Litigation-Support Software for Platform Independence

by Linda Thorpe and Bruce Douglass

In our success as a vendor of litigation-support software for a variety of platforms, M's ANSI (American National Standards Institute) standard and transportability have been essential. We have found, however, that there are some unavoidable platform and implementation differences. Therefore, complex applications in M may not be immediately transportable from one platform to another or even from one implementation to another. This article describes our small company's entrée into the legal-software market and why the nature of our software product and clientele has made the transportability issue so critical. Further, we will explain how we have achieved our goal of a transportable, single-version source for our litigation-support software.

History—The Choice of M

Our company, D-M Information Systems, was first incorporated as Delta Computing Systems in Nebraska in 1979, mainly to offer a custom medical-billing package for a pathology group. Simultaneously, however, Delta was involved in a second area, offering litigation-support services to law firms in Lincoln, Nebraska, and San Francisco, California.

Computerized litigation support helps litigators manage the large volumes of diverse documents gathered during the discovery phase of litigation. It also may provide full-text retrieval for transcripts of depositions, hearings, and trials. We first developed a noninteractive version of the litigation-support software in 1977 in ALGOL, on a Burroughs B-6700. Our clients input using IBM typewriters with an OCR font. We rented an early Optical Character Scanner at the University of California at Davis to batch-load the database. In 1980, we rewrote the system for a Burroughs B-800 in MPL, an ALGOL-like language. We chose the B-800 and MPL because in the then-remote town of North Platte, Nebraska, the only hardware support we could get was from Burroughs.

The medical-billing package was the first product we developed after Delta's incorporation, and it was written in that same MPL language. In 1981, the new owner of our client, a pathology group, requested that we convert the billing package from MPL to MUMPS.

The Litigation-Support Software Conversion

As we completed the conversion of the medical-billing package, we came to the happy realization that we could accomplish rather easily with MUMPS everything we wanted for the litigation-support package and more. The underlying global design we developed in the early 1980s has remained as the core element of what we believe to be the most flexible and efficient litigation-support database management system on the market. The document records for litigation-support applications are highly variable with respect to numbers of fields and numbers of entries per field. We use the data-addressability of M and cross-reference all entry immediately by user-specified fields. The original global design with the flexibility allowed by M has never let us down.

The first MUMPS implementation of the litigation-support software, the Case Information System (CIS-FT), happened in 1982 on a PDP11-23 using InterSystems M/11-V3. At that time, we worked as a service bureau for litigators, entering data and producing specifically requested reports. By 1984, our law-firm clients were requesting the CIS-FT software for use in-house on their personal computers.

The medical billing and other custom and service jobs we had been doing thus far gave way to the development of a polished software package for a vertical market composed of litigators in large law firms and corporate and governmental legal departments.

Early Software Installations

We first chose Micronetics MUMPS for the new PC market and placed a few single and multiuser systems in law firms between 1985 and 1987. We had now successfully transported our product to the PC platform where law-firm clients were willing to dedicate a PC with some terminals to litigation support.

Nevertheless, except for a few clients who had used D-M's services and specifically wanted D-M's software, we were finding the legal market difficult to penetrate. We had enough of a toehold in the legal market to be excited by the prospects but we were discouraged.

Our clients were attracted to the idea that the same powerful, multiuser litigation-support software running on the VAX could be available on a portable PC in depositions . . .

MUMPS was part of the marketing problem but at the same time part of the solution. We found that either our prospective clients and their new MIS departments had never heard of MUMPS or they had negative, unfounded misconceptions of it as an awkward "medical database" language. In response, we explained to our prospects that MUMPS was highly transportable because of its ANSI standard and was in rather excellent and exclusive company with the reputable languages, FORTRAN, COBOL and PL1. We began to make progress.

Transportability as a Major Issue

Transportability was and remains an issue for software developers. In the legal market, we have found that we must accept hardware choices made by our clients without our input. Furthermore, it may be an occupational hazard with attorneys that they choose and then defend their choices of hardware despite any evidence of its unsuitability for certain applications. In any case, for most law firms word-processing and accounting software are the high-priority applications; if hardware choices are based on software choices, then word processing and accounting will be accommodated first. Litigation support, on the other hand, serves only the litigators who may represent only a small percentage of attorneys in the firm. Nevertheless, our software's functionality was unusual in the late 1980s. Two law firms with large litigation departments and VAX/VMS systems chose our CIS-FT software specifically because of its transportability to their systems.

Principle of Single Source

We developed as a principle the idea of maintaining the same source code for all platforms and advertising this fact as a selling point. Our clients were attracted to the idea that the same powerful, multiuser litigation-support software running on the VAX could be available on a portable PC in depositions or in the courtroom. To accommodate the single-source principle, we implemented a simple branching scheme to either execute or skip over certain blocks of code depending on the current platform. Despite our principle, we needed a better system, as much for our own efficiency as for the satisfaction of our clients. The CIS-FT system was growing in scope and complexity and by 1986 encompassed over 1,500 separate routines. With four or five programmers working at least part-time on litigation-support software and new versions being released at a rate of three per year, source-code management became an issue. The problem was compounded by multiple MUMPS implementations and by implementors not practicing "single-tiered support," i.e., new versions of MUMPS were released for some platforms months before they were released for others. Given the disparity, the various versions might as well have been additional implementations.

