Patricia Seybold's

VOL. 5, NO. 8
ISSN: 0887-3054
AUGUST 1990

UNIX IN THE OFFICE

COMMERCIAL APPLICATIONS • TOOLS • TRENDS

INSIDE

Unix OLTP

Getting Ready for Commercial Prime Time

By John R. Rymer

ONLINE TRANSACTION PROCESSING (OLTP) and Unix have arrived at the same point at the same time. OLTP is changing. It is no longer confined to the high-capacity, high-speed, dedicated systems used by airlines, banks, and credit card companies. OLTP has spread into applications that involve smaller amounts of data and fewer users and transactions than the massive traditional OLTP systems.

Unix plays a prominent role in these new "OLTP Light" applications. The reason is low costs. *(continued on page 3)*

• E D I T O R I A L •

The User Challenge

Can Users Find Their Voice?

By Judith S. Hurwitz

A MOST SIGNIFICANT lack in the Unix community has been the users' voice. X/Open's XTRA process gave large users and independent software suppliers a sponsored forum for reporting their key requirements. But there has not been one group that allowed the commercial users to represent their own position. The creation of a user-driven group is critical to the future growth of commercial Unix.

We are, therefore, encouraged by the formation of the Houston 30. This organization hopes to create a lobbying force consisting of open systems users that can persuade vendors to come up with practical standards to meet the needs of commercial customers. These users take a similar approach to X/Open by looking to de jure and de facto standards as a platform from which to build solutions. They include OSI standards, X.400, and X.500 in their check list of requirements. While they would, hypothetically, like to see one standard operating system, they are not opposed to innovation being added to the base and then brought into the standards arena. However, unlike some existing organizations, they hope to begin a user/vendor dialogue and a cooperative process to make sure they accomplish their goals.

The participants in the Houston 30 are experienced MIS professionals who have learned from experience. They have seen what happens when users develop requirements in isolation—that is, without the participation of the vendor community. They point to the long and torturous development of the MAP/TOPS specification as an example of a user initiative that was created without input from vendors and that, therefore, took too long to emerge. They also have seen what happens when vendors try to second-guess user needs. Without cooperation, neither users nor vendors get what they want.

The formation of this organization was inevitable. And it's about time. A similar movement to embrace the adoption of open systems has been underway in Europe for the past three or four years. Because X/Open originated in Europe, its message about open systems penetrated the consciousness of European users sooner. In addition, the physical circumstances of the European community helped these users understand the benefits of an open environment.

Today, U.S. computer users are beginning to wake up to an increasingly competitive environment. U.S.-based businesses are being forced to compete with companies in a united Europe and a set of sophisticated Asian companies. Companies in the industrial sector, like Boeing, du Pont, General Motors, and Hughes Aircraft, understand this better than most since they have been hard hit by hungry and sharp competitors from around the globe. No wonder these companies are charter members of the Houston 30. We believe it is important that users from different areas of the industry join in this effort. The financial users, for example, need to have their voices heard, as do users from other service industries.

All of these users have a common requirement. In order to compete, software must be portable and systems must be able to interoperate. The message is simple, but it is not one that vendors have traditionally wanted to hear. Now they are hearing it, and they have gone overboard, adopting the term "open" as part of every product and every strategy. We hope that a group like the Houston 30 will force vendors to apply the term honestly and make it mean something once again.

The formation of the Houston 30 indicates the direction of the Unix market. If this group is successful, it could push Unix vendors into the commercial systems spotlight by teaching Unix professionals what commercial information systems users really need. If vendors commit to cooperating with groups like the Houston 30, we suspect that each will learn a lot from the other.

PATRICIA SEYBOLD'S UNIX IN THE OFFICE

Editor-in-Chief
Judith S. Hurwitz
MCI: 3538836@mci.com
Internet: hurwitz@dcm1lp.das.net

Analysts and Editors
LAURE BROWN
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
MELISSA HAGAN

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.

