Patricia Seybold's

VOL. 3, NO. 9
ISSN: 0887-3054
SEPTEMBER 1988

UNIX IN THE OFFICE

PRODUCTS • TRENDS • ISSUES • ANALYSIS

Distributed Databases

Where Are We?

By Judith R. Davis

IMPLEMENTING A DISTRIBUTED DATABASE reminds us of the television ad that begins with a salesman on the phone, confidently talking to his boss. "Albuquerque on Monday? I can do that." (Slight pause.) "Des Moines on Tuesday? I can do that." (Another pause.) "Tampa on Wednesday? I can do that." He hangs up the phone, his bravado now deflated, and asks, "How am I going to do that?"

(continued on page 3)

• E D I T O R I A L •

Winning the Popularity Race

Will OS/2 or Unix emerge victorious? Or is there even a race to be won?

By Judith S. Hurwitz

THERE HAS BEEN a lot of talk these days about which operating system will win the popularity race: OS/2 or Unix. It is easy to fall into the trap of assuming that there will be a choice to make, but this is an erroneous assumption. It is not a question of which operating system will triumph. In fact, we firmly believe that, not only will both OS/2 and Unix survive any shake-up in the operating systems arena, but they will both continue to be key operating systems in the future. We also believe that OS/2 and Unix will form a tight relationship with one another.

AN OS/2-UNIX ALLIANCE. Why do we think this? Because OS/2 and Unix have many similarities. Both are multitasking, minicomputer-like operating systems, and both employ named pipes to move between processes. In addition, developers are likely to begin making logical connections that link these two operating systems.

FINDING A NICHE. The benefits of a close alliance between OS/2 and Unix will become clearer in the future. We predict that OS/2 will become a strategic operating system in the LAN arena. It will also become critical when the next generation of PC applications begins to emerge. These object-oriented applications will be graphical and will incorporate image, voice, and data.

At the same time, applications in the Unix marketplace will be moving in the same direction. We have already seen pioneers such as Applix and Plexus integrate different data types into compound files. This trend is sure to continue. OS/2 will find its niche in the low-end LAN marketplace, while Unix will range from the PC to the mainframe. (Yes, there will be overlap with Xenix.)

THE REAL COMPETITOR IS... Xenix, in fact, is the one area where there will be some real competition between OS/2 and Unix. Quietly, over the past five years, Xenix has had taken hold in many commercial organizations. For example, many fast food restaurant chains (MacDonald's and Wendy's, for example) have implemented point-of-sale systems based on Xenix. Other small retail outlets have found inexpensive Xenix systems with specially designed applications packages to fit their needs. Therefore, it is not unusual to find organizations using Unix without even knowing what was under the covers.

In fact, we expect that many software developers in this low end of the marketplace will see the value of porting their applications software to both Xenix and OS/2. They will do this as a way of capturing both markets and as insurance—just in case OS/2 doesn't make it. Since we believe that OS/2 will make it, we expect that these two markets will therefore be complementary.

PLATFORM FOR DNC. Unix will continue to be the platform for distributed network computing as more and more applications (such as the distributed databases discussed in this month's feature) are deployed across platforms of various sizes and shapes. At the low end, OS/2 (as well as Xenix) is well suited to the distributed network computing environment. It is, thus, inevitable that these two operating systems will be joined.

Soon, the debates about winning or losing the operating systems war will disappear. The real winners will be the end users who will increasingly find applications being developed at both ends of the spectrum—sometimes overlapping, but definitely complementary.

Patricia Seybold's Office Computing Group

Publisher PATRICIA B. SEYBOLD
Editor-in-Chief JUDITH S. HURWITZ

Managing Editor RONNI T. MARSHAK
Senior Editors MICHAEL D. MILLIKIN, JUDITH R. DAVIS
Telephone: (617) 861-3926
Associate Editor JOHN N. KELLY
Assistant Editor DAVID S. MARSHAK

Contributing Editor GARY J. NUTT
Telephone: (303) 444-0341
Staff Writer LAURE BROWN
Sales Director RICHARD ALLSBROOK JR.
Circulation Manager DEBORAH A. HAY

148 State Street, Suite 612, Boston, Massachusetts 02109 Telephone: (617) 742-5200 FAX: 1-617-742-1028

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. Application to mail at second-class postage rates is pending 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.

•DISTRIBUTED DATABASES•

(continued from page 1)

How many times have you sat down at your PC and wished you had easy access to information on another computer or in some foreign database management system (DBMS) that you haven't the foggiest idea how to access? It's no big deal if some of your data is in one dBase III database and some is in another on your own workstation. You can probably figure out how to get them together, since you designed both and all of the data is in a compatible format under one DBMS. The problem becomes much stickier if some of the data is in a minicomputer or mainframe database, or if it is spread throughout the network. Solutions can be found, but they usually involve much time and effort—manual input from hard copy, logon-file transfer links with incomplete conversion of dissimilar data types, or custom software.

Ingres on VAX/VMS? I can do that. Oracle on Pyramid under Unix? I can do that. IMS on the IBM mainframe? I can do that. DB2 on MVS? I can do that. Put them all together? How can I do that?

The ultimate dream is transparent access to whatever data you need across the system when you need it, in a form that is useful without extensive cleanup.

The ultimate dream is transparent access to whatever data you need across the system when you need it, in a form that is useful without extensive cleanup. In this article, we examine the role of distributed databases in making the dream of integration a reality, and the challenges that must be met by both vendors and users.

The concept of the distributed database has been researched for a number of years. The convergence of several developments is finally beginning to make decentralized computing—and, therefore, the distributed database—an economic and organizational reality. One major driving force is the proliferation of minicomputers and PCs, resulting in the increasingly lower cost of MIPS on the desktop.

The universality of PCs has also had a profound effect on the individual's perception of what should be possible with computers and networks. These "shoulds" are gradually being fulfilled. The recent emphasis on Distributed Network Computing (DNC) illustrates this. We are starting to believe that we will eventually be able to go anywhere and get anything on a network, regardless of where it is located or how it is stored.

What Is a Distributed Database?

In a distributed database, the database itself (that is, the data maintained by the DBMS) is split among two or more logical CPUs connected by a network. These CPUs are usually, but not always, geographically dispersed as well. The multiple sites in a distributed database network may be homogeneous or heterogeneous at one or more levels:

- The hardware (VAX, Sun, Pyramid, IBM, etc.)
- The operating system (VMS, Ultrix, Unix, MVS, etc.)
- The network protocol (DECnet, SNA, TCP/IP, etc.)
- The DBMS (Ingres, Oracle, Sybase, DB2, IMS, etc.)

TRANSPARENCY. The major objective of a distributed database is transparency for both the end user and the applications developer. The distributed database software aggregates the component parts of the database into a single entity that looks to the user exactly as if it were a single, nondistributed, local database. Ideally, the applications developer generates a distributed database application exactly as he or she would if the database were entirely local.

The key concept here is location transparency—neither the end user nor the applications developer has to know or specify where data is located in the network in order to access and manipulate it.

Another aspect of transparency is independence. An example is local site autonomy; that is, the person who owns the data at a particular site continues to control the operation of the local database, even though it participates in a distributed database network.

Several sets of "rules" for defining a distributed database have been documented in the literature. One well-known example is Chris Date's 12 rules for a distributed database. However, though these rules essentially describe the attributes of a true distributed database, they do not solve the problems inherent in actually implementing one.

WHY DISTRIBUTED? An organization can arrive at the need for a distributed database from two directions:

- Using the distributed database as a controlling, or distributing entity

- Using the distributed database as an integration tool

In the first case, the organization chooses to break a single database into component parts and distribute them across a network. An example of this would be splitting a centralized customer master file by sales region and placing the physical customer records for each region on that region's computer. Here, the environment is most likely to be a homogeneous one in that the same DBMS is used at all locations, and the central DBMS controls the overall design and operation of the system. Breaking up a database into distributed pieces is done for many reasons, including better response time, availability of data, recognition of political realities, and/or reflection of organizational structure.

