TECHNICAL PAPER

M-Based Asian Character Lookup Programs on the Internet: A New Way of Making M Visible

by Ed J.P.M. de Moel and Richard F. Walters

Note: *See related Discussion Session in Conference Program: M Applications on the Web, Thursday, 1:00P.M. - 2:15P.M.*

Introduction

Many people have wondered why it is that, with all its powerful features, M isn't better known in today's marketplace. There are historic reasons (dedicated systems, and more recently the dominance of the now-outdated relational model), but the facts don't jibe with its lack of visibility. We have no doubt lost opportunities to change that situation, but now we face a new opportunity, and it's time to take a look at the potential to increase our visibility in different ways.

In today's world, the most dramatic change to hit the information scene is the World Wide Web and a source of endless gigabytes of resources, multiple search engines, the best of which indexes only about a third of the material on the Web. Putting something on the web is likely to have unexpected consequences. One of us (RFW) wrote an article for a local throw-away journal of the Information Technology operations on his campus, which was put on the web as a service of that unit. Within days, a web surfer in England found the article and contacted the author for more details, visited the author in California and will probably develop a long-term collaboration as a result.

Stories of this kind abound. Where has M been in this process? Not exactly left out of the game, but certainly not using the vehicle to gain the kind of visibility that we might hope for. MTA members around the world are working to expand the visibility of M through various web-based resources, but to date the results have not been too dramatic.

In this article, we describe a different approach: we have adapted an application package originally written for single user PC systems which has appeal to a specialized audience for use via the Web. We offer this as a possible model for future efforts to increase the visibility of M through useful general-purpose, and free, Web-based applications.

Asian Language Dictionaries

Our application example involves computer-based techniques for looking up Asian language words. Chinese, Japanese, and to a lesser extent Korean all make use of a type of character that is sometimes called an ideogram (more properly a logogram). In Chinese (Mandarin at least), they are called Hanzi characters; the Japanese refer to them as Kanji, while Koreans use the term Hanja. They all derive from Chinese writing, having been imported into Japan over a thousand years ago and Korea not much later. We will use the term Kanji in this article, but the dictionary-related attributes apply to all three languages.

Most nouns in these languages are made up of more than one logogram, so that a Japanese dictionary indexes all words beginning with a given Kanji by the total number of strokes in the Kanji that follow the indexed character (some words have three or more Kanji). Very few dictionaries provide any index to list all compounds in which a given Kanji is not the first character (we know only of one English-Japanese dictionary that does so [Spahn and Hadamitzky, 1989]).

A Kanji character is made up of a series of strokes, whose position and order of writing are learned by children starting before kindergarten age. Unlike alphabetic languages, there is no clear-cut organization of the type of strokes that might serve as an unambiguous sorting order for Kanji. There are some clues as to the pronunciation (for a given dialect) in the case of Hanzi characters, but these clues also fail to provide an adequate basis for creating a dictionary.

Instead, Kanji characters are indexed according to a concept referred to as "radical and stroke count." A radical is a component of a Kanji character, and all

Kanji consist of at least one such radical. In all, there are 214 radicals used to collate characters in dictionaries for all three languages. Each character is assigned to a radical group, and then sorted by the number of additional pen strokes appearing in the character. Radicals are themselves grouped for convenience by the number of strokes required to draw the radical. The radicals often have a derivational meaning:

口 is a symbol for mouth,

人 is a symbol for man, and

日 is the symbol for sun, etc.

A dictionary lists all characters by a radical appearing in that character, beginning with one-stroke radicals and proceeding through radicals with up to 17 strokes. Children learn these radicals early in their schooling and develop some skill in recognizing the most likely radical in a given character when they want to look it up.

There are, however, problems with this approach. We will use as our example a character which is composed of several potential radicals: the Kanji character KUU,

空

or sora, used to represent the noun sky, akeru, the verb to make (or be) empty, or kara, the adjective empty. (Here we run into another complexity of Japanese: pronunciation of Kanji is usually multiple, with Chinese derived pronunciation such as KUU above and Japanese word equivalents like akeru and kara. The Chinese-derived pronunciation is most commonly used in compound character pronunciation.)

