KBN

The Guide to Open Systems

VOL. 5, NO. 11
ISSN: 0887-3054
NOVEMBER 1990

INSIDE

EDITORIAL

NEWS ANALYSIS

PATRICIA SEYBOLD'S

UNIX IN THE OFFICE

Open Systems Defined

A Complex Evolution from Propietary

By Judith S. Hurwitz

OPEN SYSTEMS is becoming the protective mantle for vendors and users alike as each group grapples with making the transition away from proprietary technology. To avoid repeating the systems problems created by multiple architectures and operating systems, users are looking for ways to protect themselves from the unavoidable vicissitudes of the computer industry. Open systems is the beginning of the evolution from technology and systems that are isolated and incompatible to a new era when systems will have the flexibility to work together. *(continued on page 3)*

• E D I T O R I A L •

Organizational Transformation

In Tough Economic Times, Users and Vendors Have to Learn to Be Creative.

By Judith S. Hurwitz

THERE IS A chill in the air these days. The global economy is in a slump as business purchasing slows down. We expect that this turn of events will change forever the computer industry and the ways organizations use technology. Does this mean that users will stop buying computer technology and that vendors will go out of business? Not necessarily. But it does mean that the dynamics between customer and supplier will change dramatically. Users will, in the end, have greater leverage with their suppliers than ever before. Suppliers will have to work even harder to make each sale. And some of the mediocre suppliers (you know, the ones that always made you wonder how and why they even made money) will either fade from view or be swallowed up.

In this new economic reality, the less creative performers can no longer succeed by using the right buzz words. But the bright side is that companies can use these tight economic times to learn to do business more creatively and thus emerge stronger when times get better.

The smarter systems suppliers are beginning to understand that, to survive, they must adapt and adapt quickly to their customers' requirements. They are becoming less dogmatic about their proprietary technologies, and they are opening up their architectures based on what their customers tell them they need. IBM is a good example of this type of organizational learning. Many people who view IBM from a distance are convinced that the company is an ideological entity, hellbent on forcing its mainframes and Systems Application Architecture (SAA) down the throats of its installed base. This is the monolithic IBM that is populated by guys in white shirts and red ties (or is it red shirts and white ties?) who toe the company line.

The truth is much more intriguing. IBM is probably the most pragmatic company in the computer business. Therefore, if it hears that its customers do not want SAA as designed, it will change SAA. Users have told IBM that they want to be able to integrate other vendors' clients and servers into the overall enterprise. What is IBM's response? "No problem. We'll give you open SAA." Some of this is, indeed, just good marketing. But some of it is also an attempt to answer customer requirements. Therefore, don't be surprised if someday IBM lets users have Unix, Mac, or Windows clients for SAA. Don't be surprised if it allows customers to support users in having their various Unix, Hewlett-Packard, and Digital Equipment systems as a sanctioned part of SAA.

NCR is another example of this approach. The company did well in the 1980s as a Unix box-maker. Its Tower system was one of the key VAR platforms. NCR was smart enough to recognize that, to succeed in the '90s, it would have to become a software company. The result of this transformation is its Cooperation software environment.

So the moral of this story is that the companies that survive in the '90s will be those that learn to truly understand their customers and are ready to change. They will talk not just to their loyal, devoted customers, but also to the ones that want to throw them out on their ears.

Users are beginning to sense their power over vendors. It was inevitable, and, for the most part, it is good. But, at the same time, customers will have to take more responsibility for articulating what they really need. They will have to change from being passive recipients of technology to building a meaningful partnership with suppliers. It is not enough for users to say, "Give us something open." And it is not enough for vendors to promise open without explaining what that means. ◐

PATRICIA SEYBOLD'S
UNIX IN THE OFFICE

Editor-in-Chief
Judith S. Hurwitz
MCI: 3538836@mci.com
Internet: hurwitz@dcm1tp.das.net

Analysts and Editors
LAURE B. ROWAN
JUDITH R. DAVIS
DAVID S. MARSHAK
RONNI T. MARSHAK
MICHAEL D. MILLIKIN
JOHN R. RYMER

Managing Editor
DOUGLAS A. FREEMAN

News Editor
DAVID S. MARSHAK

Art Director
LAURINDA PHAKOS

Publisher
PATRICIA B. SEYBOLD

Sales Director
RICHARD ALLSBROOK JR.

Circulation Manager
DEBORAH A. HAY

Customer Service Manager
DONALD K. BAILLARGEON

Patricia Seybold's Office Computing Group 148 State Street, Suite 612, Boston, Massachusetts 02109
Telephone: (617) 742-5200 FAX: (617) 742-1028 MCI: psocg / 312 2583
Internet: psocg@dcm1sg.das.net TELEX: 6503122583

Patricia Seybold's UNIX in the Office (ISSN 0887-3054) is published monthly for $495 (US), $507 (Canada), and $519 (Foreign) per year by Patricia Seybold's Office Computing Group, 148 State Street, Suite 612, Boston, MA 02109. Second-class postage permit at Boston, MA and additional mailing offices. POSTMASTER: Send address changes to *Patricia Seybold's UNIX in the Office*, 148 State Street, Suite 612, Boston, MA 02109.

•OPEN SYSTEMS•

(continued from page 1) Unfortunately, like all complex stories, this one will not develop quickly. We fear that user organizations that are grabbing onto the open systems framework and demanding that vendors adhere only to de jure standards will have a rude awakening—at least over the next 5 to 10 years. This posture will make groups like the Houston 30 (now called the User Alliance for Open Systems) backtrack when it becomes clear that forcing vendors to implement de jure standards that are still far from being ready for the commercial marketplace will not be easy. We suspect that these organizations would be better served by a migration strategy with open systems based on de jure standards as a long-term goal.

Why Open Systems, Anyway?

In essence, open systems means never having to say you're sorry. The systems choices managers make have a dramatic impact on their political well-being in their organizations. For example, suppose you were the manager responsible for implementing a system from a computer vendor that has fallen on hard times. One of your biggest problems is that, over the years, you have formed a close, mutually beneficial relationship with that vendor. During the good years, the vendor helped you solve problems, gave you advance information about the technologies it was developing, and even let you test out a product before buying. The vendor was your ally when you had to make presentations to management. In some cases, these vendor sales and support people seemed to be part of the company. This environment has been the status quo for many businesses for the past 20 years.

In essence, open systems means never having to say you're sorry. The systems choices managers make have a dramatic impact on their political well-being in their organizations.

But more and more companies are becoming uncomfortable with such coziness. The discomfort hits home dramatically when your selected vendor—whether it be Data General, IBM, Wang, or Digital Equipment—has a bad quarter or two. Suddenly, management begins to question the wisdom of the decisions you made in the past.

Now, information systems (IS) managers are beginning to realize that choosing a system that is not so dependent on a single vendor is the only way to avoid political minefields. However, these same managers are wary of getting on the Unix bandwagon. Going out on a limb for Unix could have the same political consequences as adopting technology from a single vendor. After all, Unix could be a passing fad. If IS managers come out too strongly in its favor now, in the future they could be viewed as using poor judgment. In many circles, Unix still has the reputation of being an unsound and esoteric operating system used only by scientists.

At the same time, technology managers realize they cannot continue to buy technology as they have in the past. It is no longer acceptable to take huge risks by purchasing unproven technology without some cushion of insurance. That insurance is open systems. It is clear to many technologists that there are times when it will be necessary to try new, leading-edge software technology. Without experimentation, they face the risk of losing their competitive edge. So, if an organization can purchase a new technology that is based on accepted open standards, the risk is reduced. If software can be ported to other platforms, the risk to the organization is lessened.

While open systems symbolizes safety, it is also ripe for misinterpretation and confusion. Exactly what is "open," and how does it relate to the proprietary world? This report will define and put into perspective what open systems could mean to the industry in the long term.

What Is Proprietary?

Let's first look at what it means to be proprietary. In the computer industry, *proprietary* is the antithesis of *open*. But, as with everything else, there are shades of meaning. A proprietary offering is something that is controlled, developed, and licensed to users and developers by a single vendor. This vendor controls its development, its direction, and its use. It controls access to its code and collects licensing fees.

While this seems straightforward enough, the definition becomes fuzzy when a proprietary offering is widely accepted by a large percentage of the industry. DOS is a case in point. No one would argue that DOS is owned 100 percent by Microsoft. That makes it a proprietary operating system, right? Well, there are those who would argue that because DOS is a de facto standard on the desktop and because thousands of developers write software for its operating system, DOS is open. There are those users who would argue that systems such as IBM's SAA and Digital's NAS are open because both companies have published their programming interfaces so that third parties can integrate their applications within these architectures. A more extreme case might exist where there is a completely closed system that allows other systems to transfer information to it via a gateway. Is this a form of open? From the perspective that the transition to open systems will be an evolution, the answer would be yes—in the short term.

SPECIALIZED/PROPRIETARY. There are times when an open system based on industry standards is simply out of the question. When an uncharted area of the industry or technology is evolving, no standards may be available. In that case, the vendor and the users requiring this functionality have no choice but to introduce technology based on specialized, proprietary hardware and software. For example, at the far end of the spectrum are the pure

proprietary offerings. These would encompass specialized systems such as a medical imaging technology. These are often based on specialized hardware and may even use their own operating systems because standards have not provided the richness of functionality needed. However, as technology and standards evolve, it would be reasonable to expect that the requirements of medical imaging vendors would be satisfied by mature standards. At that point, medical imaging systems would be able to use standards and, therefore, would become more open. Another example is redundant system technology that had been first developed on a proprietary operating system base because no standards existed to support online transaction processing (OLTP). At the time that companies like Tandem and Stratus were developing their technology, Unix had not evolved to the point where it could support OLTP. Likewise, X/Open and the IEEE had not begun to design specifications for standards in this arena. It would be foolhardy to imagine that vendors innovating in frontier areas should wait until consensus or legislated standards emerge before developing technology.

As the industry evolves to open systems, the definition of what is open will also evolve.

