. They are differentiated by their own structures or skeletons. As in the previous example, the normal form and the simplified form of a character are two glyphs of that character.

It is possible that some characters share one glyph. Since we do not dealing the semantic meaning of characters at this moment, we postpone this problem as a future research topic.

Glyph does not care how nice a typeface may shown, but font cares. A font is a collection of general design rules for a certain style of typefaces. So, font is also abstract in sense. They are differentiated by their design rules. Although font specifies design rules, it has some degree of freedom that allow designers to express their own posture. A font designed may have some parameters to specify the size, the broadness of strokes, the ratio of broadness of vertical and horizontal strokes, etc.. After these parameters have been chosen, then one can see a physical typeface on media.

The described relationship among character, glyph, font, and typeface are shown in〔圖三〕. Please notice that existing Interchange Codes and software systems do not and can not differentiate character and glyph. This phenomena is the source of all troubles of Hanzi processing that we have today.

圖三：字、字形、字體和字樣的關係



此矩陣中的每一個點，表示某一字形在某一字體設計下所呈現的字樣。所以一個文字可以有好幾種字形，一個字形可以有許多字體，而一個字體設計又可呈現不同的大小、粗細、疏密、裝飾特質等等。

圖三：字、字形、字體和字樣的關係

## (3)A Glyph Model

Our glyph model is shown in 〔圖四〕. In this model, the part connected by line 12345is the formal system shown in〔表四〕。lines 6 to 7 show the composition of strokes to produce basic component ( we call it root) of a glyph. The decomposition of glyph into components can be expressed as in the following two equations:

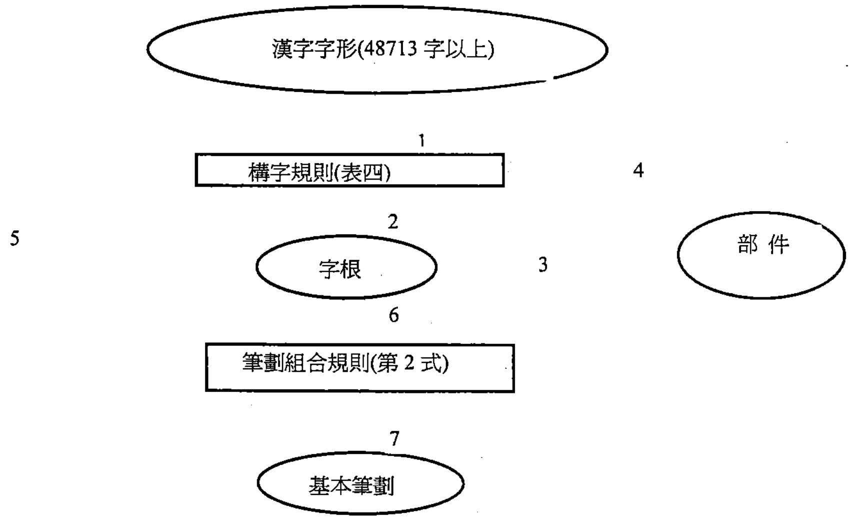
Let G be glyph, R be root, K be component, and T be basic stroke. Let p and s represent position and size , respectively, then

$$G = \Sigma K \ (p, s) \ \ldots\ldots\ldots\ldots \ (1)$$

$$R = \Sigma T \ (p, s) \ \ldots\ldots\ldots\ldots \ (2)$$

Equation (1) is iterative and should be governed by the rules in〔表四〕。

圖四:本系統之構字模式



## (4)A study on the variants and their differentiation

According to our glyph model, variants can be classified into three categories, namely stroke level variants A, component level variants B, and glyph level variants C. Their associated functions are listed as follows:【註五】

**A、** Stroke Level Variant Functions

$A_1$：The change of relative position of a stroke。

$A_2$：Add a “hook” at the end of a stroke。

$A_3$：One stroke is replaced by another stroke。

$A_4$：One stroke more。

$A_5$：One stroke less。

$A_6$：One stroke replaced by two strokes。

$A_7$：Two strokes replaced by one stroke。

$A_8$：A group of strokes replaced by another group of strokes.