•OLTP•

(continued from page 1) As users scale OLTP downward, they find that Unix systems offer the right performance at the right price for their new applications. A systems integrator reports delivering a big claims processing system for 20 percent of the estimated cost of an IBM Customer Information Control System (CICS) equivalent. Another user, David Sherr, director of equity systems software architectures for Shearson Lehman in New York, sought bids for hardware to support an OLTP application with a $3 million budget. The winning bid was $1.7 million. "We have a Unix strategy," he says. "But even if we weren't disposed to Unix, you just can't argue with economies like that." Indeed, the low cost of Unix systems makes it possible to build OLTP applications that would be prohibitively expensive on proprietary systems.

Unix has also grown up to meet the challenges of OLTP applications. Unix and its file system were not designed to satisfy the special reliability and control requirements of OLTP applications. However, hardware vendors such as Sequent and Pyramid provide extensions that make Unix reliable enough for OLTP. Relational DBMS (RDBMS) vendors fix the file system problem by creating their own file systems, and provide transaction control mechanisms like transaction queue management, transaction logging, and rollback/recovery within their database management software. The result is a solid foundation for OLTP Light applications.

But Unix OLTP users want more—they want to run bigger OLTP applications on Unix, and they want software independence. Currently, most are dependent on their DBMSs. They have the freedom to change hardware vendors when they need to, but if they change DBMS vendors, they sacrifice their applications. The best hope for software independence—what David Sherr calls "the Holy Grail of open systems"—lies with X/Open Limited and the International Standards Organization (ISO). The two organizations are working in concert to create standard protocols and interfaces for Unix OLTP in a distributed environment. Unix vendors hope the new standards will allow them to compete with proprietary OLTP systems such as IBM's CICS/MVS/VSAM to build reservation systems as well as for OLTP Light applications.

The first products based on the X/Open and ISO work will start appearing this fall, beginning a period of superheated development in Unix OLTP. This report examines the state of the art in Unix OLTP and the chances that Unix will grow to become the predominant basis for transaction processing.

The New OLTP

The OLTP applications based on Unix today are different from reservation systems or bank networks. But they're no less vital to their users. Take, for example, the OLTP applications running at Tootsie Roll Incorporated outside of Chicago. Three or four order entry clerks (more are added to handle the Halloween rush) complete about 70 transactions per hour using an Oracle DBMS running on a Sequent processor. Tootsie Roll's main criterion was Oracle's reporting and analysis tools. "We used to have a big cart that delivered batches of thick IBM reports to us every day," says President Ellen Gordon. "But, to compete, we needed more immediate information." Now, when decision-makers need information, they either get it themselves or have a quick 4GL routine written by MIS.

Access to information its transaction information is what's most important to Tootsie Roll. The company doesn't require nonstop availability of the transaction system. Tootsie Roll won't lose millions of dollars a minute if its system goes down. This is a fundamental difference between applications like Tootsie Roll's and reservation or banking networks.

Are systems like Tootsie Roll's *OLTP*? Veterans of mainframe OLTP say these are just database—primarily decision support—applications. Users like Tootsie Roll, however, say OLTP isn't useful without effective decision support. These business users live by former Citicorp chairman Walter Wriston's insight into competing in an information economy: Information about money, said Wriston, is more valuable than the money itself. The same is true of goods and services. In embracing this view, users like Tootsie Roll are changing the way we view OLTP, and changing it in a way that favors Unix.

WHAT IS A TRANSACTION? OLTP starts with the transaction. A retailer sells a red dress to a customer. A bank customer withdraws $100 from a checking account using a teller machine. A furniture maker orders a load of oak from a supplier.

Each of these interactions is different, yet each has two things in common. First, each changes the state of the participating entity's information about itself. The retailer's revenues go up. The bank customer's checking-account balance goes down. An amount is deducted from the furniture manufacturer's budget account for lumber supplies. And so on.

Second, each transaction has two states: It is either in process or complete. A transaction is complete when all parties can keep their parts of the bargain. For example, the retailer won't sell the red dress if the customer doesn't have enough money to pay for it. In OLTP, completed transactions are said to have been "committed." If a transaction can't be completed, all parties must be able to cancel them and return to the conditions prevailing before the transaction began. The retailer, for example, won't record the dress sale as revenue. In OLTP, this process is called "aborting a transaction."