In the second case, the organization has developed multiple, separate databases and now finds it desirable to treat them as a single database for the purpose of accessing related information. The distributed database is established to collect data from different sites and present them to the user as a unified, integrated database. For example, a car manufacturer has a table of parts, indicating which parts go into each car model, in the manufacturing plant. Information on the cost of each part is kept in a separate inventory database. Assessing the overall cost of a particular car model requires information from both databases.

This type of distribution most often involves heterogeneous hardware, operating systems, network protocols, and/or DBMSs. Here, it is important that the distributed database not interfere with local operations while it allows users to extract information across the database.

Several underlying forces push organizations toward one of these two situations. One is all of the inevitable vendor hype surrounding new or announced products in this arena. However, "real" initiatives come from the following needs:

Performance Improvements. Distributed database is one way to get parallel functions done concurrently. Storing parts of a very large database on different machines can allow the organization to perform queries and generate reports that may be unthinkable on a single machine. Queries can be separated into subqueries and funneled to different CPUs. This type of application also lends itself to the effective use of multiprocessor machines. An organization may also gain performance improvements by moving the data nearer to the people who access it.

The difference between distributed database and distributed processing lies in what is actually distributed—the data itself or access to the data.

Price/Performance Advantages. Closely matching hardware and software capabilities to application requirements can optimize price/performance. It can also result in the familiar multivendor environment. Distributed database helps here by hiding the differences between specialized systems while allowing the data to be combined.

Migration Path. A company may want to migrate to state-of-the-art DBMS and applications development tools while protecting its investment in existing applications and databases. A heterogeneous distributed database allows the organization to maintain current applications and databases without converting them, yet makes the data available to new applications and new tools.

Security. Some organizations want the ability to move critical, sensitive data to a secure machine when necessary.

Availability. The ability to have redundant copies of a database for "nonstop" processing in case of node failure is important for mission-critical, on-line applications.

Ease of Use. As networks extend further and begin to connect dissimilar computing environments, it becomes critical to provide ease of use and increased productivity for both end users and applications developers. If the distributed database software can hide all of the intricacies of different parts of the network, whether that be hardware, operating system, networking, or DBMSs, the user/developer will be much more productive. And, if the distributed DBMS can automatically handle concurrency control, integrity requirements, transaction management, and recovery across the network, so much the better.

WHERE THE PLAYERS STAND. We have included a chart summarizing the major features and functionality of the leading Unix vendors' offerings in distributed DBMS products: InterBase from Interbase Software Corp., Oracle from Oracle Corp., Ingres/Star from Relational Technology, Inc., and Sybase from Sybase, Inc. Please note that no one has everything today. Some, such as Relational Technology and Oracle, have implemented distributed query processing, but not distributed transaction processing. Sybase and InterBase provide both of these, but do not yet offer full location transparency.

Be cautious about evaluating distributed database on the basis of yes/no for features. Don't necessarily take no for an answer. Even though specific functionality may not be offered directly in a product, it can often be achieved with some effort on the part of the customer. The key question is the amount of effort required. And don't necessarily take yes for an answer. Differences also exist in the elegance with which functionality has been implemented.

RTI provides distributed database capability as add-on products to its Ingres DBMS. Oracle, Sybase, and Interbase offer distributed database capabilities as an integral part of their DBMSs (although networking is an add-on to Oracle). Other Unix DBMS vendors, like Informix Software, Progress Software, and Unify, do not yet offer distributed database capability. Progress has development plans underway, and Informix and Unify are committed to implementing distributed databases in the future.

DISTRIBUTED PROCESSING. We find some confusion (part of it intended, we suspect) between the concepts of distributed database and distributed processing. The difference lies in what is actually distributed. We have already explained that, in a distributed database, it is the data that is spread among multiple locations. In distributed processing, access to the data is distributed, but not necessarily the database itself. The database may still be centrally located.

Most DBMS vendors have implemented distributed processing by separating the DBMS into a "front-end" and a "back-end" component, each of which can run on a different system in the network. The front end consists of the user interface and the application. The back end is the database engine itself, which processes and accesses the data. The front end and back end communicate across the network (or within one machine in single-site situations) in performing operations on the database. The front end makes requests of the back end; the back end processes the request, returning data and/or messages to the front end. In some implementations, a single back end can support multiple front ends across the network. A DBMS that implements distributed processing is sometimes called a networked database.

Most DBMS vendors have implemented distributed processing by separating the DBMS into a "front-end" and a "back-end" component, each of which can run on a different system in the network.

CLIENT/SERVER. The client/server model is based on the concept of distributed processing, with the front end of the DBMS being the client, and the back end, the server. Both InterBase and Sybase stress their adherence to this model.

As defined by these two vendors, the client/server architecture is very specific about how and where to distribute functions: The application functions run on the workstation (client), and all data resides on the server. Furthermore, the business rules for data integrity reside with the data and are, therefore, on the server and not in the application. RDBMSs such as Oracle and Ingres do not include complete data integrity in the back-end module. Some or all integrity constraints are defined in the application itself, which is part of the front end. This can create potential problems in a large, distributed network.

The Issues

Several degrees of difficulty exist in implementing distributed databases. A basic starting point for any DBMS vendor is the ability to support a single-site, partitioned database. A partitioned database is spread among two or more disk drives. (Another term for partitioning is fragmentation.) Partitioning applies at two levels: the database level, where different tables are on different disk drives, and the table level, where parts of a single table are stored on different disk drives. A partitioned table can be split vertically (by column) or horizontally (by record). This is usually done for one of two reasons: The database/table is too big to fit on a single disk drive, or the user is looking for improved performance by getting around disk I/O limitations. Most DBMSs now offer partitioning at a single site.

The next step is to introduce a homogeneous network (e.g., Ingres running on several VAXs, all using VMS and connected by DECnet). Managing a database becomes instantly more complex when the data resides on multiple nodes connected by a network. The DBMS/user now must know the physical location of data and applications in order to access them. It is more difficult to manage functions such as concurrency control, query optimization and processing, and transaction processing. A new level of software is needed to handle global distributed database activities.

The next degree of difficulty occurs when layers underlying the DBMS become diverse, and different hardware, operating systems, and/or networks must be accommodated. The ultimate scenario introduces heterogeneous DBMSs. This is not only the most difficult to implement, but also the one that an organization, particularly a large one, is most likely to face. A further twist is the desire to connect data across different database models, relational with nonrelational.

DATA DICTIONARY. A data dictionary is a critical component of a distributed database, providing the location transparency for the user/developer in viewing data dispersed around the network. When the user asks to see what data is available, the appropriate information on tables and fields from each local and remote database must be somehow captured, consolidated, and presented to the user in a format consistent with what the user is accustomed to seeing.

At the same time, this "global" data dictionary must not interfere with each site's autonomy in defining and modifying its own database schema. Reflecting local schema changes in a global data dictionary is another issue. Is it done manually or automatically? Or is the information retrieved from the local site each time it is requested?

In today's world, Ingres uses a global data dictionary stored at a central site in the network (which may or may not be where the Ingres/Star distributed database software resides). The global dictionary contains the global table name, location of the table (node name), local database name, and local table name. The global data dictionary thus becomes a road map for everyone, providing location transparency for both users and developers. Maintaining a global data dictionary requires a global database administrator (DBA).

Currently, changes to the global dictionary must be made manually. Eventually, these changes will be automatic. Ingres's current dependence on a central site does mean that the system has a single site of failure. Once Ingres/Star includes replication capabilities (coming in the next version), the global dictionary can be updated automatically at multiple sites.

Oracle has chosen to implement a distributed data dictionary. The user has access only to the local Oracle data dictionary unless the local DBA or user establishes a link to a remote database. The user can then access remote data by including the link name in the SQL statement (e.g., "select * from employee @ HQ" where HQ is the link name). The DBA can take this one step further, creating synonyms in the local dictionary to specify the location of remote tables. This provides location transparency for the user by eliminating the need to specify a link name.