In the late 1980s, we chose to implement our application on DataTree MUMPS as well as on InterSystems and Micronetics to accommodate the demand for our product on PCLANs. With close to 1,700 programs, it continued to be important to maintain only one version of the source code for our growing system and growing client base. The simple branching scheme was becoming unwieldy, especially since the introduction of compilation as opposed to interpretation meant we were seeing long lists of compilation errors for the code which would be skipped at execution time.

Method for Maintaining a Single Source

We adopted a completely new approach to the single source-code principle, an approach which continues to this day to serve us well. The essence is automatic file-time preprocessing used to modify the MUMPS code according to the destination platform. The preprocessing uses two techniques—conditional compilation and macro-generators.

The conditional compilation is a filter and may be illustrated by the following example. Consider opening a file in a DOS environment as opposed to a UNIX or VMS environment. The preprocessing is signaled as follows.

Continued on page 9

```
;—— Open new output stream disk file ——
S FILENAME="abcd.wp"
S $ZT="OpenErr"
%%COMPILE ON=DTM
S DEV=10
O DEV:(mode="w":file=FILENAME):1 E  G NoOpen
%%COMPILE ON=M/UX,M/VX
S DEV=FILENAME
O DEV:("NWS"):1 E  G NoOpen
%%COMPILE ON
S $ZT=""
;—— Write file ——
U DEV
:
:
;—— Close file ——
U 0
C DEV
:
```

Figure 1. Opening a file in a DOS environment.

As the above example illustrates, the details of the OPEN statement syntax and behavior are left to the implementor. Without some invention of the type above, we as vendors of software for multiple platforms would need to maintain separate versions of code for separate platforms.

A second technique, our "platform-sensitive macro-generator," is designed for a small number of situations that are dealt with in a nonstandard way by implementors, for example, break detection, or echo on/off.

The macro name embedded in the M code will expand to the correct string of code depending on the platform on which it's being compiled.

```
;—— Source Code ——
%%FLUSH
I %%BREAK G INTR
;—— Generates on DataTree ——
I $I:TA=0 R *%:0 U $I:TA=1 K %
I %BREAK G INTR
;—— Generates on M/UX ——
W *-1
I ($ZJ\8#2) G INTR
;—— Generates on M11 V5 ——
W *-1
I ($V(2,$J)\8#2) G INTR
:
:  etc.
;-----------------------------------------
; Note: for DTM (DataTree) the ZTRAP routine
;       is always set to be %DMBREAK which
;       sets variable %BREAK.
```

Figure 2. The code contains the macro.

At this point we have developed macros for InterSystems M/11-V5, M/11+, M/VX and M/UX, for Micronetics MSM-PC, for PLUS-5, DataTree pre4.3 DTM and post 4.3 DTM. The code within the conditional compiles sometimes encompasses more than one implementation.

The macro method should not be used as a way to change the flavor of the M language. We use only six macros for the transportability.

It may be important to mention that the use of both the conditional compile and the macro-generator were facilitated for us by an approach to source-code management we adopted early in the 1980s.

This approach was not original with our work. (It was introduced to us by Dr. Delane Wycoff, who used it at the University of Iowa Blood Bank, and we believe it came to them from Jim Peebles when he was at the University of Arizona.) With this approach, source code is maintained in globals using an editor which also handles the ZSAVE operation. Since ZSAVE delivers this source code to the compiler, the job of the editor in processing the conditional compiles and macros is to deliver only the platform-specific code to ZSAVE at compile time. The source-code editor, by the way, is written in M and is not a difficult program to develop. As a matter of fact, the writing of such an editor had been assigned to a group of undergraduate computing science students at the University of California at Davis by Bruce Douglass when he was a guest lecturer. Several of the students have written editors that are so complete and useful that the students continued to use them for maintaining source in further M course assignments and beyond.

Importance of Transportability in the Future

Today, the litigation-support software market has grown crowded and competitive. Of fourteen litigation-support packages reviewed in the August/September 1993 issue of "Law Office Computing," though, only one other package besides CIS-FT listed UNIX as a platform and no others listed VMS. Four listed Windows in addition to DOS. We know of no other complete litigation-support package written in M.

New hardware and operating system innovations are demanding rapid development from software vendors, including litigation-support software vendors. The front-ends must be attractive and user friendly. Databases are expected to interact with or include imaging systems. Clients want Windows compatibility and more.

Our CIS-FT software is installed in more than thirty law firms and corporate legal departments, many of which have multiple licenses. When they receive updates for their multiuser VMS or UNIX systems, they expect to receive identical updates at the same time for the laptops that individual attorneys carry to depositions or to court. A new release requires testing on multiple platforms but when it's ready, the compiled version of the same code goes out the door to all our sites regardless of platform.

Conclusion

M proved early to be a powerful choice for litigation-support software. D-M has discovered that in the litigation-support software market, transportability is critical. Because of M's ANSI standard, this transportability is basically not a large problem. Truly transportable code—that is, one single version of source code for all platforms—must accommodate some differences among M implementations. A scheme involving what we call "conditional compilation" and "platform-specific macros," handled by a source-code editor, will maintain a truly transportable single source code.

Linda Thorpe is president of D-M Information Systems, Inc., which provides computerized litigation-support services and software.

Bruce Douglass is vice president of systems development at D-M.

Drop us a Note
Tell us which artical you think is the best in this issue

REPRINTS ARE AVAILABLE!!
Please call the MTA office at 301-431-4070 for details.