ONLINE TRANSACTIONS. OLTP systems don't change these fundamental characteristics of transactions. Rather, they seek to reduce the time needed to complete transactions and the time between the completion of a transaction and its reflection in an organization's data about its business—its inventory, revenues, costs, etc. The teller machine saves the bank customer from standing in line to cash a check. The sale of the red dress may be immediately posted against inventory and revenue files, giving the retailer immediate information from the field.

An OLTP system collects information about transactions and posts the changes to the organization's information dictated by updating a shared database or file. The nature of transactions imposes four basic requirements on all OLTP systems. The four, called the ACID properties, are: atomicity, consistency, isolation, and durability.

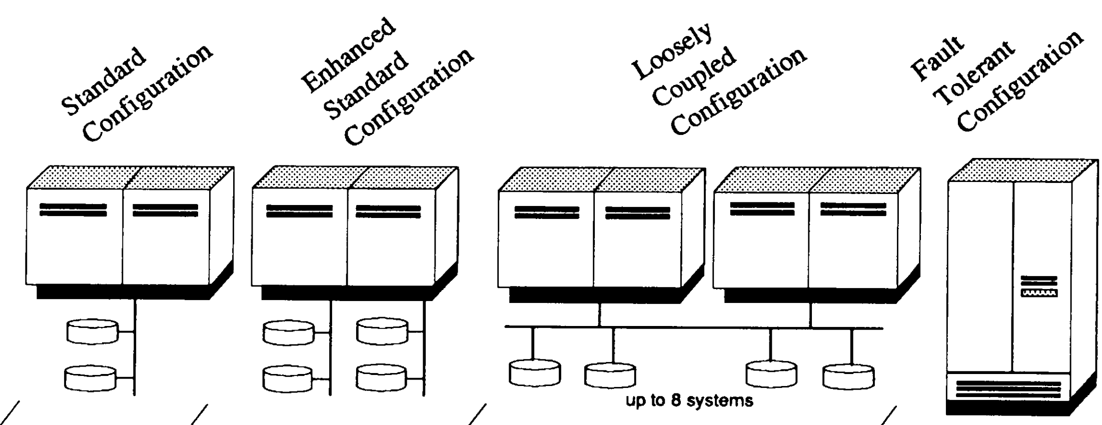
Atomicity. Atomicity recognizes that a transaction involves two or more discrete pieces of information. All pieces of information in a transaction must be committed, or none are. In the red-dress example, the sale of the dress involves, at minimum, a purchase price and an inventory adjustment. If the customer buys the dress, the purchase price is added to a revenue file, and a unit representing the dress is deducted from an inventory file. Both files must be changed to avoid inconsistent information. If one can't be changed, neither is.

Consistency. Consistency requires that a transaction either create a new and valid state of the organization's shared data or, if the transaction aborts, return the data to its previous state. If, at the last minute, the customer can't pay for the red dress, the retailer must be able to back out of the sale, returning the revenue and inventory files to their pre-sale conditions.

Isolation. While a transaction is in process, its details must be isolated from other transactions. Only when a transaction is committed can its effects on shared data be shown to other transactions. In the red-dress example, the customer may run down to the teller machine on the corner to withdraw enough money to pay for the dress. While she does, other transactions must proceed on the assumption that the dress hasn't been sold.

Durability. Durability means that an OLTP system saves a committed transaction's changes to shared data in the event of a subsequent failure. If the retail store is hit by a blackout a

The OLTP Spectrum



	Standard Configuration	Enhanced Standard Configuration	Loosely Coupled Configuration	Fault Tolerant Configuration
AVAILABILITY FEATURES		• Disk mirroring	• Disk mirroring • Automatic process recovery	• Hardware and software redundancy • Disk mirroring • Automatic process recovery
MEAN TIME TO RECOVERY	4-24 hours	Up to 4 hours	Up to 20 minutes	Up to 10 seconds
DATA RECOVERY	Last backup	Last committed transaction	Last committed transaction	No loss of data
PRICE MULTIPLE	n/a	up to 2X	up to 2X	up to 3X

Illustration 1. Online transaction processing means different things in different applications, as illustrated by this range of system configurations. Each configuration supports the base requirements for OLTP, with varying recovery-time standards. Unix OLTP systems are most commonly deployed in the first two categories.

minute after the customer buys the red dress, that sale must be reflected in the appropriate files when the system recovers.

