The IBM Effect

By George Schussel

You can't hide—Big Blue's making decisions about your computing work. In this special report, a renowned IBM-watcher discusses IBM trends you can follow to success in the 1990s...

You use software from Lotus, Microsoft, and Borland. It runs on Compaq, Dell, and NEC PC compatible computers. Novell's NetWare ties together your PCs. Large computer support is provided from two places: corporate headquarters where there are clustered VAXes and the division with HP Precision Architecture minis. You've looked at articles on IBM PS/2s and—for now anyway—a commitment to Micro Channel Architecture (MCA) is too closed for you. After all, do you really need to pay any attention to what IBM is doing?

It's my contention that what IBM does, especially in software, has an effect on the overall data processing industry that ranges far beyond IBM's customer base. In gross measures, IBM's annual sales of about $60 billion means that it holds about a third of the total worldwide market for hardware, software, and related services. Yet the standards IBM sets in its one-third has and will continue to have a powerful impact on the remaining share of the industry. My goal in this article is to show you how "IBM watching" can help you plan your data processing future.

IBM and software

With 1989 software and services sales over $20 billion, IBM is by far the largest software company in the world. Its dominance in software is actually greater than in hardware. The largest independent software vendors such as Computer Associates and Oracle have sales of several $100 million to $1 billion per annum. So, while the IBM hardware company is five

The IBM Effect *Continued*

times larger than its largest competitor, DEC, the IBM software company is 10 times larger than its largest software competitors, Oracle and Computer Associates.

IBM pays a lot of attention to software because: 1) software drives the sale of hardware, and 2) software has a higher growth rate than hardware. IBM's success in the software arena has a powerful impact on all users of PCs—whether or not the brand name on the console is IBM.

Over the years many internal IBM software developments have become important industry standards. Perhaps the first major industry software standard from the IBM labs was FORTRAN in the 1950s. When my daughter, Stacey, took an engineering computer class at MIT this past semester, I was only slightly surprised to learn that FORTRAN was the principal language used in the class. In other words, an IBM software development of the 1950s is still widely used and very influential 40 years later!

Every PC user knows that IBM can set hardware standards. The products of early (non-compatible) PC vendors like Eagle, MITS and Victor quickly fell out of favor when the IBM standard PC (and clones) became established. It should be noted, however, that what causes the PC to be an industry standard is the general availability of its operating system, MS-DOS, which has sold approximately 40 million copies. Microsoft was a small software vendor in 1981 with a CP/M clone operating system. It wouldn't be the billion dollar giant it is today without IBM's endorsements.

MS-DOS is Microsoft's operating system for PCs. But more importantly it became IBM's operating system, and therefore the standard for the entire PC compatibles industry. Even though sales of IBM PS/2s are about 20 percent of the American market for PC compatibles, MS-DOS is delivered with 80 percent of all PCs. So MS-DOS provides an example of a product that is modestly successful for IBM, but wildly successful in setting an industry standard.

IBM and database standards

For both PC and mainframe users, one of the most important emerging standards is the SQL database language. Although a number of different programming languages for implementing the relational model were developed in the 1970s and early 1980s, the IBM implementation, SQL, became the standard. This happened even though most relational database management system (DBMS) experts severely criticized

SQL's technical shortcomings in presentations and papers through much of the 1980s. SQL was adopted—as you'll see—for two reasons: political expediency and a recognition that any other DBMS language wouldn't be supported by IBM—and thus couldn't become a widely used standard.

In the 1970s, before the advent of commercial relational systems, other widely used DBMS products were available. CODASYL (Conference on Data Systems Languages) DBMSs were probably the best known. CODASYL DBMS products were sold by Cullinet, Sperry, DEC, Data General, and others, but never achieved the success that SQL is achieving now. Even though the CODASYL specifications were suggested as an industry standard, IBM refused to produce a CODASYL system and opposed its adoption as an standard. Why? Because an existing IBM product, IMS, was competitive and incompatible with the CODASYL approach. One can conclude that when IBM refuses to implement a product, by definition, it can't become an important standard.