Defining the Shades of Open Systems

Because of this evolutionary transition to open systems, there is a continuum from proprietary to open systems (see Illustration 1). As the industry evolves to open systems, the definition of what is open will also evolve. In this evolution, then, the minimalist definition of open would assume published APIs and gateways into outside systems. The ideal definition assumes that users will be able to exchange information no matter what application was used to originate the data.

COEXISTENCE WITH LEGACY SYSTEMS. Under the broad definition of open (published APIs), a user of DOS or VMS could participate in an open environment because there are clearly defined interfaces. This is important because users cannot afford to throw out their older systems, which represent hundreds of combined years of software and applications development. Open systems, therefore, must presuppose that these systems can continue to exist and participate on whatever level they can function.

THE MIGRATION CHALLENGE. Because evolution is such an overriding issue, flexibility is at the core of the open systems movement until legislated standards mature to the point where a majority of systems are based on the same standards. One of the most difficult challenges for users is to develop methodologies for migration from proprietary systems and from de facto standards to software based on legislated standards. Many of the user consorta, beginning to flex their muscles, are insisting that they want only legislated standards as their foundation. In reality, however, there has to be an evolution. Take the example of IBM's LU6.2 protocol. When it was first announced more than eight years ago, it was implemented in a closed, IBM-only environment. Then IBM published the APIs so that others could access and gateway into IBM's implementation. These vendors then made this peer-to-peer networking protocol into a de facto standard. Now, the DTP Committee of Open Systems Interconnect (OSI) is considering LU6.2 as the foundation for an open legislated standard. Likewise, an organization beginning with TCP/IP will be able to migrate to OSI legislated standards as those standards mature.

HARDWARE DEPENDENCIES. The open systems momentum has emanated from the desire to achieve hardware independence. However, in most proprietary systems, the hardware and the operating system are inextricably linked. Open systems demands that hardware dependencies be removed from the operating system and the upper level functions such as user interface and file system. However, we contend that hardware dependencies are only one of the ingredients in the evolution to open systems.

EXCHANGING APPLICATION-INDEPENDENT DATA. In the long term, users will not have to be concerned about the formats used in their particular application packages. For example, if a user created an illustration using one drawing package, another user should be able to edit that illustration with another drawing package. In most cases, this can be accomplished only if both packages have agreed upon the same intermediate translation standard (TIFF or CGM). Assuming that the vendors have provided translation facilities, the user is still responsible for the translation process.

In some areas of technology, this type of transparent data exchange has been smoothly implemented. A good example is the use of Keyword's translator software black box for translating formats between word processors, spreadsheets, and draw packages. If a vendor implements Keyword's technology, then users on that system are able to transparently translate between different word processing systems. Keyword's technology is an excellent interim solution to the lack of standards. However, in the long run, a standard from a legislated standards body will provide a common format that all vendors will be able to adhere to. Once this standard is agreed upon and widely implemented, users will not have to perform any translation between applications.

But this does not necessarily happen quickly. The X.400 International Standards Organization (ISO) standard for electronic

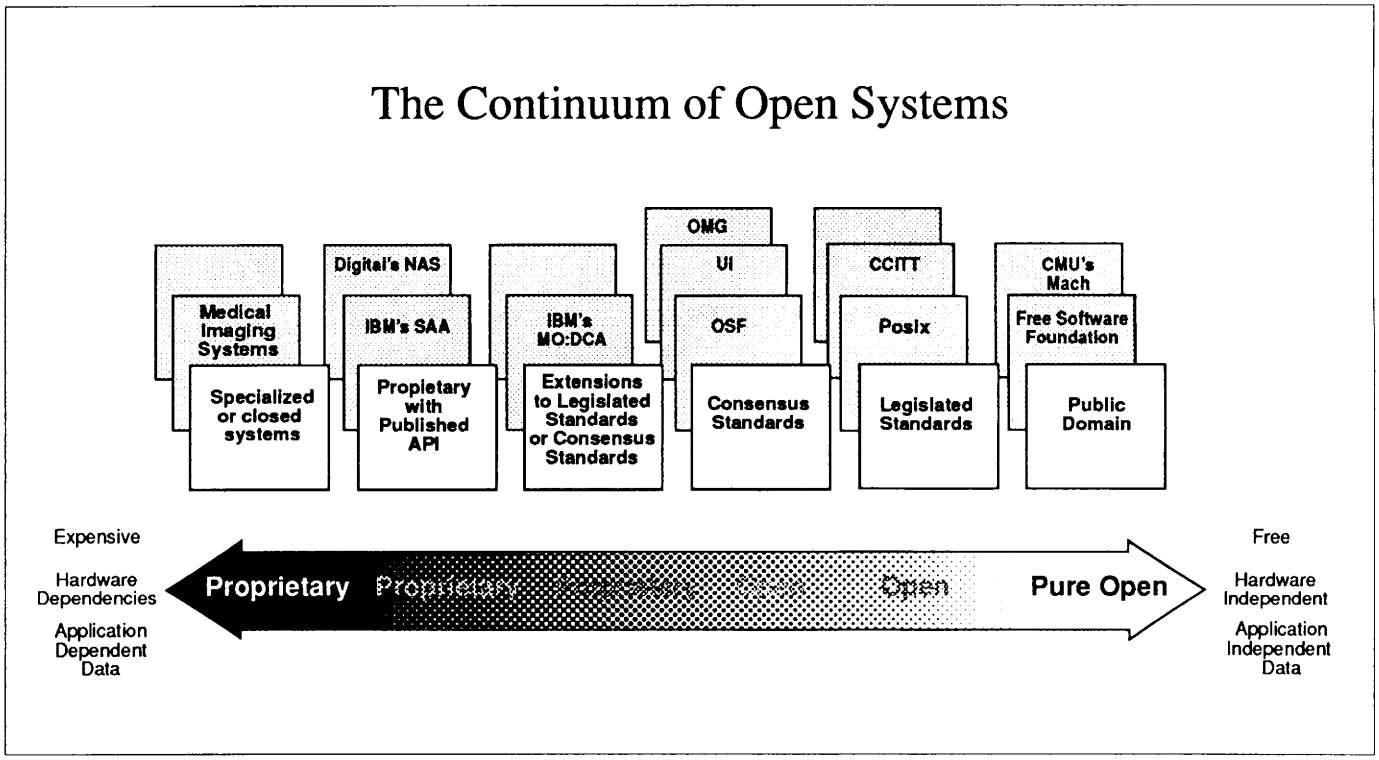


Illustration 1. This illustration demonstrates the continuum of the software world. On one end is software available at no cost. At the other end are the specialized, closed systems that provide technology not available anywhere else.

mail is a perfect example. While X.400 promises electronic mail interoperability, its implementation has led to unanticipated problems. Because a standard is, in fact, a specification, each vendor can interpret that specification differently. To make matters worse, the 1984 and 1988 versions of X.400 are incompatible. Thus, even if a company thought it was conforming to a de jure standard by writing to the 1984 standard, it would have to change all of its software to comply with the revised standard. Companies such as Soft•Switch provide a key translation between different mail systems based on X.400. Problems like this are solved in the long run when there are test suites to ensure that every vendor interprets the standards in the same way. But this could take up to 10 years.

The database field is probably the most important area for applications independence. Currently, users are finding themselves inextricably linked to their database providers, just as they have been tied to their systems providers in the past. To gain independence from their suppliers, users will have to learn to write applications that are not tied to the underlying database engine. Database vendors will have to adhere to emerging database and online transaction processing standards. For example, there will have to be better standards and enforcement of SQL standards so that users can take it for granted that SQL will be implemented on different databases.

THE EVOLUTION FROM DE FACTO TO DE JURE. Another step in the evolution to open systems is the de facto standard. In some cases, a de facto standard emerges when a majority of users implement a product because it fills an obvious void (such as DOS in the 1980s). In other cases, de facto standards emerge from public domain software that proliferates because it is offered without a licensing fee. As this technology takes on a life of its own, it is no longer controlled by the original developer. A good example would be X Window, developed by MIT but put into the public domain. Sun's NFS is an example of a de facto standard that came from a vendor.

In many instances, the technology that originates as de facto is used by legislated standards organizations and becomes the basis for de jure standards. Such is the case with both X Window and NFS. They both gained so much momentum in the market that they could not be ignored by the standards bodies.

Open Software Foundation: Pushing the de Facto

One problem with waiting for technology to gain wide market acceptance on its own is the time frame. It is increasingly clear that, to be competitive, systems developers want to implement the latest technology. However, at the same time, these companies want the assurance that what they adopt will be acceptable to users. Therefore, it helps them if de facto technology is pushed out into the marketplace faster than would happen naturally. This is where an organization like the Open Software Foundation (OSF) is making its most important contribution. The emergence of Motif is an example of pushing technology.

By taking the pulse of its members, OSF determined that an affinity with IBM's Presentation Manager (PM) and the use of X Window as an underlying technology were key requirements. Thus, through the Request for Technology (RFT) process, OSF was able to quickly bring to market a software product that is becoming a de facto standard. Most of the large corporations we have spoken to consider Motif to be their internal Unix user interface standard because of the PM look and feel.

Even more compelling is the selection by OSF of the Distributed Computing Environment (DCE). Again, by forcing a technology to market quickly, OSF has set up DCE as a de facto foundation for distributed networking for the coming decade. In addition, it has forced competing organizations to rethink their distributed networking strategies.

Once technologies such as Motif and DCE become commonplace in the industry, they can more easily become part of de jure standards from the standards-setting organizations.

Unix International

It is increasingly clear that Unix International (UI) has less to do with the open systems process than with ensuring that an implementation of one company's products is consistent with the needs of its licensees. Unix International is taking on a different market role from that of OSF. Over time, the view of the two organizations as competitors will lessen. UI is increasingly becoming the organization that pushes AT&T Software Laboratory to focus on the requirements of its members. While OSF has made progress in pushing the technology window, UI has taken on the role of making AT&T more responsive to its licensees. Therefore, if AT&T is debating the viability of two different technological directions, UI tells AT&T the direction it prefers. The pressure exerted by OSF has led AT&T to evolve System V a lot faster than it would have in the normal process of evolution.

