A Project for Dealing with the Missing Character Problem

C.C. Hsieh
Academia Sinica, Taiwan

Christian Wittern
Kyoto University, Japan

John Lehman
University of Alaska, Fairbanks, USA

Abstract

The project described in this paper takes advantage of the 7 year experience of the Academia Sinica Institute of Information Science Document Processing Laboratory in studying the missing character problem. It is intended to develop a packaging and processing system which is compatible with international standards for use in an internet or Windows environment. The initial design constraints were to use XML markup as part of a system to allow the viewing and processing of documents containing missing characters by any personal computer capable of running Windows 98, and using standard software such as Microsoft Word. It requires no modification of the user's system or software, and provides facilities for sharing data between users. The project was funded by a grant from the Republic of China (Taiwan) Ministry of Education, and is expected to be further developed to run on a larger variety of systems.

Background

It has long been recognized that conventional CJK character sets are not sufficient for representing texts in Classical Chinese. These problems arise from the fact that the set of characters in Chinese is open-ended. Glyphs (written representation of the underlying character) are missing from conventional CJK character sets because (inter alia):

1. The glyph represents a rare or obsolete character which the designers of the character set did not consider common enough to take up reserved space in the encoding table,
2. The glyph represents a character from a form of the language which is no longer used,
3. The glyph represents what is now a “foreign” writing of a character,
4. The glyph is a variant of a more common form,
5. The glyph is an erroneous writing of another form.

Unfortunately, classical Chinese texts by their very nature include rare or obsolete characters, characters from forms of the language which is no longer used, historical variants of more common modern forms, and erroneous writing or printing.

Wittern and App, in their classical study of Zen texts, 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 (外字)"…… It is clear from our work on electronic Chinese Buddhist texts that even Unicode (ISO 10646) will not significantly reduce this problem."[1]

The traditional approach, taken both by typesetters (e.g. the printers of the Taishō Tripitaka) and computer scientists (e.g. the creators of the Unicode standard) has been what might be termed "rectification:" the elimination of "incorrect" and variant glyphs, and their replacement by standardized glyph representing the underlying characters. In the case of those characters where no glyph was found in the typeset or defined character set, a glyph for a related character might be chosen.

While this approach has the advantage of simplifying and rationalizing the writing system, it has significant disadvantages as well. The two major difficulties are that it loses information, and distorts meaning. Information loss occurs because often the variant glyphs provide critical information on textual history, language background, and other context. Distortion of meaning arises from using glyphs which blur the distinctions between the original characters.

From a computer processing standpoint, a popular treatment is to create the missing glyphs in the user-defined-area in a code space. This treatment can display the missing character on the user's 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:

1. It is difficult to manage thousands of missing characters.
2. There not enough spaces for missing characters in any existing Interchange Code.
3. Most of the missing characters are variants, and there is no way to deal with the lexical problems of multiple glyphic representations of the same underlying character
4. The unsystematic creation of user-defined characters creates many information processing problems, such as matching, sorting, merging of characters.
5. Texts using user-defined characters are not sharable, and cannot be used with dictionaries and other computer-based text tools.

[1] 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

Project goals

The underlying assumptions for the project were:

1. Most users will not be computer-sophisticated
2. Most users will not be entering new missing characters
3. The system should allow reading with any software
4. The system should not require any changes to software
5. Users may already have some user-defined characters

The initial design constraints for this project were to create a system which would:

1. Work with existing missing character solutions,
2. Take advantage of existing character databases,
3. Use XML markup,
4. Allow the viewing and processing of documents containing missing characters by any personal computer capable of running Windows 98,
5. Work with standard software such as Microsoft Word,
6. Require no modification of the user's system or software,
7. Provide facilities for sharing data between users.

Existing missing character solutions

Other than placeholders or substitution, existing missing character solutions are divided into two classes:

1. Construction of new, larger character sets,
2. Standard representation of character structure (constructive approaches)

Construction of new, larger character sets may result from standards activities, or from descriptive efforts. For example, Unicode 3.0 with 27,484 characters and CCCII with 75,684 encodings (44,167 character orthographs plus 31,517 variant forms) were both developed as potential standards; either substantially reduces the missing character problem compared to Big

5 with its 13,053 defined characters[2]. The Mojikyo index (文字鏡) and the character database developed from the *Tripitaka Koreana* project represent descriptive efforts.

Standard representation of character structure (constructive approaches) have been developed by several researchers[3]. For the purpose of this paper, we will discuss the constructive approach developed by C.C. Hsieh at Academia Sinica[4], and the closely related version used for the digitization of the Taishō Tripitaka by the Chinese Buddhist Electronic Text Association (CBETA) – referred to in the diagrams below as 構字式-中 and 構字式-佛 respectively.

By way of simple introduction the constructive approach referred to above identifies a set of glyphic elements and a set of constructive operators (vertical concatenation, horizontal concatenation, one glyph inside another,...) by which these elements can be combined into any feasible glyph. Selection of a common set of glyphic elements (in this case based on extensive empirical investigation) and a normalized form of construction thus allows the unambiguous representation of a glyph based on its structure.

Solution

The proposed solution makes use of an XML-based self-extracting archive for document exchange. The above constraints are operationalized by the additional assumptions that:

1) Users will work in Big-5
2) missing characters will be placed in Big-5 user-defined character space
3) Users will see:
    a) A single self-extracting archive for each set of documents
    b) After extraction, a set of Big-5 encoded documents
    c) A New character input method
    d) A Repackaging program
    e) A Document-removal program on the desktop

The format of the proposed self-extracting archive is illustrated in figure 1.

