M COMPUTING

Executive Editor
Richard F. Walters, Ph.D

Managing Editor
Martha Ogden

Editorial Assistant
Kevin Hanson

Composition
World Composition Services, Inc.

Editorial Board
Richard G. Davis, Ph.D
Gordon A. Gebert
Michael A. Gindlesperger

Review Board
Mark V. Bettyman
Robert S. Craig
A. Clayton Curtis, M.D.
Richard G. Davis, Ph.D
Ruth E. Dayhoff, M.D.
Michael A. Gindlesperger
William F. Harvey, Ph.D
Frederick L. Hicks, Ph.D
Arthur Lee
William J. Majerski
David F. Marion
Frederick D.S. Marshall
Erik Mumford
Thomas Munnecke
Helmuth F. Orthner, Ph.D
Roger Partridge
Gail Pentrod
Thomas C. Salander
Kate M. Schell
Hans J. von Klinckmann, Ph.D
Richard F. Walters, Ph.D

M Computing is published by MUMPS Users Group trading as M Technology Association, Suite 305, 1738 Elton Road, Silver Spring, Maryland 20903. Phone: 301-431-4070, Fax: 301-431-0017.

M Computing (ISSN: 1068-7580) is copyrighted by the M Technology Association. Material in this magazine may be reprinted by newsletters providing that credit for the original publication is made to *M Computing* (specify volume and issue numbers and date), a publication of the M Technology Association, Silver Spring, MD 20903 (Phone: 301-431-4070). Permission for reprinting articles from *M Computing* must be granted by the editor.

The information in this publication is believed to be accurate as of its publication date. MTA is not responsible for inadvertent errors. All trade and product names referenced are the service marks or trademarks of their respective companies. Opinions expressed are those of the individual authors.

FROM THE EDITOR

How Public Is M?

by Richard F. Walters

Richard F. Walters

As I write this editorial (a little late as usual), I am wrestling with some troublesome thoughts. Some of these concerns emerge from reading the several interesting and provocative articles on new concepts in M that appear in this issue—there is the "Alternative to MWAPI" by Luiz Carlos Lobo, "Evolving M in Object Technology" by Jerry Goodnough and John McManamon, and even the possible relation between them in Rodney Anderson's "How OO Relates to M." Please read each one and reflect on some of the questions they raise. How will the M community turn these new ideas into productivity tools that will help M compete with the outside world over the next few years?

My other concerns stem from personal research into distance learning and education. Some readers may recall that in the February 1994 *M Computing* I wrote of an introductory course on computer science that is offered electronically to remote sites from the classroom, which enables many students access who otherwise may not participate; hence the term distance learning. Since that article appeared, research in distance learning at the University of California, Davis, where I teach, has made important strides. We are now able to offer seamless, transparent connectivity for students (or clients) seeking help on a Macintosh, DOS Windows, or UNIX platform from a remote technical assistant (we call it the RTA project). The RTA in turn views a snapshot of the student's screen, annotates it, and returns the amplified copy in real time or through a delayed graphically oriented messaging system. Several classes at Davis use RTA, and it has the potential of significantly enhancing the education of the "Distance Learner of the Future."

What concerns me about the RTA model is that M currently has no place in this research. Why? Because the RTA project is a public-domain package available on these particular platforms for a range of potential users on Internet and accessing auxiliary materials stored in a variety of locations around the world. There is no M in this environment. Students and faculty working with me on this project are writing code in C or C++, HTML (HyperText Markup Language), and even some Visual Basic. The object code is available on potential users' systems as a part of the purchase price of each hardware system.

M has no such working arrangement; users of M are required to obtain licenses for each system on which the application runs, a condition that precludes its use in the RTA project. The recently announced InterSystems's no-cost, run-time license for low-priced applications running under DTM on DOS Windows machines is an important step in the right direction. (See G. William Moore and Jules Berman's

review, November 1994). But no such model exists for the other two platforms. Dan Baer's letter to the editor (February 1995) suggested that this model might be extended. This month's editorial expands on his thoughts.

As I try to juggle my responsibilities as executive editor of *M Computing* and continue my research into distance learning, I am bothered, profoundly, that I can no longer sustain my M programming preference within my research. I may even have to program in C myself, instead of relying on my students and associates for code generation!

Is there a solution? I hope so, and I think we have a model to use as a prototype. Adobe, Inc., maker of the world's best fonts and developer of the PostScript format for text production, has developed a product called Acrobat. It uses a proprietary converter program to translate almost any format of text and graphics from any system (yes, Mac, DOS Windows, or UNIX) to a new format, which flashes across the Internet in 7-bit ASCII that any system with an Adobe viewer can read. Awesome!

The key to this equation is that Adobe, in its wisdom, elected to make the viewer itself *public domain*. Hence, Adobe will make only a modest profit selling proprietary converters to schools, government agencies, and private corporations using this product, but innumerable end users will have converted documents on any platform with no additional licensing assessments.

We can and will use Adobe's Acrobat in Davis's RTA project for term papers, manuals, and other documents requiring multiple revisions by multiple authors. Why couldn't there be a comparable M model for these platforms? Then I could use M to develop my MWAPI, computer-assisted instruction, multilingual displays, or hierarchical database of system usage, and incorporate that code in the public-domain package I will release to users of the different environments on which RTA will run. Talk about a giant step forward for education!

Sure, there would be some inherent problems to work out. Indirection and the XECUTE command may need to be restricted or proscribed in the runtime code. Some implementor agreements may be needed to ensure an open market in the development of these different systems. It is likely there would be system-specific elements requiring special-case treatment. But the potential advantages are incalculable. In my case, it would mean that I can return to my beloved M for refining the RTA project, secure in the knowledge that my code will remain in the public domain. I also could write platform-independent code for most of my applications—something that we definitely are not doing in our current development.

In a broader context, M will be able to compete with products such as Visual Basic in a head-to-head battle where the deciding factor will be the productivity of the tool rather than the limitations of its availability. Surely that ought to count for something to any profit-minded implementer.

That's my problem. Or is it *our* problem? Can anyone help me? M

Richard Walters is a professor at the University of California, Davis, and the executive editor. Write to him care of MTA's managing editor.