Open Systems Survey Results

Over the past year, DMR Group, in association with UniForum and X/Open, conducted a study of 2,375 corporate users in the United States regarding their purchasing plans, with a focus on open systems. The users were questioned about their knowledge of and attitudes towards technology, applications, vendors, and challenges. Some of the results of this study are illuminating. For example, the survey points out that, in 1989, more business establishments in the United States were using Unix for the first time than any other multiuser operating system. While overall systems sales increased only 9 percent in 1989, Unix-based systems sales grew 40 percent. One of the most interesting statistics is that, of the companies using Unix in the United States, 25 percent were implementing Unix systems for the first time. Of the corporate users surveyed, one-third is considering moving to an open systems policy. One out of every six of the sites using Unix has explicitly adopted open systems standards.

Another interesting result of the study is the relationship between open systems and innovative technology. According to Don Tapscott, research director at DMR, researchers had expected to find that few technologies would be linked to open systems. In fact, the opposite was true. "It turned out that every innovative technology is a bigger priority for open systems adopters than nonadopters," Tapscott said. The two highest priorities for both groups are data interchange and interoperability. (See the illustration below.)

Of the 2,375 respondents, 55 percent agreed that they would use open systems technologies if the technologies communicated well with their existing applications. However, the survey also revealed that the industry is still in the

	Open Systems Adopters		Open Systems Nonadopters	
New technology priority	Rank	Mean	Rank	Mean
Data interchange among applications	1	8.3	1	7.2
Interoperability among applications	2	7.9	2	6.7
Online transaction processing	3	7.3	3	6.7
Local area networks	4	7.0	5	6.1
Systems based on IT industry standards	5	7.0	7	5.5
Network management	6	6.9	4	6.4
Database redesign	7	6.7	8	5.4
Distributed database	8	6.6	9	5.4
Multivendor network management	9	6.3	13	5.7
Real-time computing	10	6.2	6	5.7

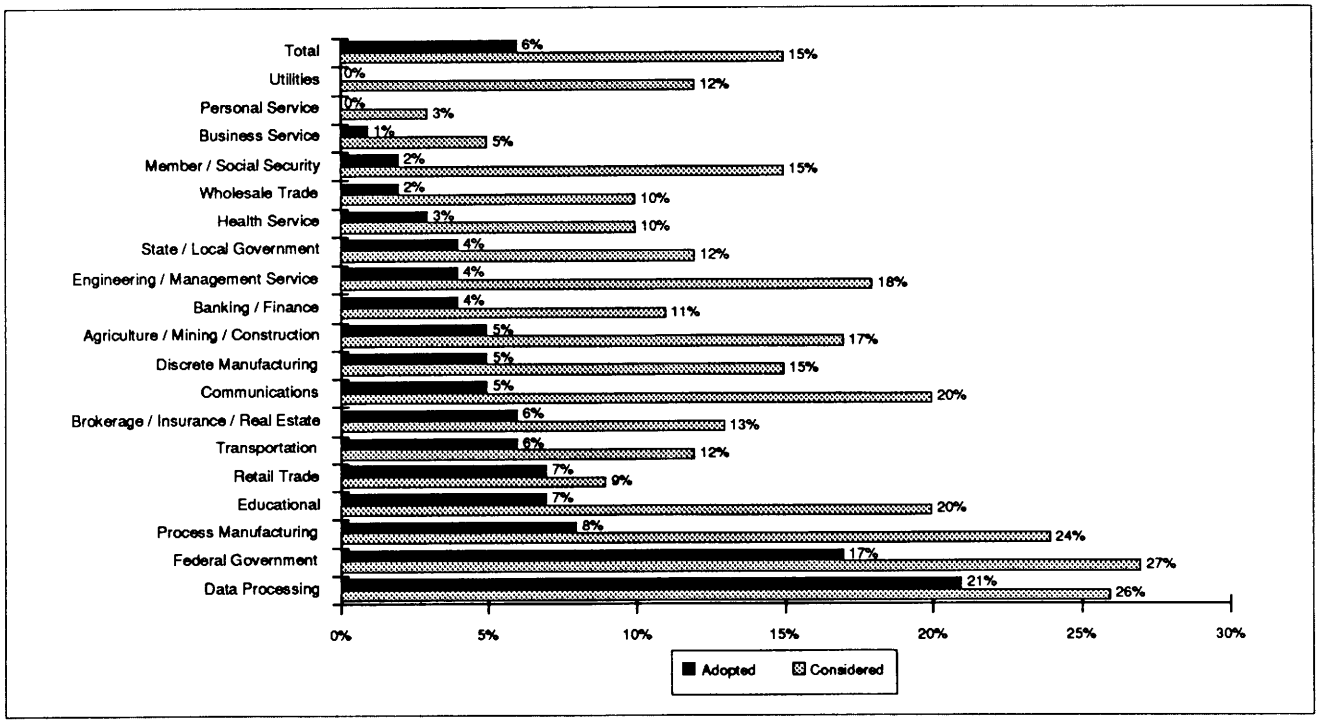
For each of 21 areas, respondents were asked to indicate their priorities for new investments over the next three years. Ratings were on a scale of 1 to 10 (1 indicated the lowest priority, and 10, the highest). This figure provides for the top 10 technologies, the overall ranking, and the mean (average) responses by adopters and nonadopters.

PUBLIC DOMAIN. As we previously stated, de facto technology often begins in the public domain. This software is freely available to anyone who wants to use it. It often comes from universities whose goal is to demonstrate their technical skill. Richard Stallman's Free Software Foundation is one end of the spectrum of public domain software. Stallman's philosophy is that no one should put a lock on ideas and information. Therefore, the technology developed by his small group of researchers is offered to the industry without restrictions and without licensing fees. While the notion that the flow of ideas should not be restricted is admirable, it is hard to justify in a world where development costs vast amounts of money. Then there are offerings that follow a different route. Technology is sometimes developed by one vendor and put into the public domain in the hope that it will become a de facto standard such as the Sun model we discussed earlier. Another twist on this model is the fact that Sun Microsystems put XView on the MIT X Window tape, making it public domain software. Since developers do not pay a licensing fee for the X Window tape and are not restricted as to how they use it, it can be defined as an open system. Some might argue that, to be truly open, this unrestricted software would have to be agreed upon by a standards body.

CONSENSUS. In a consensus organization, such as the Open Software Foundation or the Object Management Group (OMG), a group of vendors, developers, and (sometimes) users have requirements for technology they want to agree upon. Technology in this category can come from many different spheres. It could be standards-based technology, or it could emanate from the proprietary world. Usually, consensus organizations deal with software concepts that are too new for the standards bodies to become involved with. If legislated standards bodies have become involved, it is often too early in the standards creation process for vendors to be able to use emerging standards to

early stages of widespread adoption of open systems in the United States. Only 15 percent of respondents have considered a policy of using technologies and products based on vendor-independent standards, despite the fact that 42 percent agreed that open systems are important to them.

We think these statistics reflect the confusion and misunderstanding about precisely what open systems are. Users are indeed beginning to understand that having something open is important. On the other hand, until the benefits for using open systems in conjunction with existing technology are made clear, these users will not overtly commit to adopting an open systems policy. (See illustration below.)



This figure illustrates the penetration of open systems in various industry segments. Early adopters are in data processing and the U.S. government. According to the study results, open systems adoption has been slow in all other industry segments.

implement technology. But because of the rapid pace of change, these vendors need a mechanism to generalize this technology quickly so that third-party vendors have a common development target. The most obvious example of this consensus force is the development of Motif. Motif evolved from two directions: First, it took de facto standards such as X Window, and then it applied pragmatism—the need to conform to the look and feel of Presentation Manager.

X/Open: Concrete Specifications

X/Open holds a unique role as a consensus organization. Its charter is to establish an open systems environment both by taking de jure standards when available and by codifying de facto standards when no de jure standard exists. Therefore, the X/Open Portability Guide (XPG) has provided an important guidepost for those organizations making the transition to open systems. The XPG is, in fact, a linchpin document for all parts of the Unix and non-Unix worlds. Even more important are the test suites that X/Open has been developing to ensure conformance to its guidelines. Without conformance-testing, each vendor can interpret standards differently, as they can in the case of specifications discussed above ("Exchanging Application-Dependent Data").

X/Open's unique role as a consensus organization is to establish an open systems environment either by taking de jure standards when available or by codifying de facto standards.

This, then, is X/Open at its best. The organization gets into trouble when it tries to break new ground and set standards where no de facto standards exist. For example, X/Open reached an impasse when it tried to choose between OSF's Motif and OpenLook as the candidates for user interface. Vendors promoting their own candidate made sure that X/Open was hopelessly deadlocked in this effort.

The Object Management Group

The Object Management Group (OMG) attempts to achieve consensus when neither de facto nor de jure standards exist. What is more fascinating about OMG is that there are few commercially available object-oriented applications in the hands of users. If a truly open system requires that systems be able to exchange information at the object level, then an organization like the OMG could be a great help in making open systems a reality. One of the objectives of the OMG is to allow consensus among the world's leading software developers on the way objects can address each other.

The Object Management Group was a direct outgrowth of Hewlett-Packard's desire to make its NewWave environment and its Object Management Facility (OMF) the foundation of the industry standard. Over the past year of the group's existence, the OMG has taken on a life of its own (apart from HP's original intentions), although the Object Management Facility remains the group's focus.

OMG's first deliverable is a standards manual. Its goal is to provide guidance to developers to ensure that their object-oriented systems work together. The manual includes a "reference model" for object systems that defines the interfaces for interoperability, as well as an abstract object model. The OMG takes a minimalist approach to defining the object model.

Will the OMG succeed in leading the industry forward with a common object model? Perhaps. However, one danger is the organization's timing: It may be too early in the evolution of object technology. Also, there are those that are still suspicious of Hewlett-Packard's implementation of the object model and believe that more compelling models might emerge. One problem of adopting a technology as a standard too early is that vendors suffer the risk of adopting technology that is immature. By the time a newer model is available, those vendors may have invested too much time and money in the de facto standard to backtrack.