IBM has developed several DBMSs—all based on SQL. For large mainframe MVS sites there's DB2. For small mainframe sites running the VM and VSE operating systems, SQL/DS is offered. On PCs, the Data Manager component within OS/2 Extended Edition provides SQL capabilities closely compatible to DB2. And on the AS400, SQL/400 is the SQL standard. Under development in IBM laboratories at this time are SQL database engines for the UNIX world for both mainframes and workstations.

SQL, initially a mainframe language, is going to have a powerful impact on software developed for PCs. As PC database applications migrate toward multiuser access across LANs, I think the SQL standard will largely replace proprietary database manipulation languages from vendors like Ashton-Tate, Microrim, and Borland. These proprietary DBMS languages will survive, of course. But they'll be in the form of "front end" access languages that interface to the SQL DBMS engines, which actually manage the data. This will happen because the SQL standard can provide a common interface among different vendors and because it's IBM's approach.

SQL is also the basis for client/server computing and distributed databases. Both capabilities let companies link together PCs with advanced software technology and make them into "industrial-strength" commercial database

SQL was adopted—as you'll see—for two reasons: political expediency and a recognition that any other DBMS language wouldn't be supported by IBM...

engines. These PC networks will be effective in replacing mini computers and mainframes at far lower costs. The one common denominator across these technologies will be reliance on the SQL database language.

IBM and UNIX

For the most part, IBM has been a non-participant in the UNIX market of the 1980s. It's not unreasonable, however, to forecast a future in which IBM's impact on the UNIX market may be as great as AT&T's. This is because UNIX will play a large role in commercial data processing during the next decade. A key event in triggering the changed attitude toward UNIX was the 1987 formation of the Open Software Foundation (OSF), a second source for UNIX technology.

Prior to the formation of OSF, UNIX was already widely used in certain markets such as workstations and supercomputers. The OSF was unique because its founding members included both IBM and DEC—companies that are normally in an adversarial position. OSF also announced that the IBM AIX implementation of UNIX would be the base technology for the OSF/1 operating system.

In one brilliant stroke IBM managed to move into the enemy camp and take a long step toward controlling it. Why enemy? Well, UNIX is going to be a lot more than an operating system. It represents a code word for a whole set of programming standards, including data management, graphical user interfaces, electronic data interchange, and communication standards that are required to implement modern programs. IBM has such a set of standards for its proprietary architecture—Systems Application Architecture (SAA—discussed below). The UNIX set of standards will provide an alternative and competitive environment to SAA. IBM, by joining and endorsing the OSF, has said, in effect, they're going to play on both sides of the fence. And IBM plans to provide distributed DBMS capabilities, based on SQL, that will allow its customers to build applications that span data located on both SAA and UNIX SQL databases. I have little doubt that in the 1991/1992 timeframe IBM will emerge to be a major player in UNIX hardware and software.

IBM and graphical user interfaces

Macintosh-like graphical user interfaces (GUIs) with pull-down menus, mouses, and icons are bound to become more popular in the 1990s. Current workstation power in PCs creates affordable environments for implementing such technologies. Standards for GUIs will be just as important as standards in database management. Most users of computers, after all, will see the graphical interface, while the DBMS will remain hidden "under the covers."