**B、** Component Level Variant Functions

$B_1$：A root is replaced by another root. The difference between these two roots is confined in the functions listed in A.

$B_2$：A root is replaced by another root and their different is not defined in A.

$B_3$：A component (a set of roots) is replaced by another component。

$B_4$：One more root

**C、** Glyph level Variant Functions

$C_1$：Roots are not changed , but their relative position changed。

$C_2$：By simplification。

$C_3$：Irregular deformed。

The variants produced by the functions from all A's to $B_1$ are called 分毫字樣（micro-difference variant）。They do not produce new glyph, because they belong to font design variations. Others, such as $B_2$、$B_3$、$B_4$及$C_1$、$C_2$、$C_3$ will produce different glyphs.

In practice, user has the right to select one glyph from the set of glyphs of a character as the authority representative of that character. For instance, PRC will select simplified one for her National Standard Character Set while ROC and Japan may have different choices. This flexibility is allowed in our system. Besides, our system will keep on tracking the mappings caused by different chooses and consider the leftovers as variants for each authority set. Let us

【註五】詳見謝清俊《On the Formalization of Glyph in Chinese Language》世界字體會(AFII)會議，東京，1990年2月

illustrate this idea and its implementation in two examples as shown in 〔圖五〕 and〔圖六〕。

圖五：訛字相關的字形組成

圖六：字形孳乳樹之例

## (5)Attributes

The attributes we collected so far for a character are listed in the following table.

**表六　文字屬性欄位表**　（註:打"*"者，可以重複）

**甲、缺字屬性表**

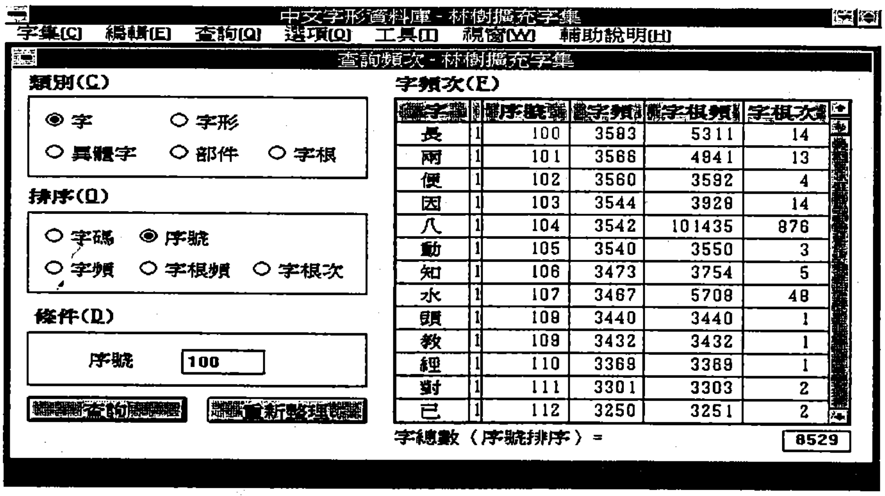
1.缺字統一編號
2.交換碼
3.內碼(造字檔內)
* 4.部首
* 5.筆劃數
6.首筆
7.次筆
8.未筆
* 9.注音
*10.異體字交換碼
*11.登錄日期及修改記錄
*12.提供缺字之各單位欄位(含編號及內碼)

**乙、字形結構屬性表**

1.所屬字集編號
2.交換碼
3.字形碼
* 4.部首
* 5.筆劃
6.首筆
7.分解方式
8.部件一
9.部件二
10.部件三
11.字頻次
12.字根頻次(當用爲字根時)
13.字根次(當用爲字根時)

The glyph database provides two modes of service. One of them let user retrieves glyph, variant, the structure of each glyph, and the family tree derived by a root, a component, or a character. And another mode provides frequency and statistical information for character, glyph, component and root. Example of these two operation modes are given in the following:

| 字形 | 字頻 | 字根頻 | 字根次 |
|---|---|---|---|
| 故 | 1685 | 6011 | 2 |
| 做 | 4326 | 4326 | 1 |



中文字形資料庫 - 林樹擴充字集

查詢頻次 - 林樹擴充字集

類別(C)