InterBase and Sybase do not use global dictionaries, although InterBase has facilities to provide location transparency. In InterBase, once a user has "attached" (opened) a database, he or she automatically gets an aggregate data dictionary for all attached databases. This process can easily be simplified by the DBA.

In Sybase, each site in the network has access only to the local data dictionary. The user/developer must know the logical node name where remote data resides in order to access it. If this logical node name changes, the appropriate programs have to be modified. Sybase plans to implement a level of location transparency in the future.

Designing an inefficient query on a single-site database is one thing. Doing the same thing over a network could leave the requester sitting there for days waiting for results.

NETWORK PERFORMANCE. A major issue in distributed database is effective and efficient use of the network. It is important to minimize the amount of data sent over the network, and to use connections cost effectively. Without high-performance networking, a distributed DBMS is not likely to provide acceptable application performance. Query optimization and the method used for packaging requests sent over the network can both be used to improve network performance.

Query Optimization. Like the data dictionary, query optimization is absolutely critical in a distributed database. The query optimizer in a relational DBMS (RDBMS) chooses the most efficient way to execute a particular query. To do this, it analyzes many variables, such as table statistics (e.g., number of records per table, the number of records likely to match the query criteria), indexes available to speed retrieval, and the number of disk I/Os required. This optimization is especially important when a query or update requires a join between two or more tables.

One of the major reasons for using a distributed database is to relate (join) data in different databases. If joins are required across multiple nodes, the optimizer also has to consider the amount of data to be shipped back and forth among sites, and then factor in the network overhead variables. These include cost, speed, response time, and what machines are on each node in terms of CPU cycles and I/O costs. Therefore, in a distributed database, the optimizer must be distributed as well. At a global level, the optimizer analyzes the query in terms of network variables and the table statistics that it has available, breaks the query up into subqueries, and sends these to the local nodes for processing. The local node further optimizes its subquery based on its latest database statistics.

Designing an inefficient query on a single-site database is one thing. Doing the same thing over a network could leave the requester sitting there for days waiting for results. The more intelligence the query optimizer has, the better it can assess optimal query execution. Some vendors are investigating the use of artificial intelligence in order to improve the optimizer.

A sample distributed query is one that selects 12 rows from one table on one machine, joins these with 1,000 rows on another machine, and returns only 12 rows. Let's say in this case, for performance reasons, it makes sense to select the 12 rows, move them to the machine with 1,000, perform the join there, and return the 12 results rows to the user.

Currently, Oracle, Sybase, and InterBase do no global query optimization. In the above example, all 1,012 rows would be sent over the network to the requesting node and the join would be performed there. Ingres uses its central Ingres/Star node for global query optimization, deciding where to move data and where parts of the query will be executed. Ingres would move only the 12 rows in the above query, performing the join on the machine with the 1,000-record table.

Request Packaging. Another factor that can impact network performance is how requests are packaged and sent from the front end to the back end. The least efficient way to do this is to send one database command (e.g., one SQL statement) at a time, and receive whole files in return. This requires two messages for each statement, and often means unnecessary data is transmitted back. Additional efficiency can be gained by sending a batch of several commands at once, by sending an entire procedure across the network to be processed on the server, and by ensuring that only the fields requested are returned to the application. InterBase does pre-processing on the client end to send a parsed (concise) query to the server and to get back only the data required.

Sybase uses what it calls stored procedures to improve network performance. These are compiled and optimized procedures stored on the database server rather than in an application program. In addition to minimizing the number of messages necessary to complete a database operation, the procedure is centralized, which means it can be shared and easily maintained. However, for this to work, the appropriate stored procedures must be available.

INTEGRITY OF DATA. Ensuring the integrity of data in a distributed database is a particularly sticky issue. One fact bearing witness to this is that only a few DBMS vendors, such as Sybase, Tandem, and Interbase, have implemented the ability to update data stored at multiple sites in a single transaction. Integrity of data becomes a factor at two levels: integrity constraints and transaction management.

Integrity Constraints. Integrity constraints include the integrity rules for records and data entered into the database. It is important to maintain integrity at both the table level and the field level. Referential integrity is the ability to define inter-table constraints within the database. Data entered, modified, or deleted in a table cannot circumvent these constraints. Examples of referential integrity: the department number entered in an employee record in the employee table has to exist in the department table; a customer record cannot be deleted if the customer has outstanding orders (i.e., if there are order records for the customer in the orders table).

Normally, there are also validation rules for data entered in specific fields. Examples include: the customer number must be unique; the sales commission rate must be between 5 and 8 percent.

In a distributed database, maintaining integrity becomes critical as more and more users and developers gain access to a particular database.

In a distributed database, maintaining integrity becomes critical as more and more users and developers gain access to a particular database. However, most RDBMSs require that the developer include integrity constraints at the application level. Although many products provide some level of data validation in the forms package, this still qualifies as building constraints into an application. Anyone with access to the database can design forms for the database. Therefore, these products depend on whoever is putting the application together, the developer or user, to ensure that all constraints are taken into consideration and properly and consistently implemented.

Triggers. Another approach to data integrity is to include the rules in the database itself. Both referential integrity and data validation rules are stored centrally in the data dictionary. These are generally called triggers or stored procedures.

Triggers provide two advantages, which apply whether the database is distributed or not. First, no application can be built that circumvents these rules. Second, the rules can be programmed once and stored centrally. This eliminates the need for each developer to include the rules in every application. Not only does this improve developer productivity, but it also makes maintaining the rules much easier. They only need to be created/changed in one location. Both InterBase and Sybase use triggers for data integrity. We think this approach is critical in a distributed environment and eminently sensible for databases in general.

Triggers move us in the direction of storing more intelligence at the database level, of maintaining within the database the integrity of operations, not just the data itself.

Transaction Management. The common approach to transaction management in a distributed environment is a two-phase commit (2PC) protocol. The node controlling the transaction instructs each remote node on the operation it is to perform. When each node is ready, it responds with a message to the originating node that it is able to commit the requested transaction. When all nodes have responded affirmatively, the originating node sends out messages to commit the transaction.

The distributed database software must include the ability to do this. A transaction consisting of updates to multiple nodes must be committed in its entirety or not at all. If the transaction is not successful at any node, it must be rolled back at every node to maintain database integrity and consistency.

InterBase provides a fully automatic 2PC protocol as part of the back end (server). Sybase has 2PC as well, but it is not an automatic function of the DBMS. The developer uses the 2PC facility to implement the protocol within the application. Ingres and Oracle do not yet provide 2PC.

SITE AUTONOMY. A distributed database must maintain local site autonomy. The local DBA retains full control over the local DBMS and its operations. Database-enforced integrity becomes a key to local site autonomy in a distributed database. A related requirement is that a distributed database should not have to rely on a central site in order to work.

REPLICATION. Storing copies of databases or tables on multiple nodes in the network is done for two reasons. One is to provide a high level of availability of data. If one or more nodes go down, the remaining copy or copies can be used to continue processing until the failed nodes are restored. Another reason for replication is improved performance. Moving copies of a database/table closer to the user can make a difference in response time. The obvious negative is that when updates are made to the primary database, the copies must be updated as well to maintain integrity.

Thus, the distributed DBMS software should provide built-in support for copy updates with the option to determine the interval at which updates are executed. In many cases, these copy updates can be deferred and do not have to be made at the same time as the primary update. No product does this automatically today, although both Sybase and InterBase provide the facility to update copies using the 2PC protocol.

CONCURRENCY CONTROL. Distributed databases generally leave local locking mechanisms and deadlock control up to the local DBMS. However, the distributed database software has to include some way of detecting and solving global deadlock situations.

ADMINISTRATION. In the distributed world, someone has to set up the initial distributed database and the global data dictionary, and then maintain it based on changes made to the system.