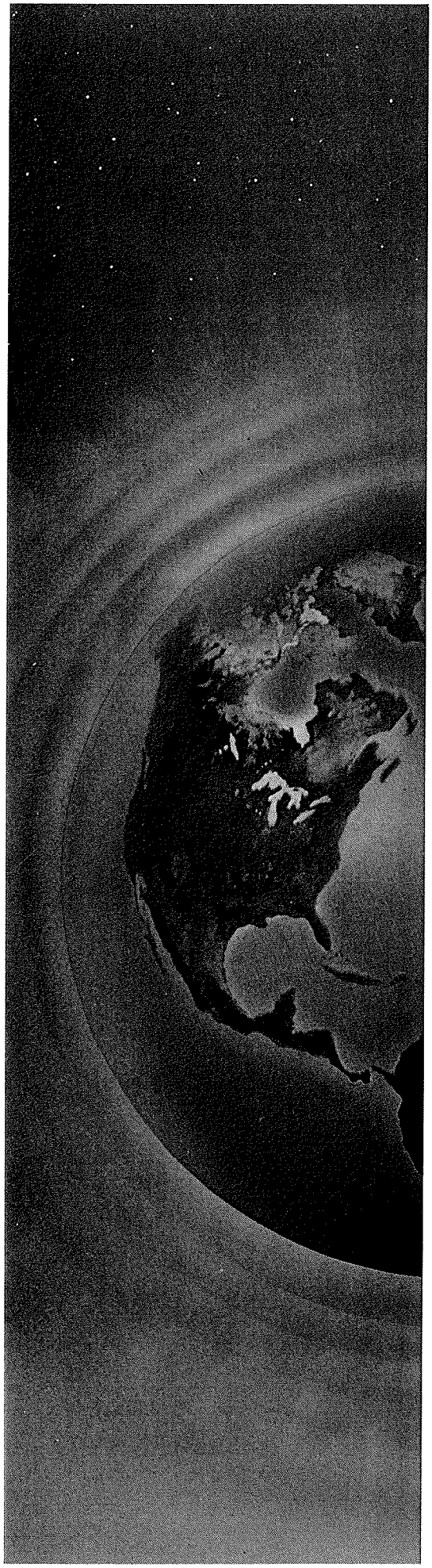
IBM is playing a critical role in the development of GUI standards. IBM's partner, Microsoft, has developed the Presentation Manager GUI, a critical component of both Microsoft's OS/2 and IBM's SAA. If users of PCs want a GUI capability, they are overwhelmingly likely to acquire products based on Microsoft's Windows and Presentation Manager technologies, both of which are endorsed by IBM.

Presentation Manager is also going to have an influence in the UNIX world. OSF/Motif, the GUI standard for OSF's version of UNIX, is based upon the look and feel of Presentation Manager. Given that Hewlett-Packard, DEC, IBM, and many others are supporting OSF/Motif, it's likely that Presentation Manager's look and feel will be predominant not only in the SAA universe, but in the UNIX world as well.

IBM's Systems Application Architecture (SAA)

SAA is IBM's vision for building a uniform and connected computing environment across IBM mainframes, mini computers and PS/2s. IBM's principal architectural approaches involve two-tiered computing with a local workstation, the PS/2, connected inter-

In one brilliant stroke IBM managed to move into the enemy camp and take a long step toward controlling it...



actively through LAN technologies to an IBM mainframe host. IBM expects its customers in the 1990s to build applications that will operate cooperatively with most application logic executing at the PS/2 level with database processing going on at the host.

The application software systems built for PS/2s will use GUI interface services built on top of Presentation Manager that conform to a "Common User Access" (CUA) specification. CUA is not a program but a set of standards that define how applications are supposed to look, what the role of icons are, where they're to be placed, how menus are accessed, etc. CUA software products will be designed and built by companies that expect to sell SAA-compliant applications to the IBM user base. This means there's going to be a lot of software available for OS/2 and PS/2 and clone hardware that will be CUA-compliant. This software, in cooperation with the IBM database standards for SAA, will access SQL databases both locally and remotely. The availability of this software is certain because independent software vendors want to be compliant with SAA as a way of making their wares attractive to the Fortune 1000, IBM's primary customer base.

CUA compliance means software developers will be able to develop automated screen generation tools for building menu-based applications. This will speed the delivery of new applications software. Also, a standard GUI means users can enjoy the ease of access that common approaches provide (just like in the Apple Macintosh world). Likewise, even non-Presentation Manager products, such as the character-based FoxPro, attempt to provide CUA-compliant interfaces—further demonstrating IBM's influence beyond its products.