The ACID requirements say nothing about OLTP system availability. A system's availability depends on the cost of lost time to conduct transactions. Reservation or telephone-switching systems require nonstop availability, called 24 x 7 (24 hours, 7 days a week) by OLTP specialists. The reason is that the cost of business lost during a reservation system outage justifies the cost of building systems that tolerate faults extremely well. For other businesses, however, short outages may be tolerable. Many users can afford lower guaranteed system availability. Users and vendors alike tend to view OLTP applications as fitting into a spectrum, as depicted in Illustration 1.

Unix: From Laggard to Leader

Five years ago, the phrase "Unix OLTP" was a contradiction in terms. There was no Unix OLTP to speak of. Today, Unix is hot as an OLTP technology. What has happened to thrust Unix into the limelight as an OLTP technology? There are several factors, both technical and nontechnical.

USERS WANT IT. The most important reason for the spread of Unix OLTP is the trend toward pushing OLTP applications closer and closer to the customer. For example, in a bank, that means installing teller machines that capture transactions directly from the customer's fingertips. In a retail store, it means capturing transactions at smart cash registers that are linked to servers running inventory and pricing databases. In the furniture manufacturer example, it means using electronic data interchange to swiftly process orders. The closer OLTP systems are to the customer, the quicker transactions can be processed, making a treasure trove of information about trends in buying, costs, consumer preferences—you name it—available to business decision-makers.

If a bank processes thousands of transactions a day and each transaction is worth an average of $100, it's not hard to justify big expenditures on OLTP systems like teller machines. But, as daily transaction rates and per-transaction values fall, traditional OLTP systems become too expensive. The new in-store systems planned by companies such as K-mart and JC Penney are small Unix servers, not big minis. Their system cost is more in line with smaller transaction volumes and values.

Low costs are part of the story. A growing number of users are moving to Unix as a strategy to adopt open systems. "We moved to Unix OLTP as part of a larger commitment to Unix for our business," says George Caneda, a systems development manager at B.E.A. Associates, a Wall Street money manage-

Open Struggles with RPCs

The big technical fight in Unix online transaction processing (OLTP) during the next year will be over the place of remote procedure calls (RPCs) in transaction processing. The issue is: What is the simplest application development method the industry can give developers? RPCs can shield developers from the details of networking, and proponents believe extended RPCs can do the same for the details of transaction control. The competing approaches in OLTP force programmers to learn transactional programming.

The arena for this debate will be X/Open, which will act as a clearinghouse for RPC standards work in the International Standards Organization (ISO). The X/Open model for distributed transaction processing does not include an RPC. It describes two communications paradigms. The first allows the binding of clients and servers to pass arguments or results. The second is the peer-to-peer, or conversational, paradigm described by ISO's Distributed Transaction Processing (DTP) Committee. ISO DTP is closely related to IBM's LU6.2.

The latest version of the X/Open model was released in July 1989. In the meantime, RPCs have become the hottest distributed computing programming method. An RPC (HP/Apollo's NCS 2.0) is at the core of the Open Software Foundation's Distributed Computing Environment (DCE). Sun is hard-selling a competing RPC from Netwise Incorporated as part of its Open Network Computing strategy.

RPCs don't understand what a transaction is unless they are supplemented with OLTP-specific semantics. The standard RPC is an asynchronous operation, which is incompatible with the requirement in OLTP to hold commitment of a transaction's parts if they all can be committed. Both Sun and HP/Apollo provide ways to perform synchronous RPCs, which should serve the need in OLTP. But there is no standard set of OLTP semantics for RPCs.

X/Open has assigned the task of figuring out the RPC's place in the XTP model to a subcommittee considering a new applications programming interface (API) to communications services in an open OLTP environment. In the current X/Open model, applications gain access to communications services through a transaction manager API. We expect X/Open to remove responsibility for managing communications from the transaction manager (TM) and assign it to a new (CM) Communications Manager component. A Communications Manager would be able to provide transaction-oriented communications services to the TM, shielding it from the communications method used to manage transactions. Thus, it would allow a general RPC to call on it for services without having to modify the RPC to understand what a transaction is. A separate CM interface would also allow an application to deal directly with communications services without going through the TM.