---

[2] c.f. Lunde, Ken, *CJKV Information Processing*, O'Reilly & Associates, 1999.

[3] Hsieh, C.C., 中央研究院古籍全文資料庫解決缺字問題的方法, 第二次兩岸古籍整理研究學術研討會」, 北京大學, 北京, 1998年5月11-13日

[4] Hsieh, C.C., A Descriptive Method for Re-engineering Hanzi Information Interchange Codes, 漢字字碼與資料庫國際研討會, 京都□東京 1996年10月4日從缺字問題 , 談漢字交換碼的重新設計──第二部分

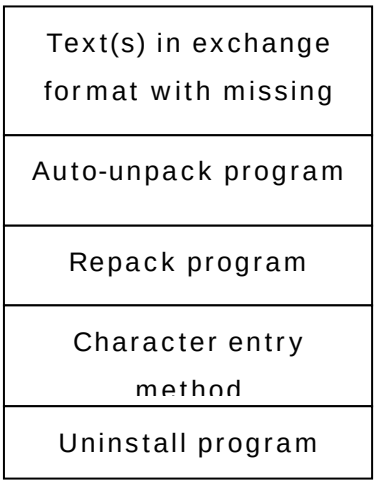

Figure 1 Format of the self-extracting archive

For the purpose of the current project, the content of the exchange format is assumed to be Big-5 encoded XML with the missing characters specified by a missing-character markup which identifies the character using the character construction approach developed at Academia Sinica.

Creating a packaged text

Figure 2 illustrates the packing process. As a byproduct of entering missing characters (displayed via the user-defined space in Big-5), a system table for this user (使用者造字表) has been constructed which documents the character construction code for each user-defined character. The system (either locally or via a server) has access to character construction databases, as well as to a table of correspondences between the character construction codes and different character standards (Unicode, CCCII, Big-5). Font information for displaying missing characters is taken from whatever sources is available (e.g. Unicode or CCCII. databases, Mojikyo,...). The result is an XML archive which includes a description of the missing characters, and the required information for display.

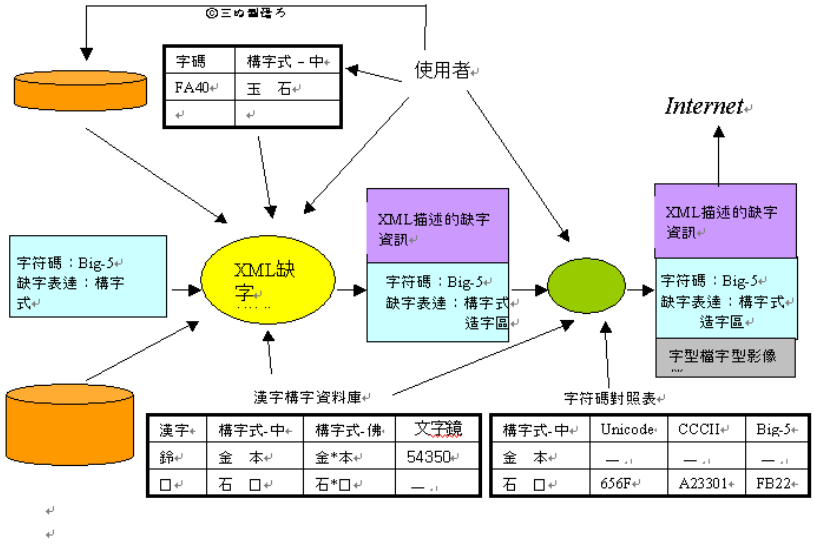

Figure 2 The packing process

Unpacking a text

The process of unpacking a text is illustrated in Figure 3. The XML text is unpacked into a Big-5 document; missing characters are added to the user's user-defined characters space, and the system table for this user (使用者造字表) which documents the character construction code for each user-defined character is either constructed or added to. Missing characters which are already installed are identified by the character construction code, and so are not duplicated.

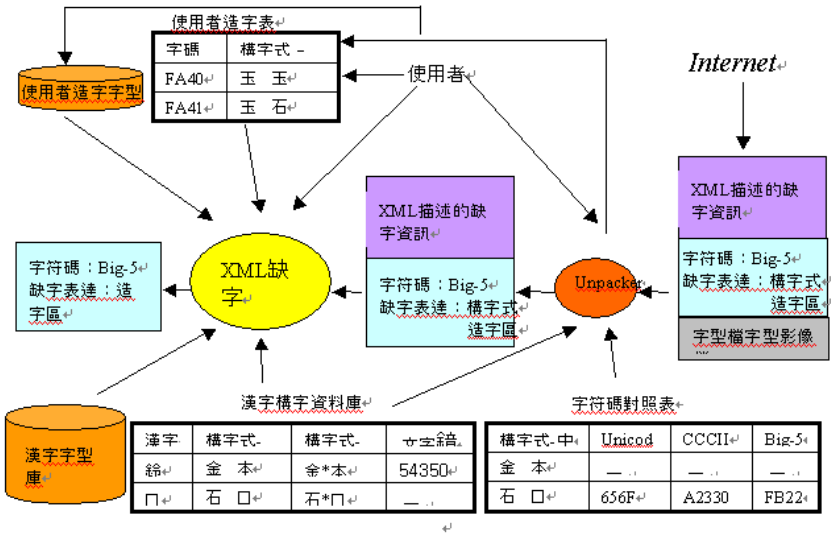

Figure 3 The process of unpacking a text

Result

While this process appears much less efficient than many of the solutions developed for existing text products, it has the great advantage that it does not require any change to the operating system or to the user's software, does not require any special software to display or manipulate texts, and it permits the simultaneous use of text products from any number of uncoordinated sources.