IBM and the CASE market

The Computer Aided Software Engineering or CASE industry's goal is to develop software techniques that allow users to build customized applications far more efficiently than traditional methods (sitting down and creating flow charts, followed by hand coding into BASIC, COBOL, or C). In a CASE environment, the application developers work with diagramming and modeling tools at the workstation level. These graphically oriented devices allow the specification of a problem. That specification is then stored in a multiuser, commonly accessible "repository." The repository, in effect, becomes the database for application development efforts. After the analyst has completed the specification of the problem, a code generator piece of software "compiles" the specifications and produces a finished application in some common language.

Throughout the 1980s, CASE was a cottage industry—many small companies sold individual and non-standard products. Last year, on Sept. 19, IBM forever changed the market when it announced a forthcoming set of products, Repository Manager, and an environment called AD/Cycle. This announcement was IBM's response to the growing interest in CASE. In effect, IBM said it was creating new standards for the way CASE tools would work and interact with each other.

Before the announcement, most CASE tools only covered part of the full life cycle of development activities. When there was a demand for an interface between tools from different vendors (e.g., interoperability between vendor A's design software and vendor B's code generator), such an interface would be custom built. IBM's announcement suggested that these two-way interfaces wouldn't be needed anymore because the IBM product, Repository Manager, would provide a standard set of guidelines into which toolset vendors would interface. The IBM standard would also define what forms of modeling and modeling tools should be provided. Some time this year, IBM should provide more detail specifying the kind of modeling logic its customers

will use, along with a tools interface that cooperating software vendors will follow to interface to Repository Manager.

An important part of the AD/Cycle CASE standardization effort will be approaches for supporting the design of databases. The Bachman Re-engineering Toolset (Bachman Information Systems, Cambridge, Mass.) has been selected as the preferred approach within AD/Cycle for database design. This means that over a period of time, the Bachman approach, as modified to integrate with Repository Manager, will become widely used in IBM customer shops that adopt AD/Cycle.

The AD/Cycle announcement is expected to bring order to what has been a chaotic, splintered market. If IBM can successfully deliver software products to implement its repository view of CASE, it will then establish standards for the CASE market in much the same way that the advent of the SQL language established a standard for the database. This will mean, however, that independent tools vendors for mainframe and PC environments will have to redevelop and interface their products to be compatible with the IBM AD/Cycle approach. We're likely to see this happen on a continuing basis through the early part of the 1990s. The result by 1994/1995 is likely to be a good standards base for CASE technology and a much wider use of CASE software products than we have seen to date.

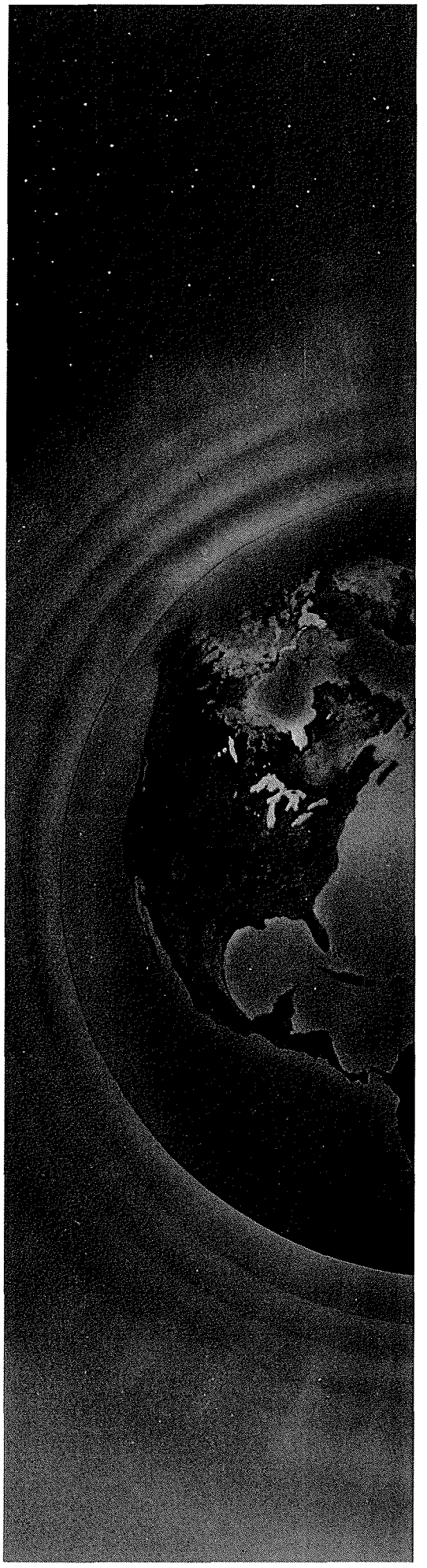
IBM and Independent Software Vendors (ISVs)

