Can Distance Learning be Used to Teach M Programming?

by Richard F. Walters, Ph.D.

Abstract

Having taught M Programming at the introductory and intermediate levels in several formats, I decided to attempt to teach a class over the Internet. This report describes two such offerings and presents some recommendations for future efforts in distance learning applied to M. The conclusions drawn from this experience are that M courses on the Internet are indeed feasible. Student motivation and preparation are major factors in the success of such a venture, but improving course materials is also important.

Introduction

M programmers are in demand in a great many settings. Few universities teach M, and the numbers of private companies offering instruction in M has, if anything, diminished in recent years. Furthermore, most commercial courses are intensive, week-long training classes that present problems in scheduling and logistics that preclude their reaching a large segment of people seeking this instruction.

There are alternative options. Correspondence courses have been around since the 1930's, and while they were no one's first choice, they met a need for asynchronous learning that was not achieved by other instructional methodologies. Technology opened new avenues of instruction in the intervening years, beginning with various forms of radio-based and televised instruction, and moving eventually to computer-supported forms of learning that took on a new life with the advent of the Internet. M has not yet jumped onto the Internet-based instructional bandwagon, but it would appear that the time is right to make such a move. This paper describes a course developed for delivery on the Internet. It represents a start towards M involvement in Internet-based training and offers some suggestions as to ways in which this process might be expanded.

Background

The University of California, Davis, has been involved in M for many years. The Veterinary Medicine Teaching Hospital has one of the best animal clinical record systems in the country, written entirely in M. M has been taught at several levels in the Department of Computer Science since 1983.

Beginning in 1990, a new introductory course on computers was offered, in which students were exposed to M programming. The course has been offered nearly every quarter since then (although some instructors use other programming languages instead of M in their versions of the course), and several hundred students have had exposure to the language as a result.

I had been preparing to offer this course in independent study mode for several years, videotaping lectures and preparing material that could be studied without attending lectures or scheduled laboratories. In fall, 1995, the first independent study course was offered in parallel with a conventional lecture course (Walters, et al. 1997). The courses were identical in content, resources, deadlines and examinations. Matching profiles of the independent students with a comparable group in the lecture section showed that the independent study students did just as well (slightly better) than students in the lecture course.

Since 1990 I have also taught M at intermediate levels for computer science majors taking database systems classes. The first term (10-week academic quarter) has an optional exercise, and the second has an exercise requiring students to perform complex reformatting of a database to invert the information. Computer science majors find the language easy to learn and are impressed with its power compared to C.

In addition, I have taught one-day intermediate M programming tutorials at MTA conferences for several years.

Administrative Details

With this background, I decided to experiment with a new Internet-based course in M, covering both beginning and intermediate level material. I approached our University of California Extension (UNEX) to ask if they would be willing to help me offer the course. UNEX was quite willing (they are anxious to get more UC faculty teaching their courses), and we decided to run an experimental offering starting early in 1997. Accordingly, an experimental course in M programming was offered to a maximum of twenty students beginning January, 1997. The course was sufficiently successful that a second, more formal offering was scheduled for fall, 1997. This time, the class was open to a maximum of 50 students. The course materials were then further revised and a third offering of the course is just beginning as this paper is being written (January, 1998).

Course Design

Independent study courses work best if there are specific activities planned for the participants, with opportunities for frequent feedback an essential component in the learning process. Courses on the Internet should adhere to these design constraints. Our campus-based independent study course was built around laboratory exercises which the students complete on their own, relying on office hours and electronic mail to get the necessary feedback. Although this course only presents introductory level material in M, it served as a good model on which to build some exercises for the intermediate level skills that were needed to complete the Internet course. Projects from MTA one-day tutorials were adapted, and exercises from chapters in *M Programming* were added to provide exposure to more advanced concepts.

The beginning-level exercises taken from the campus course had been used and refined over several offerings of the class and worked quite well in that setting. They were designed around a pre-test, which was to be completed before the student started the lab exercises, a post-test to be completed in steps as the student worked through the laboratory exercises and readings, both from lecture notes and the textbook: *The ABCs of MUMPS*, which was the predecessor to *M Computing*. This model was retained, hoping that Internet participants would send in their pre-tests, get feedback, then continue with the lab, turn in the post-test, and, if accepted, proceed to the next laboratory exercise. If remedial work was required, that would be done before the participant moved on.

In all, eight laboratory exercises were included, the last one being a project involving writing M code to solve a problem of the participant's choosing, subject to approval by the instructor. With the completion of this exercise, participants were in a position to begin designing their own projects without further assistance.

Interaction was provided in both cases by email and fax, with the usual turn-around time less than 24 hours, sometimes within a few minutes of transmission of a participant's file or question. A new interactive package called the Remote Technical Assistance (or RTA) was tested in the fall course offering, but there were technical difficulties that prevented its practical use for that group.