● 字 ○ 字形
○ 異體字 ○ 部件 ○ 字根

排序(O)

○ 字碼 ● 序號
○ 字頻 ○ 字根頻 ○ 字根次

條件(D)

序號 100

字頻次(F)

| 字碼 | | 序號 | 字頻 | 字根頻 | 字根次 |
|---|---|---|---|---|---|
| 長 | 1 | 100 | 3583 | 5311 | 14 |
| 兩 | 1 | 101 | 3566 | 4941 | 13 |
| 便 | 1 | 102 | 3560 | 3592 | 4 |
| 因 | 1 | 103 | 3544 | 3928 | 14 |
| 八 | 1 | 104 | 3542 | 101435 | 876 |
| 動 | 1 | 105 | 3540 | 3550 | 3 |
| 知 | 1 | 106 | 3473 | 3754 | 5 |
| 水 | 1 | 107 | 3467 | 5708 | 48 |
| 頭 | 1 | 108 | 3440 | 3440 | 1 |
| 教 | 1 | 109 | 3432 | 3432 | 1 |
| 經 | 1 | 110 | 3369 | 3369 | 1 |
| 對 | 1 | 111 | 3301 | 3303 | 2 |
| 已 | 1 | 112 | 3250 | 3251 | 2 |

字總數（序號排序）= 8529

圖八、部件頻次的查詢

頻次表的搜尋功能只針對目前的排序項目。例如欲搜尋字形，必須依內碼作排序；序號、字頻、字根頻、字根次的搜尋則須依序號、字頻、字根頻、字根次分別作排序。數字的搜尋並不需要完全一致，例如在下表的林樹字集中，搜尋字根次 100，此時會找到字根次 104 的資料。

| 字形 | 字頻 | 字根頻 | 字根次 |
|---|---|---|---|
| | 0 | 4023 | 97 |
| 黿 | 0 | 2769 | 98 |
| | 0 | 20791 | 99 |
| 馬 | 1509 | 8060 | 104 |
| | 0 | 6470 | 104 |

頻次表的第二欄位爲字形碼，可用來表達字和字形間的關係。例如林樹字集中的「雞-1」、「雞-3」(「雞」的第三個字形，字形碼 2 保留給相對應的大陸簡化字)及「雞-4」(「雞」的第四個字形)，其中「雞-1」即爲「雞」，「雞-3」及「雞-4」必須由結構看出。它們在資料庫中的表達方式如下：

| 字形碼 | | 分解 | 部件 | 部件 |
|---|---|---|---|---|
| 雞 | 1 | | 奚 | 隹 |
| 雞 | 3 | | 奚 | 鳥 |
| 雞 | 4 | | 又 | 鳥 |

然而對於字形「牆」及「墻」，假定「牆」爲標準字，它們在資料庫中的表達方式如下：

| 字形碼 | | 分解 | 部件 | 部件 |
|---|---|---|---|---|
| 牆 | 1 | | 爿 | 嗇 |
| 牆 | 2 | = | 墻 | |
| 墻 | 0 | | 土 | 嗇 |

瀏覽頻次表或查詢結構時，只會列出「墻」，而不列出「牆-2」(小於2的字形附碼通常省略) 。字、字形、異體字、部件及字根於資料庫的表達中有下列的特徵：

1. 字　：字頻大於 0 而且字形附碼爲 1
2. 字形：字頻大於 0
3. 異體字 ： 字頻大於 0 而且字形碼不等於 1
4. 部件：字頻爲 0，其使用頻次顯示在字根頻欄位中，其結構可繼續分解
5.

## (3)Glyph Practice

### A. Organizing basic data sets

(omitted)

### B. Representation of glyph structure

Let us illustrate the representation of glyph structure by the following example. The characters in the right branch of 圖六 are expressed as following equations in computer.

1.灕＝　離
2.離＝离　隹
3.璃＝王　离
4.擒＝　禽
5.噙＝口　禽
6.离＝亀　内
7.禽＝　离
8.隹＝　雨
9.雨＝
10.亀＝　凶
11.凶＝

These equation are called「構字式」glyph structure equations , or simply structure equations. In the structure equation, symbols , , and represent Horizontal, Vertical, and Contain operations, respectively. We can apply the structure equations repetitively to obtain a structure equation which has only roots as in equation (3).

