FROM THE EDITOR

Why Bother?

The Internationalization of M

by Richard F. Walters

Richard F. Walters

Someone once said that a hologram (a three-dimensional image made with coherent light in a lensless photographic process) is a solution in search of a problem. In fact, holograms are used mostly on credit cards to make unauthorized duplication more difficult—a far cry from their real potential.

Could the same be said about M's recent efforts to extend the language to support languages other than English? After all, computers have been based almost entirely on English during their nearly half-century existence. Programming languages are based on English commands, and if you want to program a computer, you'd better talk to it in English. Why change now?

There are a good many serious reasons why internationalization is worth considering at this time in the evolution of M. The first one is growth. As the 1992 Gartner study points out (see September 1992 *M Computing*, page 72, or call the MTA office for a summary of the report), the greatest expansion of M markets will be outside the United States in the next few years. Many new or minimally exploited markets are waiting out there for the software vendor with a product culturally acceptable to people whose first language is not English. If application packages can be readily adapted for use in these markets and, more important, easily maintained in versions that follow closely after the English version, then the vendors of those applications will have a significant marketing edge over most competitors.

The second convincing justification for internationalizing M is that, properly designed and implemented, it will once again outpace its high-level computing language competition by a decade or more. Whereas M has basic tools for internationalization, the other major languages are barely considering such a drastic change. Approaches to extending M may prove to be models for extending other languages. This being the case, there is an admittedly remote possibility that, a few years hence, computer-science faculty may even have heard of M and its innovative contributions to programming!

M as Stimulus in the Marketplace of Innovation

A third benefit of using M in non-English applications is that English-speaking users will derive significant new ideas, methods, and solutions from people who might otherwise lack access to user-friendly programming. Our willingness to "cross-fertilize," in the best horticultural sense, will broaden the "genetic pool" of computer innovations. Such innovation would be especially facilitated by effective machine-aided translation systems.

With its internationalization extensions, M is uniquely positioned to spearhead machine-aided translation. Indeed, it may prove to be the computer language of choice for the artificial-intelligence community.

Here are examples of the importance of cross-fertilization around the globe. The Japanese are masters at adapting foreign technologies, capitalizing on their adaptations, and marketing their new products to the very countries from which the original inventions sprang. (Remember when Japan first exported television sets to the United States?) The other side is that most of the world cannot learn about new Japanese ideas until they have been thoroughly assimilated into its technology. Perhaps if Japanese programmers were enabled (by M) to write code in a language easier to convert to English (or other languages), and if documentation could be translated more easily, there would be a real possibility for M to play an important role in improving U.S. competitiveness in the international market.

> *. . . a real possibility for M to play an important role in improving U.S. competitiveness in the international marketplace.*

These justifications to extend M to serve non-English applications point to even more opportunities. Consider the following project of a doctoral student working with me to solve a basic communication problem. In India, the array of no less than twenty major spoken dialects complicates delivery of public services. For example, a railway linking many regions of diverse dialects must post travel information that is understandable to a reasonable fraction of the passengers at any given railway station. (This makes U.S. attempts to publish bilingual voting information look like child's play.) The student developed an interactive train information system that, while retaining a single database of schedules, services, and prices, had external user interfaces adapted to the major dialects. With a single database, the information system could answer questions from most passengers. The techniques used by this student represent cutting-edge, artificial-intelligence knowledge in the machine-translation field today. The application is one that could have major implications in a country sorely in need of improved communications.

Consider next the internationalization of FileMan, first developed in English, but now available in German, Finnish, Chinese, and Japanese, to name only those adaptations with which I am personally familiar. In each case, the translation process has been slow and painful. As new versions are released in English, maintaining non-English versions has been tortuous. Since these efforts have not been coordinated, each translation is done independently and bilingually. If the Indian train station approach used in the example above had been undertaken to solve the internationalization of FileMan, the current status of these various projects might be significantly different.

It could be argued that since some programs already have been developed to serve non-English applications, why all the fuss? Didn't I just write that FileMan is running in several non-English environments? True, but these translations are essentially superficial, lacking many of the cultural adaptations necessary to truly support the cultural differences among cultures and languages. For instance, collation is a highly complex matter that has been partially solved by M in English because of the ASCII collating sequence. But this approach won't work in most other languages, because of diacritical characters, ellipses, and other complicating factors. Collation is culturally dependent. French Canadians do not collate their words in the same order as French people, and often collation within a single country may differ by usage (e.g., German telephone directory treatment of the "ue" character combination differs from the dictionary).

> *The subject presents a rapidly moving target, but if we have served to increase interest in the target, then perhaps more people will start shooting, and the number of hits will undoubtedly increase.*

This issue of *M Computing* introduces a few ways that the new language extensions of standard M language will pay off. The subject presents a rapidly moving target, but if we have served to increase interest in the target, then perhaps more people will start shooting, and the number of hits will undoubtedly increase. In turn, this increased activity will spur vendors and the MUMPS Development Committee to facilitate even more sophisticated capabilities in non-English language support.

We hope that you will find this issue to be an interesting potpourri of ideas, methods, and suggestions. Let's strive to continue developing and improving M so that it can truly maintain its lead in the important international marketplace. **M**

Richard F. Walters is a computer science professor at the University of California at Davis. Please address your comments to him in care of *M Computing* at the MTA.