FT Unix Extensions

FEATURE	METHOD/EXTENSION
Unix hardening	Management processes limit size of system log files and ages of daemons.
File system	Support for both standard Unix file system and the BSD's Fast File System (FFS) which reduces I/Os and cpu cycles.
Disk partitioning	Allows disk to be partitioned between file system as well as raw disk space (for use by RDBMS).
Online support	Continuous monitoring and diagnosis subsystem supports self-checking logic and reconfigures components online. Also triggers processes that notify vendor service organization.
Disk mirroring	Maps every request by the file system for a disk write into two disk accesses.
Symmetric multiprocessing (SMP) support	Parallel execution and processing provided by granular locks on intrakernel data and interrupt handlers.
Online system	Processors can be added dynamically to SMP expansion configurations; I/O devices can be added dynamically to systems.
Power-fail ride-through recovery	Support for switch to battery power to ride out short brownout or to gracefully shut down in a blackout.
Priority scheduling	Addition of fixed-priority and real-time scheduling algorithms as well as Unix's time-shared scheduler.

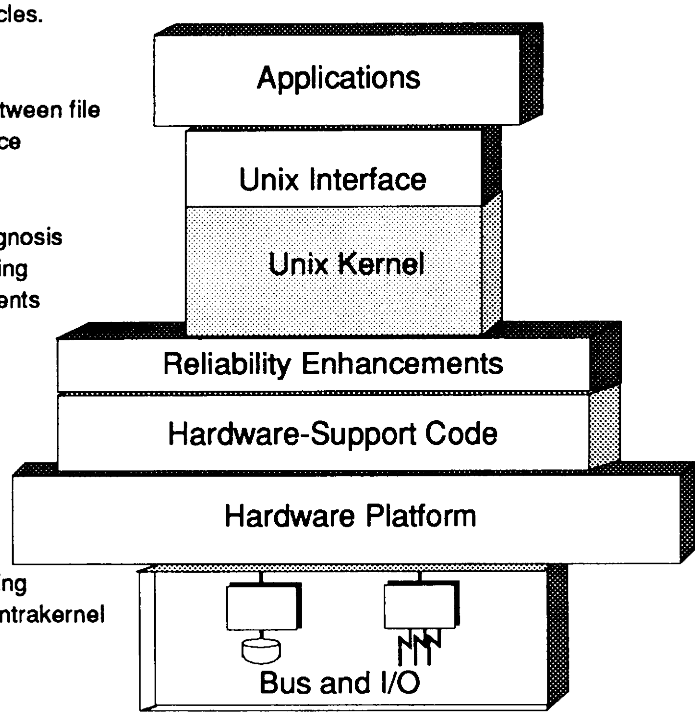


Illustration 2. The problem: How to extend Unix to support fault tolerance without compromising System V compatibility? Tandem, Stratus, and Sequoia have each taken these basic steps. The exception is support for symmetrical multiprocessing, which Tandem does not yet support on its new Integrity 2 systems.

ment firm with $9 billion under management. Telephone companies, which constitute a large market, require Unix System V compliance in their OLTP systems, as do a growing number of federal government agencies.

The phone companies, the federal government, and the number of corporate users committing to Unix add up to a substantial market. No one knows exactly how big it is, but, given that OLTP systems as a whole are believed to generate $50 billion a year in revenues, even a small share of the total is lucrative. High-end Unix performance specialists like Sequent and Pyramid are out to capture this market. Later this year, they'll be joined by Tandem and Stratus, the most successful vendors of proprietary fault-tolerant (24 x 7) systems. Unisys, Hewlett-Packard, IBM, and Digital won't be far behind.

UNIX KERNEL FIXES. A big part of the reason Unix OLTP has caught on is because vendors have fixed Unix's inherent weaknesses. The proof of their success is the small amount of attention Unix OLTP users pay to their operating system software. Most have few, if any, problems.