My premise so far in this article has been that IBM's market power can clearly create standards in hardware and software products and that those standards powerfully influence the development of products for commercial data processing. In addition to simply creating standards, however, IBM's product positioning can make or break independent companies.

One of the greatest software success stories of the 1980s has been Oracle Corp. Oracle's sales volume is expected to top $1 billion in the 1990 fiscal year. Oracle was only formed in 1979 as a company to market ORACLE, its SQL DBMS. Oracle's success during the 1980s can be attributed to a very small number of correct decisions that were made early on:

- The database programming language Oracle chose was SQL, an IBM development that hadn't even been announced as a product yet.
- In addition to providing the SQL database management language,

Last year, on Sept. 19, IBM forever changed the CASE market when it announced a forthcoming set of products



Oracle built an integrated environment of application development tools to surround SQL.

- Oracle implemented the base IBM idea, not on IBM computers, but mostly on other mini-computers and workstations. Later, Oracle's products were ported to IBM PCs and mainframes. So Oracle's early, very rapid growth and success was achieved by taking an IBM idea, improving on it somewhat, and providing it on non-IBM platforms.

Another great software success story of the past decade is Microsoft. This company, which also was formed only 10 years ago, has grown to a current sales rate of over $1 billion a year, largely by building on the fact that IBM chose MS-DOS as their PC standard operating system. Since then Microsoft has evolved away from simply being a subcontractor of IBM to being a full-fledged partner in establishing PC directions. IBM has tried to move away from the Microsoft standard, for example, in implementing its customized extensions to OS/2. That move hasn't been very successful (as I'll discuss in more detail shortly). It's likely that Microsoft, partnered with IBM, will continue to have an overwhelming influence on the PC marketplace.

As part of the AD/Cycle announcement, IBM brought in certain strategic business partners whose purpose was to create software products to compliment IBM's strategy for the CASE market. The three principal software vendors to benefit from this arrangement are Bachman Information Systems, Index Technology, and KnowledgeWare. All three sell analysis and design software technologies. They have agreed to redevelop and enhance their products to be compatible with AD/Cycle and are, almost without doubt, going to benefit enormously from the IBM affiliation. If AD/Cycle is a significant failure (unlikely, in my opinion), it isn't clear that IBM's business partners will be damaged. They'll still have received publicity, sold test software, and moved into a position to go in and carve out pieces of the CASE market for themselves.

IBM doesn't win them all

While there are clearly some important success stories that can be told by vendors that have properly cooperated or integrated with IBM's software strategies, there are also some stories of failures. For example, several years ago Comshare's System W decision support system was selected by IBM for cooperative marketing. During a conversation I had back in 1987 with the then V.P. of marketing for Comshare, almost no additional sales came to Comshare because of the IBM connection. IBM salespeople were unable to understand and/or successfully push System W.

Another good idea that didn't pan out was the combination of the ORACLE DBMS and the IBM PC/RT. My friends at Oracle were ecstatic when their product was picked as the database management system for the RT RISC computer. Unfortunately, the sales of the RT didn't live up to expectations. Therefore, Oracle's sales for that product weren't significant.