灕＝　（离　隹）
　＝　（（亀　内）　（　雨））
　＝　（（（　凶）　内）　（　（　）））
　＝　（（（　（　））　内）　（　（　）））..........(3)

The glyph「灕」in (3) has 8 roots. If we delete all operators in (3), then we have:

灕＝　　内　　……………………………………………………(4)

The right part of Equation (4) is called the Root String「字根序」of a glyph。The structure equation of a glyph has complete information of glyph structure and it is unique. Therefore it can be used as an identifier of a glyph. Root string or component string has less information than that of structure equation, but they also have quite good resolution for identifying glyphs. As an example, there are only 8 pairs in 8529 characters have the same root strings. So, in certain occasions, root string or component string can also be used as glyph identifier. （唄、員）。

In the 《中文電腦基本用字》character set, there are 9756 structure equations, among them 8529 belong to characters, 593 for variants, 629 for components。When simplified characters are added, then, we have 2284 equations for variants, 664 for components, and 11477 in total。

## C. Hanzi glyphholder

Wittern and App are the first ones to use SGML tags to annotate missing character.【註一】The technique they developed is called「漢字位標」Kanji Placeholder. It use '&' as open tag and '；' as close tag, and in-between has two parts, a character set identifier and a code word, respectively . They use this tag to locate missing glyph which has been collected in other code sets. For example,「&U4AB5;」means a missing glyph found in Unicode(U) set at location 4AB5.

The Placeholder is useful. It reduces missing glyphs. But, according to our previous discussion, it is not a cure to the missing character problem.

Following the same thought of using SGML, we extended their idea to define a new tag for missing glyph by applying structure equation or component string in-between the tag delimiters. It is called the 漢字形標 Kanji Glyphholder. As an example, in Buddhist text 阿門佛 can be tagged as 阿　門人人人　佛 or 阿　門　佛. When people using glyphholder, there is no need to look for the missing glyph in other code sets. All they have to do is to analyze the structure of the missing glyph　and then, write down the structure equation or string.

Glyphholder can also be used to identify variant by extend the structure equation, the component string, or the character code with a '.' then followed by a　glyph-sub-identifier. For example,「芍　藥・3　」represents「芍葯」, if 葯 is the third variant of 藥.

Glyphholder may contains placeholder and vise versa.

## ４．改良交換碼的建議

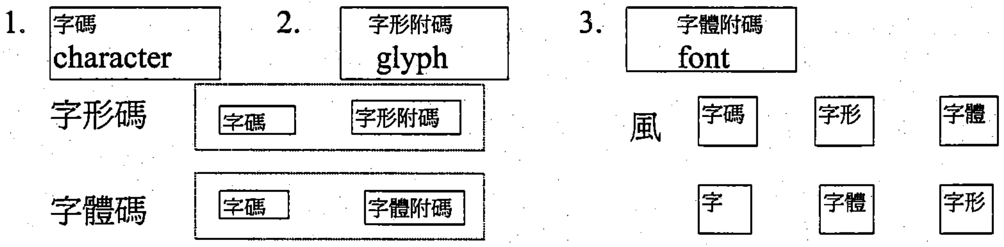
目前交換碼的根本病源，是錯把字形當作字。在此名實不符的情形下，名叫字碼（character code）而實際上卻以字形的差異來判別字。例如，五大碼中餞（C4C8）和飢（B047）分別佔兩個碼位。那麼它們是不是同一個字呢？依碼中的定義，不是；但在語用上實在是同一個字的兩個形罷了。這種情形在各交換碼中比比皆是，Unicode 和 ISO10646 也不例外。從文字學和語言學的角度看來，這種文字知識的表達根本是錯誤的，不健康的。從科學的角度來看，定義都錯了，往後推演可以勿論。像這種不健康的設計，導致目前應用上捉襟見肘，毛病百出，並不是意外的事。要改善交換碼，自應從此處下手。