What will it mean if the OMG's model emerges as a standard? Well, it could be a significant boost to the user community. In theory, if all vendors adopted a common object model, then it would be possible to achieve true interoperability among objects no matter which vendor's system they were part of. However, to arrive at an object model that is acceptable to all vendors is difficult. First, the vendors participating in the Object Management Group are at varying stages of developing their object architectures. Some have already spent a considerable amount of time and money developing an object model that they are trying to popularize. To many of the members of the OMG, it is clear that a proprietary or "non-open" object architecture will have a difficult time outside its installed base. For a company to become well-established in this emerging technology area, it will have to provide a standard underpinning. At this stage, however, it is almost impossible to know what model will succeed. Do vendors hold back on their development until a standard is decided upon? Do they go ahead and develop their own interpretation and hope that the standards will be close enough to allow them to modify their design? This is a challenge for vendors trying to plan their next-generation environments and ensure that they will not be condemned as proprietary.

Legislated Standards

Legislated standards organizations are the obvious solution to the problem of standardization among different platforms. However, it is not necessarily a short, smooth path to arrive at consensus within the standards organizations. The IEEE and ISO are dominated by representatives of the major computer vendors. Therefore, each is chartered with representing its com-

panies' interpretations and implementations of technology. A lot is at stake in this process. For example, if one vendor can convince an IEEE committee to implement a large percentage of the vendor's implementation of what will become a standard, then that vendor has a technology advantage in time to market. This means that the standards process is tedious and time-consuming. Remember, it has taken almost 10 years for the ISO protocol stack to emerge.

When these standards do emerge, they can have the optimal outcome: to provide all vendors with a common implementation of a technology that makes interoperability easy to achieve. However, this is not always simple. Emerging standards are not cast in stone. When vendors first begin to implement them, there is a lot of room for interpretation. For example, two vendors that implement the ISO X.400 protocol in their electronic mail systems may have different interpretations of what the standard mandates. Initially, when these two systems attempt to exchange messages, inconsistencies will make the communications unworkable. However, over time, both vendors will change their implementation so that the differences disappear.

When these standards do emerge, they can have the optimal outcome: to provide all vendors with a common implementation of a technology that makes interoperability easy to achieve.

Another problem confronting those organizations that implement de jure standards is that the standard often begins life implementing the lowest common denominator. Take the Office Document Architecture (ODA) Standard as an example. Here, the standards bodies are adopting a model that provides a low level of functionality. Therefore, a vendor that wants to be standards compliant may be caught in an untenable position. On the one hand, if the vendor implements a pure, native ODA, it will undeniably be an open system. However, the functionality may be so limited that the vendor may be driven by competitive pressures to add advanced functionality on top of ODA. Therefore, while vendors are waiting for a standard to mature, it makes sense to provide transforms so that the implementation can communicate with other ODA-based systems.

Another compelling example of the foibles of the standards process is the X.500 Standard. One factor standing in the way of universal directory services is a standard for global naming authority. This is much like assuring that each user's telephone number is unique. In the case of the X.500 naming standard, the standards-setting bodies are currently deadlocked because each vendor would like its model to be adopted. Rather than resolving the issue, the X.500 Committee has postponed its decision.

THE ROLE OF POSIX. The original role of the IEEE Posix specification was to provide an API interface between the application and the operating system. As originally conceived, this could have provided some of the answers to open systems. What could be better than an open interface that would let a proprietary operating system evolve to include open system calls to applications also written to that specification? In an ideal world, all applications would become Posix compliant and all operating systems would comply with these standards. A proprietary operating system would be transformed into an open system.

In fact, this is precisely the tack that Digital Equipment has taken with positioning VMS and Ultrix. Digital is hard at work ensuring that VMS does provide the APIs that match the Posix specification. However, this is more a marketing ploy at this early stage than an open systems solution. The major problem is that there are almost no Posix-compliant applications on the market that could take advantage of a Posix-compliant operating system. Another difficulty is that, as with most of the standards processes, vendors are given some latitude as to how strictly they implement a standard. Therefore, one software developer might implement a conformant application while another would implement a compliant application. The difference? The degree to which the application includes all the details provided in the specification.

Another concern with Posix is its expansive nature. What had begun as a single committee chartered to construct an interface definition evolved into a series of 23 committees, each chartered with defining another aspect of the computing environment such as real-time and user interface. Suddenly, a relatively concrete, manageable concept became a complex and probably decade-long quest for an open operating system interface.

The Role of Enforcement

Because evolving standards are open to interpretation, enforcement and verification become critical issues. As mentioned above ("X/Open: Concrete Specifications"), the XPG conformance test suites provide a way for vendors to ensure that all are interpreting the specifications in the same way. Helping organizations conform to the ISO specifications has been one of the most important roles of the Corporation for Open Systems (COS). It is beginning to take an interesting twist since the newly-formed Alliance for Open Systems has become a member of COS (see "User Alliance for Open Systems," page 10).

How Users Can Leverage the Open Systems Movement

Information systems organizations are at a critical juncture. Clearly, many corporations are at a crossroads in terms of choosing their technology directions. IS organizations have lost power over the past decade. Developing strategies for evolving to open systems may be an excellent way to reestablish their leadership. Especially in these tight economic times, companies

User Alliance for Open Systems

The User Alliance for Open Systems, which began life as The Houston 30, is representative of the frustration that users feel toward the vendor community today. The User Alliance is not the only group with a charter to promote the user perspective on open systems. For example, there is the Petroleum Open Systems Consortium, and a group calling itself the Object Interest Group has formed in the United Kingdom to ensure that the user's perspective is represented in defining object management. Therefore, our concentration on the User Alliance, the newest of these groups, is intended to shed light on why these groups are forming and the opportunities and hurdles they face.

The User Alliance is different from other organizations in that it is intended to be a cross-disciplinary and cross-industry group. Clearly, it is beginning with the assumption that open systems are desirable. In fact, the overview account of its May Houston meeting states, "The road to success must include the development of a clear vision of what the term open systems means." One of its key assumptions is that open systems are "good" and that closed or proprietary systems are "bad."

How is the group defining open systems? "Open systems refers to standards-based, vendor-neutral information technology products. Vendor-neutral means interoperability with and application transportability to hardware and operating systems of competing systems that likewise are open systems." Within this context, the group has approved the work of standards bodies including ISO, CCITT, ANSI, and IEEE, as well as consensus groups such as MAP/TOP Users Group and X/Open.

At the Alliance's most recent meeting at a hotel across the street from General Motors Headquarters, the group decided to negotiate to become a working group within the Corporation for Open Systems (COS), with a role similar to that of the MAP/TOP group. Therefore, the Alliance would be considered a requirements interest group for COS's Industry sector. However, the Alliance is very concerned that it be perceived as an independent organization.

What does the group hope to accomplish? One of its first priorities will be to gain acceptance from its corporations' top management. Also, the Alliance understands that it must tie future technology architectures to the business goals of its member organizations. The group hopes to be able to provide leadership within the general user population. To accomplish this, it will work on such tasks as conducting its own research, developing case studies of successful implementations, and creating an information bureau for educational purposes. In addition, it intends to form an influence lobby to let standards organizations and consensus groups know about user requirements. The group will also help users lobby their internal management to move to open systems. To help users understand implementation issues, the group intends to publish a list of specific actions that can be taken now to implement open systems, primarily in the areas of payback and cost-justification.

The Alliance also has a second goal: to establish acceptance guidelines for open systems products. This will include promoting certification for products and a standards checklist.

The User Alliance for Open Systems has carved out a large niche for itself. We think it is tackling a lot of the right issues, but we have two major concerns. One is its ability to accomplish these very ambitious goals. The member companies will need buy-in from their upper management, and the participants will need to find the personnel to accomplish all the work they have committed to do.

Our other concern is the group's decision to make the adoption of de jure standards the primary objective of the organization. In the long run, the Alliance is correct. However, the organizers need to put on their real-world hats. They must understand that it will take many years before any corporation can implement an integrated enterprise environment based only on de jure standards.

Despite the obvious pain that the User Alliance will go through as it grows in sophistication, its efforts should help companies to understand what open systems are all about and what techniques others have used.

begin to question how they can better use the technology they already have and how they can plan for the future to make technology into a competitive advantage.

We therefore recommend that corporate information systems managers take a very pragmatic view of open systems. They should attack the implementation of this technology from several different perspectives.

INVOLVE UPPER MANAGEMENT. Ironically, some of the motivation to move from proprietary architectures to open architectures has come from an upper management push. From a financial perspective, it makes sense to choose technology that allows the user to have some leverage with vendors. Now that this movement is taking on a life of its own, information systems managers are beginning to understand that it is not safe to remain with a closed system.

When approaching upper management, IS managers must put open systems into a real-world context. No, open systems is

not an overnight solution to all problems. It is a transition from a narrow set of choices to a technology base that will be more adaptable. In the long run, it will make it easier for users to access all data, no matter what application or hardware they reside on. Therefore, upper management has to understand that implementation of open systems is not an immediate change in the way information systems are designed and managed, but a 10-year strategy that will have excellent cost and business benefits.

PLAN A TRANSITION TO OPEN SYSTEMS. The first step for IS managers is to plan how they will interconnect their legacy systems with their open systems technology. While the long-term goal should be the legislated standards, IS management must be ready to use whatever transitional technology is available. It is dangerous to insist on nothing but a de jure standard.

PILOT NEW STANDARDS-BASED SOLUTIONS IN DEPARTMENTS. When new technology becomes available, begin implementing it in departments first. For example, IS should work closely with user departments to implement distributed applications based on nonproprietary platforms. This will allow users and IS to gain experience without committing the entire enterprise. One of the most dramatic ways to convince upper management of the power of a certain solution is to pilot a technology and show an impressive result. One successful implementation is worth hundreds of meetings and memos.