Japanese native speakers would instantly recognize that there are at least three candidates for the indexing radical in this character:

 (radical # 40) appearing at the top,

 (radical # 12) in the middle, and

 (radical # 48) at the bottom of the character.

The general rule in indexing is to use first a recognizable radical from the top or left-hand side of a character, and use others if this method fails. As a consequence, most native speakers of Japanese would pick the top three-stroke radical, 40) as the

most likely index for this character. (we have tested this on dozens of Japanese friends, and to date only one did make this choice). When told that this character is not indexed by that radical, they move next to the bottom component 48), and finally to

工

the middle, though by that time they have little confidence that this would be a likely selection.

In fact, the correct choice is radical #116, usually written as

穴

consisting of the five top strokes of the Kanji. When told this, native Japanese readily agree that yes, this could be the choice, but interestingly enough, very few of them think of it.

Non-native speakers of Asian languages would be far less likely to come up with the correct choice, since they are less familiar with the complete radical set used for all Kanji. Further, since familiarity with the 2,000 common Kanji requires a good deal of time and continued practice, few foreigners are able to remember all Kanji, so when reading texts, they tend to look for the character in a compound noun that they recognize, which may not be the first character in the word. Here again, dictionaries fail to provide an answer, and the reader is left with a frustrating gap in understanding the text that may last minutes, hours or days.

Enter the computer, and, for practical reasons, M. Unlike printed texts, computers can index by any or all components of a Kanji. It can even define components that do not appear in the common list of radicals. And it can index compound characters by all characters found in the compound. This is precisely what was done with the most commonly used Kanji characters, a set called "Joyo Kanji" learned by all Japanese students before they finish high school. One of us (RFW) work-

ing with a native Japanese speaker, an (American) instructor of Japanese, and a programmer, wrote a program in M that would allow users to look up Kanji based on any combination of radicals, conventional or new (selected by the native Japanese speaker and instructor), by its pronunciation(s), and by its English meaning(s). This work was done using DTM student version [Walters, et al., 1992], in a program the authors named Kanji- Lookup. It was an interesting, pre-GUI, DOS-based system that combined M with assembly routines that took care of the display of multiple fonts as well as the radicals and Kanji characters.

Rise and Fall of Kanji-Lookup

Kanji-Lookup enjoyed a fair amount of success in a circle of users that was small in number but large in geographic distribution, including language instructors and even native speakers from the U.S., Japan, and Australia. One computer scientist in Australia even made a far more extensive program, indexing several thousand additional characters and adding more compounds [Breen, 1997]. The latter version has been put on the web, with some features that differ significantly from the original.

Chinese-speaking students at UC Davis undertook to write a comparable program for the characters used in Mainland China, and a Hanzi-Lookup program emerged that was a modest start towards a program that would be useful for students of Chinese. (It was limited by not having many compounds, no inclusion of Cantonese pronunciation, and the absence of the larger character set used in Taiwan and Hong Kong.)

Users of Kanji-Lookup were pleased with its potential, and some instructors used it in their intermediate and advanced level courses. There was need for improvement, however, and the authors of that package sought to extend it by adding a few more Kanji and many more compound words. However, the work on this project was suspended when the M platform on which it was based was no longer available. Student DTM licenses expired in about 1994, making it necessary for users to reset their computer clocks in order to continue to use the package. With no further funding, and no immediate alternative in terms of a free or very inexpensive version of M, there was little incentive to continue to make additions to the system.

Despite these set-backs, work did continue on some additional indexing of a few more Kanji characters (enough to complete the so-called level 1 JIS X0208 Character set, approximately 3,000 characters in all), and work was begun on incorporating a larger set of compounds into the package. This work proceeded slowly, however, because of the uncertainties regarding a suitable platform and lack of funding to continue development.

Enter MSM

In search of a suitable platform for a longer-lived version of M, Dick Walters contacted Ed de Moel of Micronetics to see whether that company would be interested in letting students use MSM for this package. Other arrangements had already been made to use MSM in teaching M over the Internet (See *M Computing*, March, 1998), so the likelihood of being allowed to use MSM seemed high.