Course Delivery: Winter, 1997

The introductory offering of *M Programming on the Internet* began in January, 1997, with some participants signing on as late as mid-February, the cutoff date for that group. Participants were forewarned that the course was experimental. The course fee was $250. Those who enrolled received the laboratory manual, a set of lecture notes, a copy of *The ABCs of MUMPS*, and a floppy disk with UCD MicroMUMPS. They were informed that UNEX (University Extension) would send them a certificate of completion if the eight laboratory exercises were finished within a 15-week period (approximately the beginning of June, to allow for late starters), for material that was estimated to require approximately ten weeks of effort at a commitment of about ten hours per week.

A total of 15 students eventually signed up for the course, including one from Sydney, Australia. Communication was almost entirely via email, with turnaround of less than a day in most cases.

Of the 15 participants, one withdrew because of a job change early in the course. The completion record for the others was as follows:

7	Completed entire course
4	Completed most of the course (5 of 8 exercises)
3	Completed at least 3 exercises

The reason given by all for not completing the material was the press of other activities. Evaluation of the course content was favorable, with few suggestions for specific revisions. One student wrote: "I think this method of teaching is ideal for those who cannot get away from the homefront." This sentiment was echoed by a number of others taking the course, including some who were not able to complete it.

Second Offering

The relative success of the first offering of this course persuaded us to try a full-fledged course in fall, 1997. Our goals were to test the model of an Internet-based course for UNEX at Davis (no others had previously been offered by that unit) to see how effective such a course might be in learning achieved by the participants and to see whether this model might serve as the basis for expanding the offerings of M instruction using the Internet. We also wanted to develop accurate measures of the time required for delivery of the course.

Given these goals, we decided to offer the course to a maximum of 50 students. The course was changed in the following ways:

* The laboratory exercises and lecture notes were revised, completing the tentative assignments used in the first offering. Pre-tests for individual laboratories were eliminated as impractical. One new post-test question was added to each exercise: participants were asked to estimate how many hours they had spent on each laboratory exercise.

* The new text: *M Programming: A Comprehensive Guide*, which appeared in June, 1997 was used instead of the earlier text.

* A course pre-test designed to provide background and information on entry-level skills was included with the handouts, as was a basic course introductory syllabus.

* The disk included, in addition to the M material described earlier, a copy of the RTA client for PC systems, and participants also received instructions on how to log on to RTA and use it to send screen snapshots, dialogues, and attached files.

* A new computer account was set up to manage course submissions by email. This account made it possible to separate messages relating to this course from others received by the instructional staff, and it also provided a convenient archive for later analysis.

* The course fee was raised to $450, to better reflect the content and value of the course in comparison with other Internet offerings.

Participants were once again given 15 weeks to complete the course, with enrollment starting in September, 1997 and extending, with a few late entries, to approximately October 31. Completion date was set for January 15, representing slightly more than 15 weeks from a starting date of October 1.

Actual Course Delivery

A total of 47 participants completed enrollment and were sent the course packet. There were two participants from abroad, one from Sri Lanka, the second from England. Based on information provided in the pre-test, the students ranged from full-time programmers working mainly in other languages to individuals who had never programmed in any language.

Communication via the RTA packet proved unworkable, owing in part to technical problems connecting participants who used America Online as their Internet service provider. This problem has not yet been resolved. The need for RTA-based interaction was felt by many, but it did not prove practical in this offering.

As a result, the majority of messages sent to the instructor were either email or fax, with return response usually the same day. This proved satisfactory in most respects, although some had difficulty capturing screen output and sending code conveniently.

Student Performance

Participants varied widely in their dedication to the course. Their completion record is as follows:

7	completed course
3	completed lab 4 or more
6	completed through lab 3
3	completed through lab 2
9	completed through lab 1
3	sent in pre-test only
6	did not respond after receiving material
3	withdrew
7	transferred to the winter, 1998 course

Students completing the course were sent a formal notification by UNEX that they had been assigned a PASS grade for the course. Others were notified that they had received an INCOMPLETE grade which could be turned into a PASS if the work was completed within twelve months of the date that grades were turned in.

The length of time required to complete different assignments also varied considerably. For the first four laboratory exercises, the times reported by students are summarized as follows:

Lab #	Average hrs	Min	Max
1	4.1	2	10
2	11.3	4	40
3	10.9	3	40
4	19.6	5	60

These figures reflect only those completing the exercises, with a smaller sample size in the last exercises. The distribution was decidedly bimodal, with one or two students requiring a great deal more time than the somewhat better prepared group.

As the instructor responding to participant assignments and questions, I found that I spent an average of an hour and a half a day answering email, commenting on assignments and handling administrative details of the course.