ORGANIZATIONAL ISSUES. Issues of control and independence cannot be underestimated in a distributed database environment. Current definitions of distributed databases stress the need to maintain local autonomy, that the local system manager or DBA maintain control over the integrity, security, and operations of the local database. However, there are those who believe that organizational struggles will make the implementation of truly distributed databases difficult, if not impossible.

It is important to realize that, to achieve a successful distributed database, it may be necessary to impose policies or procedures. What if two different databases use completely different formats for part numbers or customer numbers? Will people agree to give up control over the way they define information in the database? Do you force one group to change its codes, or do you build a mapping table? What if implementing a distributed database means that one group has to convert its data and applications to a different DBMS because a gateway doesn't exist yet?

One vendor representative stated that, in his experiences with designing and implementing distributed databases, organizational issues become more critical if no policies are in place. Problems can arise if organizational units that have not had to communicate in the past now must cooperate in the implementation of a distributed database. Don't expect a distributed database to solve the problems caused by lack of policies and procedures.

OTHER ISSUES. Many other issues arise in successfully implementing distributed databases. Because of all the complexities of distributed database and the fact that no one product provides all the necessary capabilities, the developer must begin applications design at a higher level. Applications development and deployment is much more difficult, and overall management, including source code control across multiple machines, becomes critical.

It is important to realize that, to achieve a successful distributed database, it may be necessary to impose policies or procedures.

Another major concern is transparency of tools across the network, such as the forms design capability, report writer, applications generator, and fourth generation language (4GL) for custom applications. This presents a flexibility problem. If all of the desired tools are not available on all platforms, you may have to limit the base of tools you are willing to endorse.

Organizations want applications designed for the distributed database to also provide transparency at the presentation services level. Ideally, the developer designs one application that can run on a terminal, a PC, a Macintosh, or a Sun workstation and take full advantage of each workstation's graphics and user interface capabilities. This piece of the puzzle is one that several DBMS vendors are attempting to put in place.

THE OPERATING SYSTEM. Two potential trends in the industry bear watching. One is the concept of implementing distributed functionality at the operating system level rather than at the application (DBMS) level. Some think it makes sense to evolve an operating system that provides generic functionality, such as a lock manager and a transaction recovery mechanism, that can be used by the DBMS as well as by other applications software. The other trend is to include database functionality in the operating system (a la IBM's new AS400).

Heterogeneous Distributed Databases

The need for heterogeneous interoperability among DBMSs is increasing rapidly. The key, again, is transparency, which can become exceedingly difficult when a DBMS must access the data in a foreign system, especially if the two systems are not both relational.

Making a foreign database look familiar to the user requires mapping the data structures, translating data access statements (SQL, DL/1, etc.), and converting data types where necessary. Specific data types are defined differently by some systems. VAX floating point numbers are different than those defined by IEEE. In some cases, a translation between ASCII and EBCDIC data, as well as between two different collating sequences, is required.

In spite of the RDBMS vendors' claims of compatibility with ANSI-standard SQL, there are, in fact, different dialects of SQL. Not all SQLs are the same, and each vendor has created its own proprietary extensions to SQL. What if the capabilities of one database exceed those of another? An example here is the fact that DB2 allows 300 columns per table, Oracle allows 256, and Ingres allows 127. What do you do when one database supports transactions and another doesn't? These differences have to be accounted for in the distributed database software or successful integration won't happen.

GATEWAYS. Some DBMS vendors are developing gateways to foreign databases. Relational Technology, for example, has already introduced gateways to Digital's RMS and Ashton-Tate's dBase III, allowing Ingres users to access RMS and dBase III data. Gateways for IBM's DB2 and IMS are coming. InterBase, which is compatible with Digital's Digital Standard

Relational Interface (DSRI), can access Rdb, RMS, and Unix file systems.

Oracle states that gateways from Oracle to DB2 and SQL/DS (the first offerings under the company's long-awaited SQL*Connect product) are in beta test.

The problem with the gateway approach is that the user must now rely on the DBMS vendor to provide access to all of the required DBMSs. While many DBMS vendors have announced their intentions to support the "biggies" like IMS, DB2, and RMS, it is entirely possible that this will not be sufficient. If one or more of the customer's existing DBMSs do not present the vendor with "interesting numbers" across the market as a whole, the gateway may never see the light of day. Gateways from a relational database to a nonrelational may, in some cases, remain read-only, since updates will be difficult at best.

A gaping hole so far is the lack of desire on the part of Oracle, Relational Technology, Sybase, etc. to develop gateways to each other.

And, in case you hadn't noticed, a gaping hole in these announcements so far is the lack of desire on the part of Oracle, Relational Technology, Sybase, etc. to develop gateways to each other! No one is talking about that yet; there's too much fierce competition for a "closed" solution to the problem. So gateways present some real problems.

Toolkits. One potential solution to the problem of gateways is for the vendor to offer a toolkit to customers and third-party vendors to develop their own gateways to foreign databases. Sybase indicated that it is developing such a toolkit.

INTERNATIONAL STANDARDS. Another approach to distributed database is to establish formal international standards for the interaction between different DBMSs, a set of protocols with which dissimilar DBMSs can exchange queries and data. Here, we need standards at three levels at least: the structure and format of the messages to be exchanged between dissimilar DBMSs (a transmission protocol standard), the contents of the message (a data access language standard, such as SQL), and a common data format (a data interchange standard). Each vendor would then provide an interface between its DBMS and the standard, allowing other DBMSs that also support the standard to access proprietary data.

To date, the international standards effort has been slow to reflect reality in the DBMS arena. The American National Standards Institute (ANSI) Technical Committee X3H2 on Database has already established a baseline standard for SQL Level 1. It is now actively working on two extensions to SQL, one for referential integrity and one for embedded SQL, as well as additional SQL statements (SQL Levels 2 and 3).

The committee has also set up a subcommittee to work on a Remote Data Access (RDA) standard, a protocol for accessing a physically remote database. RDA is essentially the envelope for a communication between a client (requester) and a server in the network. It also includes rules for the negotiation between client and server as to what the envelope contains. RDA does not assume that the statements passed are SQL statements. It is designed at a generic level to accommodate heterogeneous environments. However, the RDA task force is also working on defining the encoding of SQL statements. The current SQL standard defines the syntax for SQL, but not the encoding rules. RDA is now in the draft proposal stage, and will probably not become an international standard until late 1990.

One of the limitations of RDA is that it covers only the communication between a single client and a single server. It does not include several specifications that would be required for establishing a standard for a distributed database, such as data dictionary structure, a 2PC protocol, etc.

The bottom line, according to the vendors, is that, without standards, each vendor will develop only certain specific solutions to provide heterogeneous support. Until we get adequate standards, and until the vendors all agree to adopt the standards, we will not have true heterogeneous interoperability among DBMSs. Then the problem becomes, who will be first? Who will give up a proprietary solution for an open one? As one major vendor stated, "What's our incentive to support standards until someone else does?"

AD HOC STANDARDS. Another viable source for standards is industry collaboration. Sybase has begun such an effort with some of its partners, including some DBMS vendors (although none are its direct competitors). The company believes that the international standards committees will not keep pace with industry development. InterBase conforms to Digital's DSRI, as do several vendors of fourth-generation languages.