此外，應知漢字集合從集合論的觀點來看是一個開放集合（open set），隨時可能增加字。它和封閉式的字母集合（close set）不同，字母是不會再增的。這些基本性質的不同，將導致其不同的性質和處理方式。可是，目前的做法卻是套用 ISO 646 處理字母的方法，來編漢字的碼。這真是削足適履。以上的說明和批評是希望大家了解，要改良現有的交換碼必須要爭脫既有的思考巢臼，否則不會有徹底解決的辦法。以下就是我們根據上述的原則，對改善現有交換碼所提出的方案。

### (1)三段式的編碼

首先，我們建議對字、字形、字體分三段編碼，如〔圖十〕：

基本元素：



圖十：字碼、字形碼、字體碼、風格碼的關係

若字形附碼與字體附碼都用時，則前後不拘，此時若只選字體而無字型參數，則稱風格碼（style code），若含有字形參數，則稱爲字樣碼（type face code）。字碼的使用與現在使用交換碼相用。字形碼、字體碼，風格碼或字樣碼等，則可以用前述之漢字形標載之與字碼一起用。字樣碼可能較長，因爲其中可能包括許多字型的參數，這無關本文主，詳情略而不談。字形附碼只是一個正整數的編號。在本系統中它可以用來區別異體字，也可以用來存放某字集中所用的字形。例如，我們把《中文電腦基本用字》表中的繁體放在字形附碼爲１的位置，把大陸標準簡體字放在２的位置上。這樣使用字形附碼的方式可以存放更多的文字信息。

### (2)各碼表達的忠實度

這種編碼方式的使用可由其傳輸失真的程度上分爲下列鈿種組合：

1. **只用字碼時，失真最大。此時不在乎所用的字形或字體，如台灣和臺灣都是一樣，只求明白語義就好。許多應用至此已可滿足。**
2. **用字形碼時，表示還要求正確的字形；用字體碼時，重在選定字體（不在乎字形）。這情形比上列的情形失真少。**
3. **用風格碼時，失真又比前者少。不止字形要對，字體也不能錯。**
4. **用字樣碼時，幾無失真，字的大小、粗細、……使用者與接收者完全一樣。**

以上四種情形可斟情使用以應付不同的需求。這是分成三段碼能提供的彈性。

### (3)構字式的運用

每個字形的構字式都不相同，所以它可用作字形的識別碼。這情形在討論漢字位標及形標時已述及。此節說明此觀念在設計交換碼的可能做法。

用構字式作字形碼最大的好處，是它是一個封閉系統：只要有一個封閉的字根集合，配以〔表四〕的規則，就行了。如此便可省去成千上萬的碼位。《交大字根系統》用 496 個字根就能產生 48713 字形的碼，並且還有不需要修改系統就能對付新字或缺字的彈性，就是最好的例子。其次，構字式是一種知識表達，它不僅較數字碼易讀易懂，所孕藏的構字知識更有利於應用程式的處理。然而，它的缺點是構字式長短不一，也嫌它太長，以致用起來較不經濟。

部件序或根序較構字式簡潔許多，用它們來替代構字式是很自然的想法。以《中文電腦基本用字》的 9122 個字形，加上 629 個部件，457 個字根，亦可產生 48713 個字形。此時 9122 個形只用一字碼，而其餘約 4 萬字，每個構字序卻可簡潔如 15 頁的那 11 個式子。換言之，可用漢字形標來表示剩下的四萬字形，每個字形之長度僅二或三個字碼。由於 9122 個字形的累積使用頻率已高於 99.9%，餘下千分之一以下的機會用較長的碼，對系統之效率影響極爲有限。所以善用構字式實是改善目前交換碼的一個好方法。

## 誌　　謝

感謝本實驗室的同莊德明、張翠玲和許婉蓉，沒有他們數年的努力，這篇文章不可能和大家見面的。本系統所有的程式開發是莊先生一人做的，勞苦功高。文字資料的整理許小姐出力頗多，張小姐除整理資料、稽核文獻外，還包辦了協調、檔案管理，以及本文繕打排印等工作。而本人，似乎只動了動口而已。