Another example of a failure caused by IBM's software market strategy is Cullinet. Cullinet was the great software success story of the 1970s. By 1984 Cullinet had grown to $300 million per year by selling IDMS, a database management system that implemented the CODASYL "standard" on IBM mainframes. IBM chose not to develop a CODASYL DBMS. Instead, it sold its hierarchical database manager, IMS. IDMS had software capabilities and productivity benefits that made it in some ways technically superior to IMS. IBM's response to Cullinet's success was the creation of an entirely new database management system, DB2, based upon the relational model. IBM aggressively marketed DB2 with a six-month free trial of the software. The result: new sales of IDMS dropped to under 20 per year. Ultimately, by 1989 Cullinet was unable to stay in business after several years of losses and was acquired by Computer Associates.

Continued

Of course, a software vendor that follows IBM strategy closely isn't automatically guaranteed success. There are some notable examples where IBM has failed with software strategies; likewise, there have been plenty of companies that thought they were been properly aligned with the IBM approach, but found out later they weren't. Here's a quick list of some of IBM's on-going problem areas:

1. VSE. This is IBM's operating system for its smaller client base—companies with machines like 4300s and 9370s. VSE has a large base, but some of IBM's newer, enhanced software capabilities for VSE—like its relational DBMS, SQL/DS—haven't been highly rated by customers. VSE doesn't participate in SAA; therefore, the new AD/Cycle approach won't be available to VSE customers. Other companies, led by Computer Associates, more oriented toward smaller mainframe environments are likely to be more successful than IBM in selling software products to these customers.

2. OS/2 Extended Edition. This customized and enhanced version of Microsoft's OS/2 includes the IBM Communication Manager and Database Manager products. This operating system has been notably unsuccessful in the marketplace. Some of the reasons have been technical:

- IBM's LAN Server is expensive compared to Novell's NetWare.
- LAN Server is considered feature poor.
- LAN Server requires expensive mainframe support for full connectivity to the IBM environment.
- There's little to no support for third-party products like DCA's IRMA board (a popular PC product that allows emulation of IBM 3270 terminals).

But perhaps the most important point is that OS/2 EE doesn't run on non-IBM hardware; there are very few customer sites running only PS/2s. In an attempt to address this problem, IBM and Microsoft announced at last Fall's Comdex an agreement to "open up" OS/2 EE. The approach would have Microsoft port the EE environment to non-PS/2 hardware platforms.

3. Micro Channel Architecture. This proprietary approach for the bus on PS/2s has only been modestly successful in the marketplace. While IBM has sold millions of PS/2s, its share of the overall

American PC market has declined to about 20 percent from its 1984 high of 40 percent.

4. Workstations. IBM's PC/RT workstation computer turned out to be a market failure largely because of inadequate hardware performance and no compelling software advantages.

5. Laptops. IBM's total lack of competitiveness in the most rapidly growing segment of the PC marketplace remains incomprehensible. Companies like Compaq, Toshiba, Epson, and Tandy all seem to be able to make significant headway where IBM, with a far larger research and development budget, can't create a competitive product.

My conclusions

If you buy, use, make, or sell database tools, IBM's influence is bound to affect you. Over the next few years, as people interact with DEC mini computers, more of them will do so with GUIs that resemble SAA's CUA and Presentation Manager. As Macs become more widely used in business, interfaces to IBM SQL client/server databases will allow those Macs access to corporate data. As UNIX platforms become more widely used for commercial applications, we're likely to see OSF/Motif being used to access the same IBM databases as the Macs. It's likely that the information modeling technologies that IBM delivers in AD/Cycle will appear eventually in AIX and then the general UNIX community. Over the next several years, PCs will become mostly networked and the days of lone PCs and sneaker-net will end. These networking and data interchange standards can only happen when there are good market standards to build on. Those standards are likely to be the ones that IBM software laboratories have developed and implemented.

George Schussel is President and Principal Consultant of Digital Consulting Inc. (DCI), Andover, Mass. He founded DCI in 1981 with the purpose of providing educational seminars with a software focus. DCI has grown to be one of the world's largest IS oriented seminar, conference, and exposition promoters. Schussel's role at DCI is to consult and research future software industry directions. Conferences that he has created and chairs include: Database World, Software Futures, Schussel/Yourdon on Emerging Software Technologies, Application Development in the '90s and the UNIX Today! Conference. ❐