BEGIN DESIGNING FOR THE FUTURE. IS should begin to separate the code programmers write from the applications used. This is especially critical in the database arena. Users are beginning to realize that they are just as dependent on their database vendors as they had been on their systems suppliers. We expect that the next target will be databases themselves. Users will begin to separate their data from the actual database so that the data will become portable. The sooner users gain experience keeping their code separate from dependencies, the sooner they will be able to keep their own applications open. One implication will be that users will have to stop trying to improve performance by taking advantage of application-specific design elements. It is just as troublesome to tie programs to a specific database implementation as it is to tie an application to a hardware platform.

The Fallout from Openness

One of the greatest challenges for users and vendors as we move into the era of open systems will be how to judge the quality of technology if everything is open. The conformance to standards will make it more difficult for vendors to sell technologies because so many of them will appear to be the same. Vendors will be forced to add value at more subtle levels, such as the way users manage technology. They will increasingly be promoting their ability to support and help users integrate their technologies. It will be no accident when suddenly every vendor becomes a systems integrator.

While having more openness will be beneficial for users, judging just how good a solution is will be increasingly difficult. Therefore, users will have to take on the greater burden of asking to be shown how well the new technology integrates with what they already have. Vendors will have to implement added value on top of standards in a way that protects users. In this new reality, it will be more important than ever for users to experiment to see firsthand how technology being promoted as open really stands the test of time. ◐

The title of next month's Unix in the Office *is "Solbourne Computer: A New Model for Innovation."*
For reprint information on articles appearing in this issue, please contact Richard Allsbrook at (617) 742-5200.

A Note to Our Readers

Ziff/Seybold Deal Does Not Include Patricia Seybold's Office Computing Group

On October 23, 1990, Ziff Communications Company announced the acquisition of Seybold Publications and Seybold Seminars run by Jonathan Seybold. This acquisition does not include Patricia Seybold's Office Computing Group (OCG)—the OCG was spun off from the original Seybold family enterprise in 1985.

Just as John and Jonathan Seybold have shaped the electronic publishing industry, Patricia Seybold and her team of analysts are the leading forces in distributed network computing, object orientation, workgroup computing, and open systems.

The OCG publishes three monthly analytical newsletters—*The Office Computing Report*, *Unix in the Office*, and *Network Monitor*—and an audio newsletter titled *Paradigm Shift: Guide to the Information Revolution*. The OCG also publishes special research reports, sponsors seminars, and provides customized consulting services to managers of information technology and to information system and software suppliers.

NEWS ANALYSIS

PRODUCTS • TRENDS • ISSUES • ANALYSIS

• INSIDE •

• DIGITAL •

From VAX to RISC

Digital Equipment is trying to change its image from a company that prizes its proprietary architecture and operating system above all else into a company that is perceived as a leader in open systems and RISC technology. Digital is following on the heels of competitors by boosting performance on its VAX systems and decreasing prices on its low-end RISC systems. The most significant change for Digital is its plan to move the VAX to a RISC architecture within the next two to three years.

PLANS FOR VMS. Part of Digital's Open Systems plan will call for VMS to become "standard compliant." Digital, as promised, will have Posix compliance for VMS by next year. VMS will conform to Posix 1003.1 (interface definition), P1003.2 (shells and utilities), and P1003.4 (real-time extensions). In addition, Digital intends to have the new improved VMS branded by X/Open (that means it will pass the test suite that X/Open has established for operating systems and for application conformance to its Portability Guide (Version 3). Digital has come to the realization that continuing to sell VMS, especially in Europe, will force it to move to as many standards as it can cram into its operating system.

Implementing DCE on VMS. Digital has stated that it will implement at least parts of the layers of the Open Software Foundation's (OSF's) Distributed Computing Environment (DCE) on VMS. Naturally, developers are beginning with the four components that Digital had a part in contributing, namely: the Remote Procedure Call (RPC), Distributed Timing Services (DTS), Distributed Naming Services (DNS), and Concert Multithreaded Architecture (CMA). They are investigating whether to port the other four parts of DCE.

VAX RISC. Now the company is saying publicly what we've been predicting for the last two years—Digital is going to change the VAX chip, and, within two to three years, it will be a RISC chip. Digital is quick to add that the MIPS chip in DECsystems will not obsolesce. However, we would suspect that, over time, the MIPS architecture might not be perceived as a critical technology.

Subtle Change in VAX/Unix Message. For the past year, Digital has been announcing that VAX and Unix were not a good union. If you need VAX, you probably want VMS; if you want Unix, buy the DECsystems. Now this message has changed. Digital says that purchasing Unix for VAXs is a wonderful idea. In fact, Digital's marketeers are eager to point out that the 9000 is an excellent Unix high-end system.

Ken Olsen Loves Unix. When Digital explained its new platforms and direction to industry analysts, Ken Olsen went to great lengths to talk about how important standards are to Digital. It was clear that the marketing folks were trying to keep him from making any new "snake oil" comments about Unix. Olsen did try to explain that quote as his attempt to equate all the claims for the openness of Unix with snake oil. In this new climate, where "Open Equals Goodness," someone might now take him at his word. Olsen is now on a standards kick, and, this time, he might actually come across as credible.

The Unix Story. The Unix story from Digital is getting more interesting. Digital has stopped talking about Ultrix as its own precious version of Unix. It now plans to have three versions of Unix for the future: OSF/1, SCO Unix, and System V. Digital is therefore going to drop its reliance on the Berkeley Software Distribution (BSD) base it has had for years. It will have its first OSF/

1 port in 1991. OSF/1 and its follow-ons will be Digital's strategic operating system for large accounts; SCO Unix and the Intel line will be Digital's strategy, first for alternative distribution channels (VARs), and then for the medium- and small-business markets. System V will be reserved for sales to the telecommunications industry. OSF/1 gives Digital a face-saving way to move away from BSD.

Have It Your Way. Digital is telling its software ISVs that they should port their applications to both VMS and Ultrix. It is telling its customers that they should do the same. "Feel free to move your applications to Unix boxes, and, if we let you down, you can port to another hardware base." This is a very different approach from Digital's heritage. No longer can the company assume that customers will trust it to always do the right thing. To regain customer confidence, Digital now has to prove itself all over again. It is starting by promising openness.

Motif. Digital used to talk only about DECwindows. Now, it has decided to talk a lot about Motif as the strategic user interface on both VMS and Ultrix. It announced the availability of the development kit for Motif Version 1.1, the newest release. Digital also announced VUIT—a visual user interface tool intended to help users build Motif applications.

BookRead. BookRead is a Documentation Library service for Ultrix that puts a graphical front end on system documentation. It will be available on CD-ROM.

Personal Computing Systems Architecture (PCSA) for Ultrix. Ultrix services for PCs were announced so that the 5500 (new DECsystem) can be used as a file server for PCs using TCP/IP or DECnet. This could be a very popular system for users who already have a lot of PCs. Digital has done well with PCSA on the VAX. It should boost the viability of its Unix servers in the commercial environment.

THE NEW HARDWARE. Digital has announced several new pieces of hardware in addition to announcing price changes and performance enhancements for some existing products.

VAX Systems. The VAX announcements were performance enhancements and price changes. For example, the VAX 6000-500 was changed so that performance was increased 85 percent and the price, 18 percent. The VAXstation 3100 performance was increased 100 percent and the price by $4,000. The MicroVAX 3100 had a performance increase of 45 percent and no price increase. Also, I/O has been improved on these VAXs. The only price decrease was on the 6000 Model 300, which was reduced by 30 percent. Digital added the RISC I/O subsystem from the 9000 to the 6000. This looks like a simple move to breathe more life into slow VAX sales, and it might help.

RISC DECsystems. Digital added two new models of its RISC server line, both aimed squarely at Sun's SPARC-servers. The DECsystem 5500 is the follow-on to the 5400 and is rated at 28 MIPS. It is based on the 30 MHz CPU, and includes a PrestoServer (NFS accelerator) on the CPU board. Digital claims that Sun sells this PrestoServer as a $6,000 option, and compares it to the IBM RS/6000/520, which is a 27.5 MIPS machine. Digital is selling the 5500 for a base of $63,000, compared to $57,000 for the IBM machine. The DECsystem 5100 replaces the 3100. It is a 19.4 MIPS machine that sells for $10,995. It competes with the IBM 6000/320, which is a 27.5 MIPS machine that sells for $20,735.

All the software for these machines have been "factory installed." Digital has also repriced the DECstations 2100 and 3100 to take on low-end SPARCstations. As a diskless workstation, a 2100 will sell now for $4,995 (the same as Sun's pricing). But Digital's 2100 can be upgraded with internal storage, while the Sun SLC is a closed box. The 3100 will now sell for $6,995. Digital is promising a follow-on to the 5800 sometime in the future.

CONCLUSION. Digital is sounding much better these days. It is not as mixed up about its two product lines as it has been in the past, and it is climbing onto the open systems bandwagon. With some work, Digital might just be able to convince some customers that it is indeed serious about open systems and standards. Ironically, this ill-defined thing called open systems may be more comfortable for Digital to take on, and it may help the company come to terms with Unix and non-Digital-invested technology more than anything else it has worked with lately. The new hardware announcements are not startling, but they are appropriate. They reposition the VAX to make it more competitive in price/performance; they add some competitive hardware in the RISC arena; and they put Digital back in the game. Now, Digital's challenge for the next two years will be squarely in the software arena. —*J. Hurwitz*

•SUN•

Distributed Printing

Sun's basic contention is that printers are poor network devices. They are connected with slow RS232 or parallel connections. The workgroup cannot easily access printer capabilities. Printing is a closed environment: Software is locked in ROM, innovation is controlled by a few companies. Sun insists (and we agree) that printing in the '90s should be network resident and a general purpose capability. Users should be able to send jobs to any printer regardless of manufacture. Sun's initial attempt at solving this is NeWSprint.

Sun's new NeWSprint software delivers PostScript imaging to PostScript and non-PostScript output devices anywhere on the network. In the samples we saw, output from an HP DeskJet looked every bit as good as that you'd expect from a LaserWriter.

The only configuration require-

ment is that the workstation, PC, or print server to which the printers and plotters are attached is running a copy of NeWSprint. When it is used in conjunction with OpenWindows, NeWSprint gives users the same image-rendering for screen and hard copy—complete WYSIWYG.