Conclusions

We've progressed from centralized computing with mainframes and dumb terminals to adding departmental minis to adding millions of PCs at the individual workstation level. What's critical now is the effort to tie everything together. Distributed database is slowly becoming a reality—the technology is making it easier, and there are compelling reasons for implementing it. However, it is early in the evolution of "true" distributed databases. Most vendors claiming distributed database capability have a distributed query function with minimal location transparency. A few have the ability to update multiple sites in a single transaction with a two-phase commit protocol. So the technical solutions are not here yet. A great deal more effort is required to fully implement distributed database technology and make it robust enough. And *(continued on page 12)*

UNIX DISTRIBUTED DATABASE PRODUCTS

Vendor	INTERBASE SOFTWARE CORP.	ORACLE CORP.
Distributed database product	Included with InterBase RDBMS	Included with Oracle RDBMS
Current platforms	Apollo (Aegis and Domain), DEC VAX (VMS and Ultrix), Sun (OS)	All platforms currently supporting Oracle RDBMS, including DEC VAX (VMS and Ultrix), IBM MVS and VM/CMS, DOS, OS/2, plus many Unix and proprietary platforms
Network support	The Interbase RDBMS includes support for DECnet, TCP/IP, and Apollo Domain.	SQL*Net, an add-on to Oracle, provides support for DECnet, TCP/IP, MAP, Xodiac, asynch, SNA VTAM (LU0), Named Pipes for OS/2, Novell. Coming is support for MAP 2.1, SNA APPC (LU6.2), TLI, Named Pipes for MS-DOS, NETBIOS, 3Com, Banyan.
Location transparency	No; a transparent environment for both retrieval and update can only be set up at the application level.	Yes, for retrieval by building locations (synonyms) into local dictionary
Ability to retrieve data from multiple sites in a single query (distributed query processing)	Yes	Yes
Distributed query optimizer	No	No
Ability to update multiple sites in a single transaction with two-phase commit protocol (distributed transaction processing)	Yes; automatic; part of the back-end DBMS	No
Database-enforced integrity (referential integrity and data validation in data dictionary)	Yes	Yes, via views defined with "with check" option; access to underlying database/tables must be restricted to the view.
Support for data replication	No	No
Access to heterogeneous databases DEC's Rdb and RMS Unix file systems (All are full read/write.)	DEC's Rdb and RMS Unix file systems (All are full read/write.)	None; gateways to IBM's DB2 and SQL/DS are in beta test.

Vendor	RELATIONAL TECHNOLOGY, INC.	SYBASE, INC.
Distributed database product	Ingres/Star, an add-on to Ingres RDBMS	Included with Sybase RDBMS
Current platforms	DEC VAX (VMS), Sun (OS) (While Ingres/Star runs on a limited number of platforms, it can manage Ingres databases on any platform supported by Ingres/Net, as well as supported heterogeneous DBMSs.)	DEC VAX (VMS), Pyramid (OSx), Stratus (VOS), Sun (OS)
Network support	Ingres/Net, an add-on to Ingres, provides support for DECnet, TCP/IP, LU0, SNA APPC (LU6.2), asynch.	The Sybase RDBMS includes support for DECnet and TCP/IP.
Location transparency	Yes for both retrieval and updates via global data dictionary	No
Ability to retrieve data from multiple sites in a single query (distributed query processing)	Yes	Yes
Distributed query optimizer	Yes	No
Ability to update multiple sites in a single transaction with two-phase commit protocol (distributed transaction processing)	No	Yes; not automatic; built into application on front-end
Database-enforced integrity (referential integrity and data validation in data dictionary)	Yes, for single-table validation via integrity constraints; no, for referential integrity	Yes
Support for data replication	No	No
Access to heterogeneous databases DEC's Rdb and RMS Unix file systems (All are full read/write.)	Gateway to DEC's RMS (read only); gateway to dBase III (full read/write); gateways to IBM's DB2, SQL/DS, and IMS are in beta test.	None

(continued from page 9) making distributed databases easy to develop and use is a key component of acceptance in the marketplace. It is much more difficult to write distributed applications than it is to write single-site applications.

We are still at least a couple years away from the mainstream distributed database with true location transparency and both distributed queries and transactions. Heterogeneous interoperability will come even later, with the key here being a set of industry standards.

In addition to the issues inherent in providing full distributed functionality within a DBMS, two large obstacles block the road. First, distributed database assumes a network. If all systems are not connected, then a distributed database cannot be implemented. This is still a real problem within many organizations, including some of the largest.

Second, heterogeneity, with all its complexity, is the real world, folks. All of these DBMS vendors are going to have to talk to each other sooner or later. In our opinion, the one that does it first will gain a competitive advantage—if, of course, the vendor can truly stand the competition by providing state-of-the-art DBMS capabilities, applications development tools, and networking performance. The only reason we can see for not building gateways to competitors' products is the fear of competition. And the alternative to building gateways is to take a proactive role in establishing standards for the industry. Then each vendor can stand on its own merits.

NEWS

PRODUCTS • TRENDS • ISSUES • ANALYSIS

ANALYSIS

• INSIDE •

•STRATUS•

Adding Pick and Unix

Stratus Computer, one of the leaders in the fault-tolerant computer marketplace, is on a roll these days. Revenue for its second quarter (ending July 3, 1988) was $62.9 million, an increase of 47 percent from $42.7 million for the same quarter last year. Even more impressive, its earnings were up 73 percent from the same period last year. What's the secret behind this Marlboro, Massachusetts, company's superb growth? First, Stratus was smart in locating a wide-open market niche in 1980—fault tolerance. This choice combined with strong technology thrust Stratus into the limelight when two key players, Olivetti (in 1982) and IBM (in 1985), established OEM marketing relationships. At the same time, Stratus has been careful to balance its channels of distribution by incorporating direct sales, distributors, and VARs.

Another interesting aspect of Stratus's strategy has been its venture into software markets. For example, in April, the company announced an SQL relational database called SQL/2000 based on Sybase, the hot relational database management system (RDBMS) vendor. The hardware base for this RDBMS is a workstation architecture called XA2000. The company has designed these workstations specifically for the distributed on-line relational database environment. The company has wisely put a lot of emphasis on networking these workstations via distributed as well as conventional Systems Network Architecture (SNA) networks.

But the company has not stopped there. It is paying close attention to the way the wind is blowing (at least the computer industry winds) and has made bids for both the Unix and Pick markets.

ENTERING UNIX. To capitalize on the growing interest in Unix, Stratus intends to develop a multiprocessor, fault-tolerant version of Unix for its XA2000 Continuous Processing System. In addition, the company has joined the Open Software Foundation (OSF). Stratus hopes that, by porting to Unix, it will be able to capture an even larger share of the online transaction processing (OLTP) marketplace, which, with the emergence of Digital as a player, has gained more visibility.

The Stratus version of Unix will be based on AT&T's System V.3 and will incorporate the kernel code developed by Olivetti. Stratus acquired the license to Olivetti's kernel last January. Stratus intends that its product will conform to Posix as well as System V Interface Definition (SVID). (Ah, standards.)

Other characteristics of the new operating system will include:

- Online maintenance, which is intended to allow users to add and replace system components while the systems are on-line
- Automatic system diagnostics such as fault detection, isolation, and reporting
- On-line remote services (provided by Stratus's Remote Service network), which are also the types of services that IBM is providing on its recently announced AS/400 systems
- Fast power failure recovery
- Disk mirroring

ENTERING PICK. While Unix will help Stratus with the standards industry (Stratus has its eyes on some lucrative government contracts), Pick is also an important marketplace. Stratus is quick to point out that market researcher InfoCorp estimates that the value of the Pick market was $1.3 billion for 1987, and it is expected to reach a compound annual growth rate of 19.2 percent between 1987 and 1992 with a dollar value of almost $3.2 billion.

What makes Pick attractive? Applications. While Unix vendors are just beginning to *(continued on page 16)*

The Sixth Annual Seybold Executive Forum:

TEAMWORK, TECHNOLOGY, & ORGANIZATIONAL PERFORMANCE

October

25 – 27, 1988

Royal Sonesta Hotel

Cambridge, MA

This year's topic is *"Teamwork, Technology & Organizational Performance."* Join us as we explore these issues with the computer industry's leading executives. Hear first-hand what their visions are for future technology directions. Hear their response to the challenges today's business and government organizations are facing. Learn why they are banding together with their competitors in often-surprising alliances to address customer needs. And interact with them in a relaxed, informal dialogue—no canned speeches, just straight talk.