Evaluation

The response to the course by those who completed it was overwhelmingly positive. Participants felt it met their expectations, prepared them to work with M code effectively and to communicate with others about M code on an equal footing with experienced programmers. The text also was well received, although some minor typographical errors were encountered.

Although the laboratory exercises were considered helpful, several participants noted that they needed greater consistency in stating exactly what was to be submitted. There were also a few typographical errors, and some problems with downloading the M code used for later labs in the class. These problems created more difficulties for participants with minimal background; those with some programming experience did not appear to have noticed these minor inconsistencies at all.

Discussion

It came as something of a surprise to discover that, out of over 40 individuals enrolled in the course, less than a quarter finished the course in the projected time. As the deadline was approaching, I sent email to all who had not yet completed the course, asking whether they wished to continue even if they did not receive a formal passing grade at the end of the course. The response was in almost all cases a request to be allowed to continue with the course. Reasons given for not having completed the material usually related to work or home demands on their time that made it difficult or impossible to keep up. The most common reason given by those withdrawing from the course was change of job responsibilities that no longer made learning M necessary. It was encouraging, however, in the light of the poor completion rate, to learn that many wished to continue on beyond the initial time period.

The unusually long time required to complete some of the exercises came as a surprise. Students at Davis working independently tended to spend about twice as long completing the difficult laboratory exercises as compared to those who had help during scheduled laboratories, but the participants' time spent on these exercises seemed longer than students had reported at Davis. Further analysis indicated that persons experiencing the greatest trouble usually wasted a good deal of time trying to answer questions on their own before seeking help. Only in one or two cases did those who sought help early continue to have major problems with an assignment. Novice programmers also spent quite a bit of time reading the text and other materials as compared to those for whom some of the material was already familiar.

It is, however, reassuring to realize that in no case did a participant simply give up because they were not able to master the material. The progress made by some beginners was gratifying; some caught on and were able to move much faster on later exercises that were in some respects more challenging.

On the administrative side, this course presented some new challenges for UNEX. Independent study courses should be available on an as-needed basis, rather than having to adhere to academic schedules. UNEX is not yet in a position to provide this flexibility, but they now realize the importance of addressing this problem.

Scalability

In these course offerings, I was the only instructor available for consultation on the class. Some participants had help at the office (several were in the same office as other participants in the course), but there were no other instructors. We had hoped to have two additional assistants for the course, but technical problems of obtaining a high speed link to the campus computer proved insurmountable in the time frame involved.

The course is, however, scalable, given the right situation. With the RTA package working for all participants, and with high-speed links to the course server, it is theoretically possible for the course to be offered using a number of volunteer or paid instructors. We will explore these options in the weeks ahead.

Conclusions

From these experiences, we can draw several conclusions:

* M can be taught over the Internet

Persons interested in learning how to program in M can do so by using a course of this type. There is a great advantage in eliminating a scheduling problem requiring instructor and participant to be in the same physical facility at the same time. Many persons wishing to learn M simply cannot find time to travel or rearrange their schedules for intensive courses even if taught locally for the participant.

* Effective course content design is essential

Even though the materials used in this class had been used for independent study in the past, the differences between independent study and distance learning (in which office hours and other personal contacts were not available) required important revisions. Even small inconsistencies can cause problems in a distance learning environment.

* Not everyone will complete an independent study course

The pressures of other duties and time factors will inevitably prevent some from completing a course for which there are no fixed deadlines. It requires a good deal of self-discipline to stick to such a program; not everyone is able to maintain disciplined progress when confronted with other more immediate options. Some people also require a great deal more assistance than others; instructors should be prepared to provide a great deal of moral and technical support in some cases, whereas other participants may require little or no assistance beyond printed course material.

* M Programming on the Internet should be scalable

The skills required to design effective course materials for an Internet course are not those required to respond to questions or evaluate assignments submitted by course participants. It therefore stands to reason that a course of the type described above could be "taught" by a number of people familiar with M and experienced in providing one-on-one assistance in working out coding problems. This option requires further study and will be addressed in the current offering of M.

Acknowledgments

I am grateful to those who participated in this venture at several levels. Their patience with little problems was gratifying. Their willingness to point out even small problems with handouts has helped improve the materials for subsequent offerings. I also want to express my thanks to the UNEX staff, especially Nancy Bruss, for their enthusiasm and cooperation in bending the normal procedures to enable us to offer this course.

References

Walters, R.F., and Reed, N.E. (1997), "Outcome Analysis of Distance Learning: A Comparison Between Conventional and Independent Instruction," Available online:http://sunsite.unc.edu/horizon/CD/NSCI-HTML/Walters/html.

Richard F. Walters, Ph.D., is a professor at the University of California, Davis.
Email: walters@cs.ucdavis.edu