NeWSprint uses the workstation or server CPU to handle the imaging. Considering the power of the SPARC chips, this adds a great deal of performance to printers and enhances the image quality. Sun is removing the task of processing the image from printers and distributing it out to more capable platforms.

As further evidence of this redistribution of intelligence, Sun also announced a SPARCprinter: a low-cost, desktop laser printer rated at 13 ppm with 400 or 300 dpi resolution. The printer (based on a Xerox engine) has no processor or memory. It is a dumb device driven by a server or workstation using NeWSprint. It also costs a mere $2,695. As CPU capabilities increase, so does the performance of the printer (up to the constraints of the network connection). —*M. Millikin*

•SYBASE•

Mainframe Connection

In today's world, the standard approach to heterogeneous database access is a vendor-supplied gateway. In addition, the gateway almost always requires the use of some version (i.e., common subset) of SQL to communicate between the "native" and the "foreign" database management system (DBMS). Therefore, if you want to communicate between your relational DBMS (RDBMS) and another data source, the bottom line is the following. Rule #1: If the vendor doesn't provide the gateway, you're dead. Rule #2: If the data source you wish to access isn't SQL based, you're also dead—unless, of course, the vendor has provided the gateway (see Rule #1). Examples here are existing gateways to IBM's IMS (from Ingres) and to Digital's RMS (from Ingres and Progress). Neither IMS nor RMS understands SQL, and both require the translation of incoming SQL via the gateway.

Enter Sybase with a different approach to gateways, one that looks at the problem from the broader perspective of supporting an entire *business* transaction. Sybase views the distributed computing environment as a set of autonomous services that need a mechanism with which to communicate among themselves. To Sybase, vendor-supplied, SQL-based gateways usually solve just part of the overall problem for the customer, focusing on *database* transactions only and requiring agreement on a common subset of SQL. These types of gateways cannot provide adequate access to the myriad non-database, non-SQL types of information present in most large computing environments. In these cases, SQL simply cannot serve as a common method of communication.

Last fall, Sybase introduced its Open Server/Open Client APIs that allow the customer to integrate virtually any front end or back end into the Sybase network. (See Vol. 4, No. 11, News and Analysis Department.) The competition emphasizes, naturally, the fact that using the Open Server/Open Client APIs requires the customer/developer to write code to accomplish the integration. Wouldn't it be better if the nice RDBMS vendor did all that work for you? Sure, depending on whether the vendor would agree to do it, how much it would cost, when it would get done, and what functionality would be provided. In terms of non-SQL-based gateways, none of the RDBMS vendors will touch one unless the market volume and interest is there (e.g., IMS, RMS).

Sybase is now building its own gateway products using the Open Server architecture. Recently, the company announced three products that provide a platform for integrating IBM MVS mainframes into the Sybase client/server environment. With these products—the Net-Gateway, the Open Server for Customer Information Control System (CICS), and the Open Gateway to DB2—the customer can access any MVS data, applications, and services. Sybase is not just offering a gateway for DB2, as the traditional approach would dictate, but access to all IBM MVS mainframe data and services using a single set of APIs. (See Illustration 1.) This first set of products is limited to access through the CICS transaction monitor, but Sybase plans to provide Open Servers for other environments in the future, such as IMS/DC on the IBM mainframe. The big benefit for Sybase is that the company now has a coherent integration solution for customers with IBM MVS mainframes. This is a key requirement to meeting the needs of Fortune 1000 companies.

NETWORK GATEWAY. Net-Gateway provides the connection and protocol conversion between the LAN (client/server network) and the IBM SNA/LU6.2 network. To the client, the Net-Gateway looks as if it were a Sybase SQL Server. It maps database remote procedure calls (RPCs) issued by the client (which can be another SQL server or Open Server as well as any Open Client) to CICS transactions; the routing of the request is transparent to the client. The client can attach to multiple mainframes, multiple CICS regions within a single mainframe, and to multiple transactions within a CICS region.

Sybase has paid a great deal of attention to security and control requirements, which are particularly important in the mainframe environment. Net-Gateway includes the ability to define transaction permissions to limit users to a list of specified transactions, and connection permissions for limiting users to specified regions or subsystems within the mainframe. Net-Gateway also interfaces to mainframe security packages, such as RACF and Top Se-

cret. The Net-Gateway can reject a request that doesn't meet security requirements before it even gets to the mainframe.

Another critical area is system administration. Net-Gateway provides online monitoring and restart capability, an accounting log to track usage (e.g., for charge-back purposes), a data trace and error log, a timeout feature, and diagnostics for problem detection and debugging.

Net-Gateway currently runs on an IBM RT and will be ported to the RS/6000 workstation in the future. Another potential platform for Net-Gateway is OS/2 (although Sybase must first port the Open Server to OS/2). Sybase plans to port Net-Gateway to enough platforms to support at least DECnet, TCP/IP, and PC LAN connections to the mainframe.

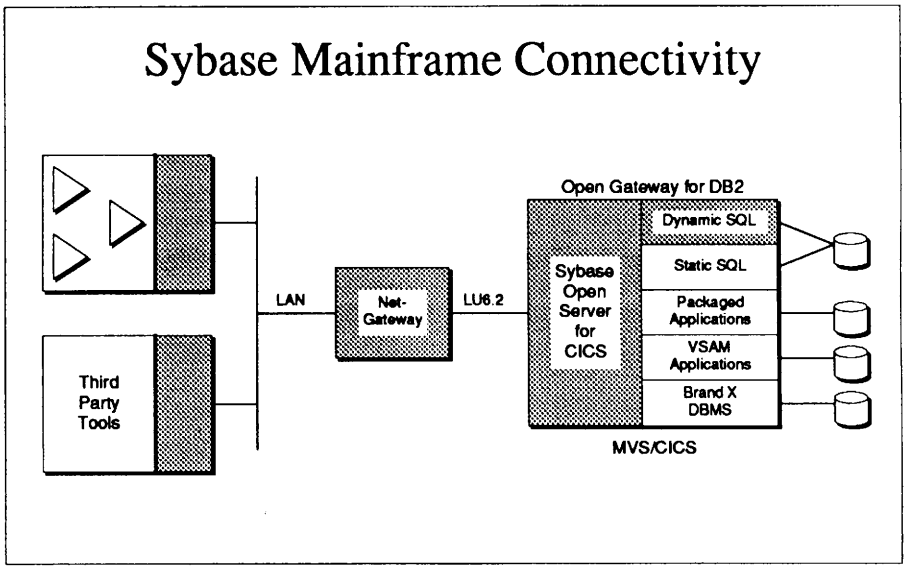


Illustration 1. Three new products from Sybase—the Net-Gateway, the Open Server for Customer Information Control System (CICS), and the Open Gateway for DB2—give users in a Sybase client/server environment access to IBM MVS mainframe data, applications, and services.

OPEN SERVER FOR CICS. The Sybase Open Server for CICS is a set of system services for CICS transactions. Through CICS transactions, the client can access any data on MVS, including VSAM, DL/1, sequential files, static and dynamic SQL access to DB2, and other DBMS data. The client can also access MVS applications and host services such as Job Entry Subsystems (JES), CICS queues, Intelligent Synchronous Communication (ISC) and Multi-Region Operation (MRO).

A major benefit of the Open Server approach is the option to use the same tools for accessing data sources that are used currently. Thus, giving additional clients access to mainframe data and services doesn't have to involve developing new paths to the data, with the concurrent headaches of maintaining control and security. The Open Server to CICS can be merely another way to access the current paths to data.

OPEN GATEWAY FOR DB2. The Sybase Open Gateway for DB2 provides access to DB2 plus all of the functionality of the Open Server for CICS. The DB2 gateway supports read and write access to DB2 data via dynamic SQL. It also provides automatic error-mapping and data type conversion (ASCII/EBCDIC conversion is done on the Net-Gateway).

The Open Server for CICS component of the Open Gateway for DB2 can be used to access DB2 using static rather than dynamic SQL. This is very important in production environments, since static SQL is pre-parsed, pre-optimized, and precompiled. Static SQL offers significant performance enhancements over dynamic SQL, which is the only option provided by competing DB2 gateways. Unlike using dynamic SQL through the DB2 gateway, using static SQL though the Open Server for CICS does require development on the part of the customer.

Sybase describes its gateway to DB2 as providing "turnkey" access to DB2. It is important to understand what this means: While the gateway provides automatic protocol, data type, and error code translation, the client must send DB2 SQL language through the gateway. The user cannot run a Sybase application against DB2 as if it were a Sybase SQL server, unless, of course, the application is restricted only to those Sybase SQL statements that are completely DB2 compliant. The benefit is the ability to use all of DB2's considerable functionality with no restrictions. The downside is the need to know DB2 SQL. For a customer who wants a common subset of SQL for heterogeneous communication, this will obviously not be the answer. However, it is also important to realize that the client does not have to be a Sybase client; it can be any front end that uses the Sybase Open Client APIs. So using Sybase SQL, or a common subset, may be irrelevant. Another point is that *client* is not necessarily synonymous with *end user*. A developer can put together a client application that accesses DB2, and the end user may never have to know any SQL at all, nor that the data comes from DB2.

Sybase does not translate Sybase SQL to DB2 SQL for two primary reasons. The first is performance. Every translation step in the process takes time. The second reason is functionality. Sybase does not believe the customer should be forced to use a common subset of commands to access data. This is the basic philosophical difference between Sybase and its competition: ease of use and development through commonality versus flexibility and functionality. Sybase will maintain that its approach is the correct one and

that there are a lot of customers out there who agree. And we don't doubt that some enterprising software developer will fill in the gap with the necessary language translation as an add-on product to the Sybase gateway.

AVAILABILITY. The Sybase IBM MVS connectivity products are all available now. Open Server for CICS is priced from $75,000 to $155,000. Open Gateway for DB2 (which includes Open Server for CICS) ranges in cost from $100,000 to $210,000. The Net-Gateway costs from $1,260 to $64,800.