DAY 1: TUESDAY, OCTOBER 25, 1988

In Search of a New Paradigm
The Office Computing Group
Michael D. Millikin, Vice President, Chief Technologist

The Dawning of a New Age in Computing
Sun Microsystems
William Joy, Vice President, R&D

Digital Equipment Corporation
William Strecker, Vice President, Architecture & Product Strategy

The "New" Organization Through Information Technologies
International Center for Information Technologies
Dr. Peter G. W. Keen, Executive Director

Cooperative Computing for Competitive Advantage
Hewlett-Packard
John Young, President

JFK School of Government
Robert Reich, Economist

Measuring the Productivity Payoff
Computer Aided Manufacturing-Int'l.
Thomas E. Pryor, Program Manager

Chevron Chemical Co.
Don W. Matto, MIS Manager

DAY 2: WEDNESDAY, OCTOBER 26, 1988

Restructuring American Business
IBM
George Conrades, General Manager, IBM-U.S., Marketing & Services

Managing for Multiple Constituencies: Stakeholder Management
NCR
Charles E. Exley, Jr., Chairman, CEO

The Payoffs from Computer-Supported Cooperative Work (CSCW)
Institute for the Future
Robert Johansen, Director, New Technologies Program

Groupware User Panel

Approaches to CSCW
Coordination Technologies
Anatol Holt, Chief Technical Officer

Lotus Development Corporation
James P. Manzi, President, CEO

MIT—Sloan School of Management
Thomas W. Malone, Professor of Information Technology & Management

Moving Toward "Real-Time" Businesses
Stan Davis, Author, *Future Perfect*

Managing Change
Index Systems
Tom Gerrity, Chairman, CEO

Xerox Corporation
Paul A. Allaire, President

DAY 3: THURSDAY, OCTOBER 27, 1988

Technology: The Trojan Horse for Organizational Change
The Office Computing Group
Patricia B. Seybold, President

On the Road to "Real-Time" Systems: Monitoring the Business Environment
Salomon Brothers Inc
Edward Boyhan, Vice President & Manager of Advanced Technology Group

Information Systems as Change Agent
Nolan-Norton
David Norton, Principal

Partnering for Progress
McKinsey & Company
Ian Somerville, Partner

User Panel

Planning for the Future Today
ON Technology
Mitch Kapor, Chairman and CEO

Concurrent Sessions:

Technical Sessions:

Object Oriented Environments
The Office Computing Group—Michael Millikin, Vice President, Chief Technologist

Data General—Chris Stone, Manager, Office Systems Software

Metaphor—Dave Liddle, Chairman

Wang—Jean Friedman, Department Manager, Wang Core Applications

Hewlett-Packard—William Crow, NewWave R&D Manager, Personal Software Division

Moving Toward EDI
Soft•Switch—Michael Zisman, Chairman

The Role of Standards
The Office Computing Group—Judith Hurwitz, Senior Consultant

The National Bureau of Standards—James H. Burrows, Director, Institute for Computer Science & Technology

X/Open—Robert Ackerman, Chief Marketing Officer

Open Software Foundation—Alex Morrow, Director of Strategic Relations

Leveraging ISDN
Northeast Consulting Resources, Inc.—Jim Herman, Telecommunications Consultant

Teamwork & Technology Workshops:

Simulating Organizational Dynamics
Digital Equipment Corporation—Skip Walter

Coordinating Actions
Action Technologies—Russ Redenbaugh

Inventing The Future
Northeast Consulting—David Mason

Building Consensus
Wilson Learning—Kalen Hamman

The Natural Planning Model
Insight Consulting—David Allen

CALL TODAY!

(617)742-5200

SPACE IS LIMITED TO 300 PARTICIPANTS.

☑ PLEASE REGISTER ME!

Discounts are available for two or more attendees from the same company. For information, call Deborah Hay at (617) 742-5200.

Send registration form and payment to:

The Seybold Executive Forum
Patricia Seybold's
Office Computing Group
148 State Street, Suite 612
Boston, MA 02109
(617) 742-5200
Fax #: 1-617-742-1028

☐ My check is enclosed for $1095 (Please make check payable to Patricia Seybold's Office Computing Group)

☐ Charge to my:
MasterCard/VISA/American Express (circle one)

☐ My purchase order number: ____________________

Card Number ____________ Expiration Date ____________

Signature ____________

Name ____________ Title ____________ Company ____________

Address ____________ City, State, Zip ____________

Telephone Number ____________

Cancellation Policy:
Should a registrant be unable to attend the Forum, the Forum Office will refund the full registration if notified before September 27. Cancellations from Sept. 28 to Oct. 11 are subject to a $50 service charge. There will be no refunds as of Oct. 12. Substitutions may be made at any time.

(continued from page 13) build up a backlog of key applications, Pick has long been a rich source of application software. Pick is especially strong in financial institutions, hospitals, municipal governments, retail distributors, hotels, and telemarketers. Stratus hopes that the large base of Pick VARs will gravitate to its hardware.

CONCLUSION. It is evident that Stratus is a smart marketing company. Branching out to encompass two hot markets (Unix and Pick) should carry the company into the coming decade. We are also pleased to see Stratus demonstrating its long-term commitment by becoming an early joiner of OSF.

—*J. Hurwitz*

•MERGERS AND ACQUISITIONS•

Fueling the Booming Unix Market

Now that Unix is officially hot, vendors large and small are looking for ways to strengthen their offerings. Two such moves come from opposite poles of the marketplace: Industry giant Unisys swallows Convergent Technologies, while smaller players Massachusetts Computer Corporation (Masscomp) and Concurrent Computer Corporation agree to merge.

UNISYS AND CONVERGENT (POWER OF TWO?). Rumors have been circulating for months that Unisys was on the prowl for an acquisition. Speculation was that NCR or Wang were potential targets. Conventional wisdom had it that Unisys needed some boxes in the mid range. However, given Unisys's strong move to Unix, it is not surprising that the company chosen was Convergent Technologies. Convergent, once a hot OEM company with customers ranging from AT&T to Prime, has had tough financial problems over the last few years. Lately, it had been hoping to find its niche by targeting "business Unix," but it was a little too late for the faltering company. Convergent's problems stemmed from the fact that it never moved from its original position as a hot box company. As the competition in the OEM/VAR arena heated up, Convergent never undertook the changes necessary to compete.

In contrast, the Convergent acquisition was a smart move for Unisys, which already OEMs Convergent Technologies' systems. Unisys seems to be succeeding in just the area where Convergent failed: It has adapted its strategy to keep pace with a changing marketplace. For example, it has begun to implement a plan to tie its various proprietary and nonproprietary platforms together with consistent software and networking.

One company which may be hard hit by this merger is Computer Consoles, Incorporated (CCI). CCI's Power 6 line of Unix minis have also been OEM'd by Unisys. But now that Unisys will be putting money into its own pocket by using Convergent's hardware, where does that leave CCI? Further complicating the plot is the fact that Convergent has an OEM agreement to market CCI's Unix-based integrated office software on Convergent's hardware. OfficePower is a direct competitor to the Unisys Ensemble suite of office applications.

MASSCOMP PLUS CONCURRENT: BIGGER REAL-TIME. As Unix grows more commercial, characteristics like real-time support become important. Therefore, the merger of Masscomp and Concurrent almost seemed destined. Both companies have based their businesses on the real-time scientific and engineering market. The difference is that, while Masscomp concentrated on the low- to mid-range processor market, Concurrent concentrated on the high end. Both companies have been interested in expanding their bases from a concentration on scientific and engineering markets. Masscomp, for example, has its base in the measurement, control, and defense industries. Concurrent, on the other hand, has concentrated on parallel processing and data acquisition systems. The combined company intends to concentrate more resources on such potential markets as real-time simulation for applications like business modeling for the financial industry. The new joint venture will take the Concurrent name. It was the obvious choice, since Concurrent is valued at $400 million while Masscomp is worth about $80 million.