The results exceeded the original request in a number of important ways. Instead of simply sending a single user version to UC Davis, Ed de Moel was interested enough in the project to attempt a number of trial implementations using modern (GUI) technology. The implementation using the "web" ended up being a very convenient and efficient vehicle for the "Lookup" program. Using the web-based interface, end-users in Davis (California) could work directly with the prototype at Micronetics (in Maryland). Since the amount of information that needs to be transmitted is not very large, and web-browsers tend to cache their graphical images, the performance of the web-display pages was much better than originally expected.

To make the "Lookup" program work from the M side, not many modifications were needed. Since the work for the different languages was done independently by different students, there were minor differences in the lay-out of the global variables that were used to store the static information. The information in these global variables is now consolidated in a form that supports all languages, and one of the subscripts in the new global variable is the name of the language, so that the "Lookup" program can switch languages as easily as giving the variable that is used for that subscript a new value.

I.e., the current version of the "Lookup" program is available through the URLs:

```
http://www.micronetics.
com/ucdlookup/lkup.web?
EP=INIT&lang=Kanji
```

and

```
http://www.micronetics.
com/ucdlookup/lkup.web?
EP=INIT&lang=Hanzi
```

In order to make the Kanji characters visible, the DTM program used a series of bitmap files, and needed a special extension, written in assembly code to display the information from these bitmap files. For the web-implementation, a choice was made to store every Kanji character in its own little .GIF file. Since all browsers "understand" .GIF files, there no longer is a need for platform-specific extensions to make the Kanji characters visible. At first it was feared that the transmission of these .GIF files would be a major performance problem, especially when accessing the pages from "far away". As it turns out, these .GIF files are small enough (between 50 and 1000 bytes) and are cached well enough by the browsers, that the transmission time in this context plays a very minor role.

Another aspect of the web-based approach is the availability of hypertext links. Web users are familiar with the possibility to click on entities (in this case Kanji characters) in order to see more information about those entities. As a result, the current version can be used with hardly any need for instruction.

Results

Kanji-Lookup and Hanzi-Lookup are now both alive and well on the Web. The Introductory home page gives proper acknowledgment to Micronetics' version of MSM, and the program can be accessed by anyone, since it has now been placed outside the firewall that protects Micronetics' internal systems.

These two packages have been demonstrated to the Chinese and Japanese language faculty at UC Davis. The Hanzi package is linked to the home page of the UC Davis Chinese language group, and plans are under way for the Japanese version to be similarly included.

With the stimulus of these events, we have revitalized the extension work on the Kanji package. Two native Japanese speakers are now working on additions and improvements to the existing package.

Implications for M Visibility

There are multiple ways in which this success story can be applied to improving the visibility of M. First, it makes sense for M developers to create similar application packages that will run on the Web, and to place them so that they can be accessed by all.

Second, it is important to include some front page information on licensing M for those intrigued by the capabilities of such packages.

Third, we need to find ways to publicize these sorts of applications, using Web-based tools that will get them into the major indexing systems (Alta Vista, Yahoo, etc.).

Finally, we need to publicize (and give appropriate credit for) the availability of versions of M that can be used without requiring a separate M license for a static application. The Kanji package, for instance, does not need M programming support, since it is static, with development continuing only at the originating site. This is a profound step forward in encouraging M developers to create marketable products that demonstrate the power of M in forms that are competitively priced. M

References

Breen, J. *Personal Communication* (July, 1997). For further information on this package contact Professor Breen: jwb@dgs.monash.edu.au.

Spahn, M., and Hadamitzky, W. *Japanese Character Dictionary.* Boston: Cheng and Tsui, 1989

Walters, R.F., Fahy, D.W., Nakamura, A.Y. and Reid, C.M. "Kanji-Lookup: a Computer-Based, Multi-Indexed System for Beginning Students of Japanese," *Jour Computer-Based Instruction*19:1 (1992):27-32.

Ed de Moel is the past chair of the MDC and works with Micronetics Design Corporation. His experience includes developing software for research in medicine and physics.
His email address is: demoel@radix.net

Richard F. Walters, Ph.D., is a professor at the University of California, Davis.