—J. Davis

•ADMINISTRATION•

Security Solutions

As more organizations begin to implement Unix as a means for heterogeneous distributed computing, a weakness in the operating system has become especially dramatic: administration and, within that, security. Unix never evolved with the kind of administrative and security tools that commercial systems require. Unfortunately, systems vendors are all but shrugging their shoulders on this issue, leaving users to fend for themselves. We hope things may change—at least with security—once vendors start implementing the Open Software Foundation's (OSF's) Distributed Computing Environment (DCE), which includes a security standard based on Project Athena's Kerberos Authentication system.

In the meantime, many organizations are turning to third parties for support. Unitech is one example. A system administration software vendor located in Reston, Virginia, Unitech markets a security system called USecure, which features two tools for better security management: UPass, which provides password management, and UShell, a command restriction tool.

PASSWORD CONTROL. UPass is an add-on utility that manages password administration. It provides a layer over current login procedures, automatically reminding users to change old passwords and providing Help screens for them to do so, generating user profiles and audit trails, and the like.

Like Kerberos, UPass stores passwords as encrypted data; thus, no one can hunt for passwords within system databases. Furthermore, the system keeps an eye out for crackers trying to reach your system by creating reports with lists of who's logged into the system, who's not logged in, expired passwords, locked passwords, and unsuccessful login attempts. It also separates root privileges, keeping system administration distinct from security functions.

UPass also has a degree of customizability. You control login procedures on a per-port basis, and can grant dial-up access by user and by device. You choose the maximum number of login attempts allowed before access is denied and specify the wait period between attempts.

COMMAND RESTRICTION CONTROL. Unitech describes UShell as "an enhanced Bourne shell." The Bourne shell features a restricted mode, which limits the commands and directories users can use depending on their login. It's a practical way to keep reins on your system. Likewise, UShell provides extensive command restrictions on a per-user, per-group, or system-wide basis. Like UPass, UShell provides audit trails to keep reins on login activity. UShell lets you limit users to specific terminals as well as limit the number of superusers on the system.

CONCLUSION. These security features are certainly advantageous. They're also administrator tools, not developer tools. While a systems programmer might be able to configure your Unix system with similar capabilities, USecure has a relatively straightforward, menu-driven interface and online Help system.

USecure runs on an assortment of Unix platforms, including: Digital, Hewlett-Packard, Sun, Pyramid, Sequent, and Unisys. Licenses are available for $1,000 to $4,000, depending on CPU and Unix version.

—L. Rowan

•UNYSIS•

IIE: An Architecture, A Turning Point

Unisys's Integrated Information Environment (IIE) is the company's first unified architecture. Announced in early October, IIE also embodies Unisys's strategy to turn around its fortunes in a world dominated by other vendors.

Unisys doesn't set agendas in corporate information processing; IBM, Digital Equipment, Microsoft, Sun Microsystems, and others do. The IIE strategy recognizes this fact. Rather than challenge the leaders—particularly IBM—Unisys hopes to enter environments the leaders control and build on their successes with software development, online transaction processing (OLTP) products, and excellence in integrating diverse systems.

At the same time, IIE gives the users of Unisys's proprietary systems—the 1100/2200 mainframes (Sperry), the A Series (Burroughs), and CTOS (Convergent Technologies)—much greater multivendor integration opportunities. IIE shows existing customers how they can use standard networking today to introduce new functions and applications to mixed-architecture and mixed-vendor systems with minimal pain. And, for tomorrow, IIE shows proprietary customers how they can migrate to standards-based systems and/or Unix.

We analyzed Unisys's emerging architecture and strategy in April 1990 (see "A Study in Contrasts," *Network Monitor*, Vol. 5, No. 4). That analysis

holds true today, although Unisys has filled in more details—an X.500 directory service and Unix-based OLTP. The company's approach with IIE is cautious and realistic. IIE isn't flashy. It lays the groundwork for innovation atop standards, rather than introducing new APIs or infrastructures. IIE is a distributed computing architecture, but it doesn't redefine the leading edge of distributed computing.

IIE just might win Unisys a place in more corporate networks.

WHAT IS IIE? IIE encompasses strategy and a technology architecture designed to deliver on the strategy. Unisys's strategy: provide application services and platform software that allows the customer to integrate systems and information.

The technology architecture that implements the IIE strategy, the Unisys Architecture (UA), is to Unisys what Systems Application Architecture (SAA) is to IBM. UA provides a single framework of networking and data management facilities across Unisys's diverse systems. It defines applications services across platforms. And UA defines services, such as SNA networking, that allow Unisys systems to interoperate with the systems of other vendors. (See Illustration 2.)

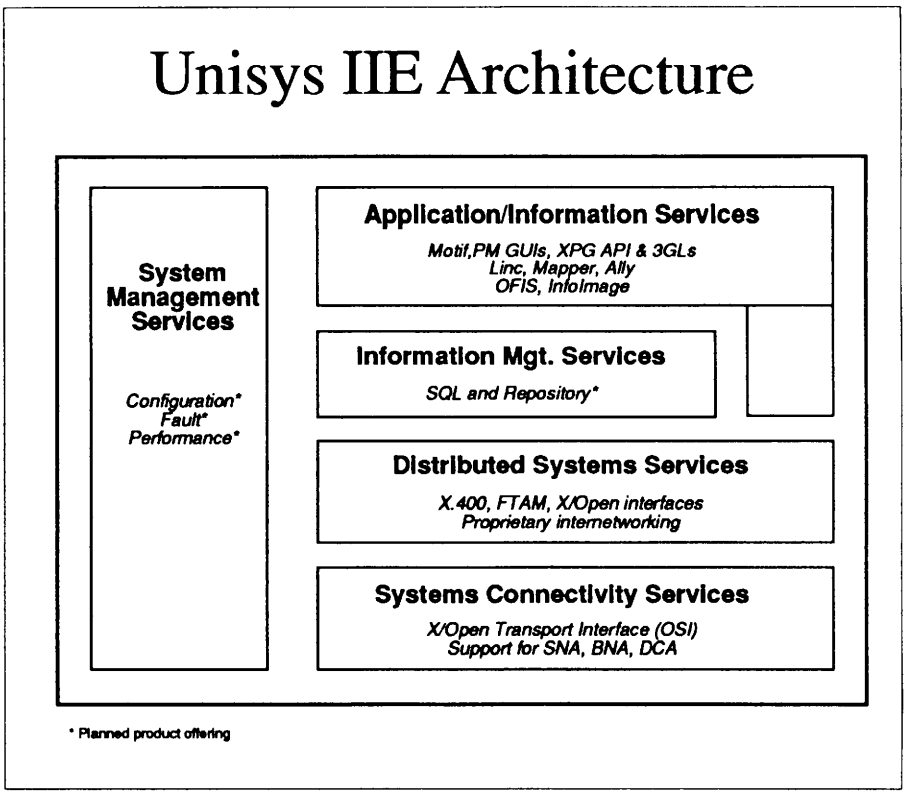


Illustration 2. Unisys's Integrated Information Environment is built on a series of cross-platform software services, from basic connectivity services to common APIs to planned system management offerings.

The Platform View. The goal of the Unisys Architecture is to allow users to work with three classes of computer architectures: Unisys's proprietary systems, Unix-based "open" systems, and systems made by other vendors.

- Proprietary systems. Unisys's primary thrust is adding the OSI-transport services defined by X/Open Limited in its X/Open Portability Guide Release 3 (XPG3) to the 1100/2200, A Series, and CTOS platforms. Unisys is establishing a common base of transport and information-distribution services for its various platforms.

- Open systems. Unisys's definition of an open system can be found in its Value Added Platform (VAP). A VAP is built on a software base of Unix System V and OSI transport services, with Unisys applications such as the OFIS office suite, EDI services, and imaging services.

- Other systems. Unisys has also added its well-regarded SNA interconnection software to UA's connectivity services. Where's DECnet? Unisys has a U Series-DECnet connection today, but it isn't part of UA. Unisys is betting that Digital's migration of DECnet to encompass OSI standards will allow UA-DECnet integration at some future date.

The Application Services View. Applications portability and interoperability are the reasons Unisys has identified and implemented common services across the UA platforms. Common services span all of the UA platforms, including those made by other vendors.

Information Management Services comprises a generalized database access layer, which includes Oracle's SQL implementations today, access to the Codasyl DBMS on the 1100/2200 series, and structured file systems such as Sun's NFS.

Unisys will add support for other SQL implementations in the future, as well as support for access to object-oriented DBMS and repository services. An SQL interface to the Semantic Information Manager (SIM), an object-oriented DBMS for the A Series, is due in the first quarter of 1990. We expect Unisys to announce at least a repository direction, if not a product, in 1991; the company needs a repository to make its application-development tools strategy work. Object-oriented DBMS is a longer-term project. However, Unisys has a base upon which to build—its Semantic Information Manager. We expect Unisys to generalize SIM as a cross-platform DBMS.

The Application and Information

Services part of IIE contains four major areas: common user interface, common APIs, common development facilities, and end-user information services.

Common User Interface. IIE's common user interface is the look and feel shared by the Open Software Foundation (OSF) Motif, OS/2 Presentation Manager, and Windows 3.0. The same look and feel will be implemented across the proprietary platforms. In adopting this strategy, Unisys is basically turning its back on the Macintosh as a full participant in IIE.

Common APIs. IIE's APIs are those defined in the XPG3.

Common Development Facilities. IIE includes support for XPG3's third-generation languages standards.

IIE anticipates that Unisys's Linc, Ally, and Mapper application-development tools will be used to create both applications that are split across platforms and portable applications. These tools are available today on many platforms, including third-party systems. In the near future, Unisys plans to build an integrated workbench based on these tools.

End-User Information Services. IIE encompasses Unisys's OFIS suite of end-user applications and tools.

UNISYS'S VALUE TARGETS. The ability to move information (file transfer and mail) and applications (portability) from one platform to another won't be enough to get Unisys into new user sites. Unisys hopes to innovate in three broad areas, each of which leverages UA's base of standard connectivity and applications services. The areas are: CASE and application-development tools, Premium Services, and distributed systems management.