To gain market acceptance, Concurrent will move from its proprietary chip to a standard. Masscomp has long used the Motorola 68000 family. Rumor has it that a deal with MIPS Computer is pending. In addition, the new company intends to invest in user interface tools, real-time database engines, and visual- or object-oriented programming—all, of course, with a real-time orientation. The merger of two companies that have such an important and specialized niche could make a significant impact on the industry, especially as real-time moves into the commercial software world. The new Concurrent is a company worth watching.

WHAT DOES IT MEAN, ANYWAY? Are these two mergers/acquisitions flukes or a sign of the times? It is likely that, as Unix becomes more and more important in the marketplace, the competition will intensify. We expect that the frenzy to add new features will cause many vendors to seek partners to strengthen their offerings.

—*J. Hurwitz*

•DIGITAL•

Digital Introduces OSF-Compliant Ultrix

As the dust from the establishment of the Open Software Foundation (OSF) begins to settle, each member company will be quickly called upon to "put your products where your mouth is." Digital, in a series of Unix-related product announcements, has done just that, introducing an OSF/Posix-compliant Ultrix, while more tightly connecting Ultrix into its VMS and DECnet/Open Systems Interconnect (OSI) architecture.

PLAYING THE STANDARDS. Ultrix-32 Version 3 operating system is designed to be compliant with a range of key industry standards including:

- IEEE 1003.1 Portable Operating System for Computer Environments (Posix). The IEEE approved the Posix specification on August 22, one day before Digital's Ultrix announcement.

- OSF Application Environment Specification, Level 0, which covers aspects such as operating system (Posix and X/Open), windowing and systems (X-Window, PHIGS, etc.), connectivity (OSI and TCP/IP), languages (Fortran, Cobol, C, Ada, Common Lisp, etc.), and data management (SQL).

- X/Open Common Application Environment (CAE).

- National Bureau of Standards (NBS) FIPS.

- X-Window System Version 11.

- System V Interface Definition (SVID) (Release II, Volume I).

In addition to the standards compliance, Ultrix-32 Version 3 offers several major enhancements. Most interesting is the ability to use Network File System (NFS) protocols in a VAX-cluster environment. Known as Ultrix NFS Clusters, this function allows as many as 16 VAXs to interoperate in a distributed file system over a dual path, 70 Mb link using NFS protocols. In this environment, a shared database of larger than 75 gigabytes is possible.

Other enhancements include:

- Berkeley 4.3 BSD features, including Berkeley Internet Name Domain (BIND) server

- Support for VAX 6210/6220 and VAX 8810/8820

- Communication software under TCP/IP, NFS, IBM 2780/3780 Remote Job Entry (RJE) emulation, and a LAT/TELENET gateway

- NFS file-locking

- VAX C for Ultrix systems

New Workstation Software. Digital introduced Ultrix Worksystem Software (UWS) V2. UWS V2, which includes graphical subsystem software components for Digital workstations, is based on X-Window System Version 11. It features Digital's X User Interface (XUI), which is the first implementation of Digital's recently announced DECwindows program. DECwindows implementation will permit a common graphical user interface and common programming interface across all Digital platforms.

Digital hopes that its XUI specification will be adopted by OSF as its standard user interface component, extending this interface into the multivendor world. Digital is also working separately to license DECwindows to a number of vendors. (At Xhibition '88, the X-Window conference held in August at MIT, XUI was seen running on a number of Sun 3 workstations.)

The OSF user interface decision is due this fall. According to Digital, if XUI is not chosen, it will do everything to bring the interface into compliance with the ultimate OSF standard while supporting applications developed under XUI.

Key features of XUI include:

- XUI Toolkit

- User Interface Library (UIL Compiler)

- Programmers' Style Guide

- Window Manager

- Session Manager

- Support for monochrome, gray-scale, 4- and 8-plane color systems

- Industry-standard libraries

- Extension Libraries such as PEX (PHIGS Extensions for X), PostScript, and imaging tools

Digital is also bundling a number of applications within XUI including a Session Manager, Window Manager, User Executive, DECterm, and utilities such as a clock, calendar, calculator, E-Mail, and DECpaint.

Version 1 of XUI does not include the DECwindow User Interface Construction Set (DWUICS), Digital's interactive graphical user interface builder. When DWUICS (pronounced Dweeks) is available, developers and users will be able to generate and edit user interfaces without writing any code.

INTEGRATING ULTRIX. Digital is also making good on its pledge to closely tie Ultrix into VMS and DECnet/OSI by announcing three products—VMS/Ultrix Connection, DECnet-Ultrix V. 3, and Ultrix Mail Connection—that will make access to either or both systems much simpler.

VMS/Ultrix Connection is a VMS software product that provides VMS services to Unix clients by adding Transmission Control Protocol/Internet Protocol (TCP/IP) and NFS support to VMS. With this product, a VAX or VAXcluster can act as an NFS server to any NFS Unix workstation (including UWS V. 2). Files stored on the VAX or VAXcluster can be accessed by VMS users as well. Key features include:

- Transmission Control Protocol (TCP)
- Internet Protocol (IP)
- User Datagram Protocol (UDP)
- Address Resolution Protocol (ARP)
- File Transfer Protocol (FTP)

DECnet-Ultrix V3 enables Ultrix users to tie Ultrix systems to all operating systems supported by DECnet Phase III/IV. The Internet Gateway allows an Ultrix system to serve as a gateway between DECnet networks and TCP/IP-based networks, eventually providing the ability to migrate heterogeneous environments to OSI environments through DECnet/OSI.

Ultrix Mail Connection V.1 provides connection to Digital's MAILbus message transfer service. This will enable Ultrix Mail users to exchange messages with users of All-In-1, VMS Mail, all X.400 systems, IBM PROFS and DISOSS, and other Unix-based mail systems.

CAPTURING THE ENTERPRISE. Digital's Ultrix products are part of a large set of announcements aimed at "Integrating the Enterprise," the theme for DECworld '88, to be held this month in Cannes, France. At the same announcement, Digital introduced many new products intended to increase connectivity, both within the DECnet/OSI world and to the Systems Network Architecture (SNA) world. Earlier in the month, Digital announced products that would enable VAX-Macintosh compatibility.

The encouraging theme running through these announcements is that we will soon see some actual product (the Ultrix products will be demonstrated at DECworld and this fall) springing forth from the discussions, heated debates, and promises that we have followed so closely. —*D. Marshak*

•MACWORLD EXPO•

Has A/UX Become a Four-Letter Word?

While the Apple faithful used MacWorld to hyper-celebrate the first anniversary of HyperCard, evidence of Apple's commitment to A/UX was difficult to find. Apple's new product announcements made no mention of A/UX, while third parties did little better than reiterate their plans to port applications to the platform "when it is ready." Two companies (TOPS and Information Presentation Technologies) did demonstrate A/UX Mac II file servers for their networks, and, in the conference's most exciting announcement, Oracle demonstrated its database software distributed among Macs, Unix machines, PCs, VAXs, and mainframes.

A/UX Network Servers

Two companies that focus on integrating Macintosh into the Unix environment both announced networked implementations under A/UX. Information Presentation Technologies (IPT) announced the availability of its uShare networking software for A/UX. With the new implementation, uShare services—AFP file service, distributed print-spooling, Unix-compatible electronic mail, VT 100 terminal emulation, virtual hard disk, network search and launch, and voice annotation (see Vol. 2 No. 12)—will be available to Macs and PCs using a Mac II under A/UX as the file and communications server.

For IPT, A/UX does not hold a special place, other than giving the company the ability to deliver an all-Apple solution. Strategically, A/UX is but one of many Unix implementations upon which the company plans to build its services. IPT sees its eventual place as seamlessly integrating Macs, DOS, and OS/2 machines into the corporate-wide network, be it Unix, SNA, or DECnet.

TOPS, A Sun Microsystems Company, also demonstrated its software running with a Mac II A/UX server. The product, developed by StarNine, provides file and print services and integrates a native electronic mail (QuickMail) with Unix Mail.

TOPS's colorful founder, Nat Goldhaber, recently left the company to form his own venture capital business (though he retains his role as company spokesman). The new TOPS, under Vice President and General Manager Rich Shapiro, is seeking to define its identity, with some very strong, albeit somewhat schizophrenic, results.

Self-definition has become difficult for TOPS as the company itself is trying to serve two very different markets. Its bread and butter (TOPS has gone from shipping 2,000 units per month when it was acquired by Sun in early 1987 to shipping over 60,000 units last month) is a mass market aimed at small installations of 3 to 10 systems (both PCs and Macs). At the same time, as part of Sun's overall strategy, TOPS is planning on a future release which will enable Macs and PCs to fully participate in Network File System (NFS) as both clients and servers. In this implementation, the NFS Remote Procedure Call (RPC/XTR) will replace the upper level of TOPS. This goes far beyond any current PC or Mac implementation, as Sun's current PC/NFS is client-only.

Can TOPS succeed with both a low-end and high-end approach? Thus far, it seems to be possible.

Oracle

Oracle Corporation announced a complete Macintosh implementation of its relational database management system (RDBMS). Using a HyperCard front end, developed by Oracle and named Hyper*SQL, a Mac user has full access to a local or distributed Oracle database. Oracle for Macintosh consists of:

- Oracle DBMS
- Hyper*SQL
- SQL*Net
- Pro*C
- Oracle Call Interface
- Database utilities
- System Stackware
- Example Stackware

The Macintosh version of SQL*Net networking software and drivers enables communications between Oracle residing on the Macintosh system and Oracle and IBM databases on remote systems. Thus, database applications can be distributed across Macintoshes, PCs, Unix machines, VAXs, and IBM mainframes.

Perhaps even more exciting is the HyperCard-based user interface. Hyper*SQL not only provides the user an easy graphical alternative to Structured Query Language (SQL) commands, but it also supplies an easy-to-use interface for managing and administering any part of a distributed Oracle network.

Oracle for Macintosh should be available in November. It will be released in developer and networking versions at $199 and $999, respectively.

Oracle Corporation also announced that it has formed an alliance with Network Innovations (which was acquired by Apple last spring) to jointly develop and market a CL/1 facility in Oracle for Macintosh to process CL/1 language requests from other Macintosh applications. Pricing and availability will be announced in January 1989.

—D. Marshak

•WANG•

Native Unix Operating System for VS 5000

As Wang tries to compete with IBM, Digital, and other systems vendors, Unix is playing an increasingly important role in its strategy.

In July, Wang introduced the VS 5000—dubbed the "VS On a Chip," an entry-level system aimed directly at the IBM AS/400 and Digital's miniVAX. Priced from $8,800, the VS 5000 is, according to Wang, four times as powerful as the VS 5 and 32 percent less expensive.

With the VS 5000, Wang, for the first time, is offering Unix as a native-mode operating system. Up to now, Unix has been available as a co-resident operating system with the native VS operating system of high-end VSs. This allows these VS systems to run, under a virtual machine environment, both VS and Unix operating systems and applications concurrently on the same system.

The new release of VS IN/ix, an AT&T System V.2-compliant implementation for the VS, now operates in native or standalone mode on Wang's new VS 5000 systems (the VS operating system is also available on the VS 5000). Unix applications already developed or ported under the co-resident mode of operation can be run without modification under the native mode (and vice versa). Wang estimates that over 600 applications have already been ported to VS IN/ix.

Wang is targeting the VS 5000 IN/ix system at domestic and international governments, banking, manufacturing, and commercial businesses. The company is hoping that the improved price/performance and the wide range of available Unix applications will enable it to stop the erosion of its installed base of large customers while giving it an entry point into small business.

The VS 5000 IN/ix systems will be available in October. Pricing will range from $12,500 to $113,000.

—D. Marshak

Patricia Seybold's Computer Industry Reports

ORDER FORM

Please start my subscription to:

		U.S.A.	Canada	Foreign
☐ *Patricia Seybold's Office Computing Report*	12 issues per year	$385	$397	$409
☐ *Patricia Seybold's UNIX in the Office*	12 issues per year	$495	$507	$519
☐ *Patricia Seybold's Network Monitor*	12 issues per year	$595	$607	$619

Please send me a sample issue of:

☐ *Office Computing Report* ☐ *UNIX in the Office* ☐ *Network Monitor*

☐ **My check for $_________ is enclosed.** ☐ **Please bill me.** ☐ **Please charge my subscription to: Mastercard/Visa/American Express** (circle one)

Name: __________________ Title: ________________

Company Name: ________________ Dept.: ______________

Address: ________________________________

City, State, Zip code: ________________________

Country: ________________ Bus. Tel. No.: ______________

Card #: ________________

Exp. Date: ________________

Signature: ________________

Checks from Canada and elsewhere outside the United States should be made payable in U.S. dollars. You may transfer funds directly to our bank: Shawmut Bank of Boston, State Street Branch, Boston, MA 02109, into the account of Patricia Seybold's Office Computing Group, account number 093-118-7. Please be sure to identify the name of the subscriber and nature of the order if funds are transferred bank-to-bank.

IU-0988

Send to: Patricia Seybold's Office Computing Group: 148 State Street, Boston MA 02109; FAX: 1-617-742-1028; MCI Mail: PSOCG
To order by phone: call (617) 742-5200

Topics covered in Patricia Seybold's Computer Industry Reports in 1987/1988:

Back issues are available, call (617) 742-5200 for more information.

Office Computing Report

1987—Volume 10

#	Date	Title
7	July	End-User Computing—Tools for Visionary Users
8	Aug.	Will Wang Make it? Can Fred's Leadership Turn the Tide
9	Sep.	Voice Messaging Systems—User Perspectives
10	Oct.	IBM Office Software—Implementing A Cohesive Vision
11	Nov.	User Developed Applications—They Know What They Need
12	Dec.	Hewlett-Packard's New Wave

1988—Volume 11

#	Date	Title
1	Jan.	Office Computing: 1988
2	Feb.	Mac Attack
3	Mar.	Electronic Data Interchange—The New Fontier of Electronic Mail
4	Apr.	PC LANs Revisited
5	May	Remission or Recovery—Will NBI's New Direction Bring Renewed Life?
6	June	Service And Support *Second Feature:* LAN Support

UNIX in the Office

1987—Volume 2

#	Date	Title
7	July	Texas Instruments—Blending AI with Commercial Systems
8	Aug.	Banyan Systems—A Unix Solution to the PC LAN Conundrum
9	Sep.	AFCAC 251—Defining the Future Unix Office
10	Oct.	The Oracle Relational DBMS—The Leader of the Pack
11	Nov.	Expres Initiative—Is there an Office Use for University Communications
12	Dec.	Unix Roundup—Unix-Based Integrated Office Systems

1988—Volume 3

#	Date	Title
1	Jan.	Beyond Unix—Creating the Standard Operating Environment *Second Feature:* X/Open Champions Open Market
2	Feb.	Progress
3	Mar.	Pyramid Technology
4	Apr.	IBM's Unix
5	May	AI and Unix
6	June	SmartWare from Informix Software—The Quiet Contender

Network Monitor

1987—Volume 2

#	Date	Title
7	July	Apollo's New Thrust—Shaping Computing's New Generation
8	Aug.	Proteon Inc.—A Better Mousetrap Isn't Enough
9	Sept.	Digital's DecNet
10	Oct.	Data General—Better Late Than Never
11	Nov.	IBM and SNA—Meeting the Challenge?
12	Dec.	Micom Interlan Inc.—Learning From Past Mistakes

1988—Volume 3

#	Date	Title
1	Jan.	Taking Stock, Looking Ahead *Second Feature:* Mac to PC: Making the Connection
2	Feb.	Excelan—Riding the Crest with TCP/IP *Second Feature:* Update on Banyan
3	Mar.	3Com—Opening a New Window
4	Apr.	OS/2 Networking
5	May	Hewlett-Packard—Riding with the Wind
6	June	Exploring Distributed Network Computing