CASE/Application-Development Tools. Linc and Mapper are arguably the most widely used CASE tools. Linc is aimed at generating large systems from a single specification. Mapper is an end-user system development and support tool. Ally, a relatively new tool, generates workstation-based, client-server applications.

In IIE, all three tools occupy a central role. IIE defines an infrastructure of services across a variety of systems. Linc, Ally, and Mapper allow users to more easily build applications atop that infrastructure. Unisys has a three-part strategy for making sure this happens.

- Unisys is positioning Linc, Ally, and Mapper as the foundation for an integrated suite of CASE tools. Integration awaits Unisys's decision on a repository.
- Unisys is accelerating deployment of graphical versions of Mapper, its main end-user development tool. Mapper was designed with a character-based interface.
- Unisys is porting its tools to other vendors' systems and selling them through a software subsidiary called Foundation Systems Incorporated. Ally and Mapper were ported to Sun workstations in September 1990; other platforms will follow.

Premium Services. A Premium Service is a proprietary application service that complements de jure standards. The leading example is Unisys's Open/OLTP service, an unannounced product that is in customer field tests. Open/OLTP uses available standards—the X/Open XA transaction-oriented DBMS interface, AT&T's implementation of the X/Open OLTP API, and OSI networking—but adds proprietary synchronization and management services adapted from Unisys's XTPA mainframe OLTP product. (See Illustration 3.)

System Management. Unisys hopes to apply its experience on proprietary platforms such as XTPA to distributed systems management. The company is also a leading participant in the OSI/Network Management Forum and the Posix 1003.7 Systems Management Standards Committee.

Unisys has no systems management tools that work across its target environments today. It plans to offer fa-

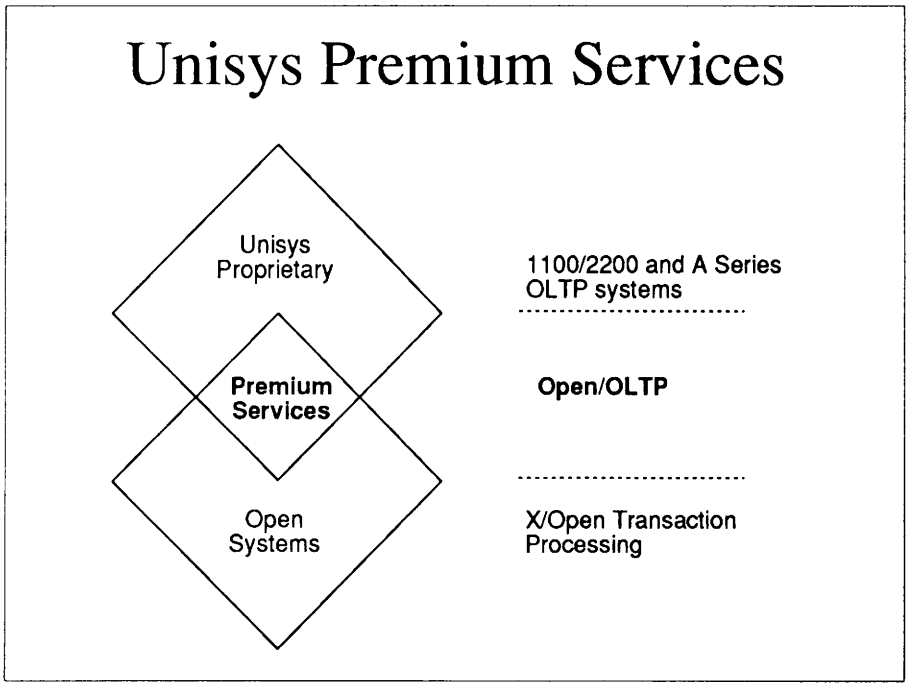


Illustration 3. Unisys hopes to attract new users by offering innovative Premium applications services. A Premium Service builds on an incomplete de jure *standard with a proprietary service or extensions. Unisys's Open/OLTP, an unannounced product, is the leading example of a Premium Service.*

cilities in the areas of configuration, fault management, and performance monitoring based on the OSI Common Management Information Standard (CMIS) in the near future.

IS THIS DISTRIBUTED COMPUTING? UA is built on a strong networking foundation, including OSI peer-to-peer communications services. By the first quarter of 1990, Unisys will have completed an OSI networking foundation across the 1100/2200, A Series, CTOS, and U Series. But does IIE define a distributed computing environment? Comparing the Open Software Foundation's Distributed Computing Environment (DCE) and Sun's Open Network Computing (ONC) frameworks to IIE, where's the standard RPC? Where's the distributed security service?

UA doesn't encompass all of the facilities defined in OSF's DCE and Sun's ONC. Unisys says it will in the future. The company is in the process of deciding how best to add RPC support to the Unisys Architecture. A major concern is selecting effective tools for users.

To us, UA appears to embody a conservative definition of distributed computing. IIE defines the systems in an enterprise as having three tiers—hubs (such as mainframes), departmental systems (such as Unix servers), and workstations (such as CTOS networks). Using Linc, a developer can split the functions of an application to run at different levels of this hierarchy in the client-server or peer-to-peer models. (Using Ally, a developer can split functions between clients and servers.) In addition, applications can be migrated to different processors on the network. (See Illustration 4.)

But, as UA exists today, the relationships between the subparts of a cleaved application must all be predefined. Without an RPC, there's no way for an application task to dynamically call on another task running at an arbitrary location for a service.

Unisys is likely to add the services needed to support a dynamic view of distributed computing during the next couple of years.

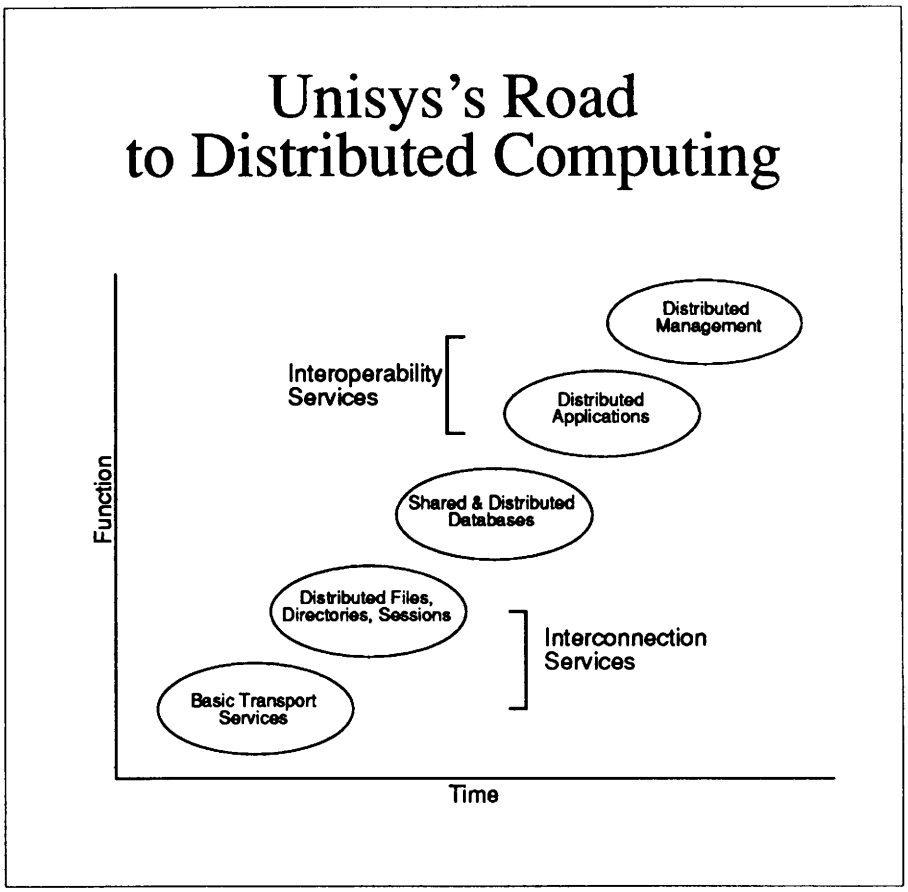


Illustration 4. Unisys appears to have created a conservative approach to distributed computing. The company hasn't yet incorporated a standard RPC, distributed security, and other advanced elements found in environments such as the OSF Distributed Computing Environment.

MANAGING IIE. Unisys has never had anything like IIE and UA. Just creating the architecture—and largely implementing it—is an accomplishment for a company that has heretofore been ruled by the parochial interests of its many product groups.

Unisys is managing IIE and its evolution through a "grass roots" management process. A small architecture group based in Roseville, Minnesota, works with the company's individual product groups to identify and codify consensus on standard elements across Unisys's platforms. This is how IIE came into being: It is the company consensus on available standards. The reason IIE says nothing about RPC, security, and other areas is that, in the architecture group's judgment, no consensus exists there.

Unisys's motto could be: "We shall sell no standard before its time." The company recognizes that building a consensus on each technical question will slow it down. However, the architecture group is empowered to make quick decisions on vital areas—such as the common management services and OLTP—and lay down the rules for development groups. This is the intent of IIE's Premium Services component. Also, Unisys appears willing to risk proposing standards in Unix-based OLTP and object-oriented data management.

Unisys also maintains an IIE Customer Technical Advisory Committee and a conformance and interoperability lab.

CONCLUSIONS. Everyone's got an en-

terprise systems architecture—is IIE anything special? That depends on your point of view. For users planning to move aggressively into distributed computing environments, such as the OSF DCE or Sun Microsystems' ONC, IIE probably isn't interesting. However, for users trying to leverage an installed base of mainframes, IIE makes a lot of sense. And for Unisys's existing customers, IIE is salvation itself.

For non-Unisys users, the big question is: Why would I want to add Unisys to my environment? If Unisys responds: "Because we've got IIE," it will fail. Unisys's best hope at this critical juncture in its history is to sell superior and open OLTP, superior and open software development tools, superior and open image processing, and so on. And so, in that sense, Unisys has just begun to fight. —*J. Rymer*