Vendors such as Sequent, Pyramid, and, more recently, Stratus and Tandem, have each had to solve the same basic problems in their Unix versions. They've done so without compromising the standard Unix interface. (See Illustration 2.) However, the Unix kernels provided by Sequent, Pyramid, and others in this field are no longer the standard System V kernel. Interestingly, many of the common fixes undertaken by these vendors will be incorporated into Unix System V Release 4, which will allow Unix OLTP vendors to bring their kernels into closer compliance with the standard kernel over time.

The fixes fall into four major areas: system robustness, data integrity, file system, and process management.

System Robustness. Unix has a reputation for being unreliable that is no longer deserved. It used to be that system errors, known as *panics*, could hang Unix. The default remedy: reboot, and often. Management processes that clean out system log files and kill off obsolete daemons eliminate the most troublesome panics. Unix System V Release 4 incorporates these fixes in a standard way.

Data Integrity. Standard Unix processes I/O by writing first to system buffers and then flushing the buffers to disk. Applications run using data in the buffers. This design raises a risk to data integrity. A power outage can blow away data in buffers before it is written to disk. The simple solution is to use Unix's synchronous write-through feature to bypass Unix's buffers and go directly to disk. This improves reliability, but at the sacrifice of performance.

A big part of the reason Unix OLTP has caught on is because some vendors—Sequent, Pyramid, Stratus, and Tandem—have fixed Unix's inherent weaknesses.

The popular approach is to let the DBMS take care of the problem. The DBMSs institute their own buffer schemes, and all have begun using parallel management of multiple queues to preserve data integrity without compromising performance.

File System. The standard Unix file system is tuned to access lots of small files—exactly the opposite of what's needed by OLTP applications. Most OLTP applications demand fast access to large files at random. Indexed file schemes like ISAM and IBM's VSAM are very successful in meeting this need.

The simple solution is to use a more efficient, tunable file system, such as the Fast File System (FFS) included in the Berkeley Unix, or to provide access to ISAM. The RDBMS vendors start from scratch, creating their own file systems using Unix's "raw I/O" feature. Raw I/O allows an application to take direct control of an I/O device, in this case, a disk. The raw I/O approach adds a proprietary element to an open system. Dharma Systems Incorporated (Hollis, New Hampshire) is the one vendor we've found that rejects the raw I/O approach. The reasons: Users have access to standard Unix system admin tools from within Dharma, and the Dharma system is more easily ported across Unix versions.

Memory Management. Unix's standard memory management is inappropriate to OLTP. Unix assigns one process to each user. The operating system must load new memory page tables, process control areas, and user registers—a procedure called a *context switch*—every time one user session ends and another begins. Context-switching eats up a lot of system overhead and memory, and can limit the number of users on a Unix OLTP system to a couple of hundred. Supporting more users requires memory management or a subsystem that can field user requests for services, store them in queues, and match them to available processes as they become available.

Unix OLTP vendors and users deal with this problem in a wide variety of ways, from writing a daemon to field and handle multiple user requests for individual server processes to writing their own transaction-control software—called a *transaction monitor*—to service multiple user tasks within single Unix processes.

24 x 7 AND SMP. Solving the fundamental limitations of off-the-tape Unix is only part of what's necessary to support OLTP applications. High availability OLTP applications require disk-mirroring and/or automatic process recovery. In disk-mirroring, every disk I/O is written to two disks, preserving a spare copy of the data. Automatic process recovery shifts processing to a hot standby machine in a "cluster" configuration.

The next level up on the OLTP Spectrum (see Illustration 1) is fault-tolerant systems, which duplex every system component, provide for online component replacement, and support sophisticated diagnostic and configuration management capabilities. Applications in this range are called "24 x 7" because they, theoretically, will never go down.

Lastly, there's a big push on among Unix OLTP vendors to support symmetrical multiprocessor (SMP) hardware designs. The Unix kernel and RDBMS software both need special hardware extensions to support these configurations. SMP promises to allow Unix systems to grow without swapping out processors. In the future, vendors will also boost performance by using multiprocessors to process instructions, queries, service requests, etc. in parallel streams.

Proprietary OLTP systems like Stratus's VOS and Tandem's Guardian don't have these difficulties. They were written to support OLTP, and they've been improved during years of experience with critical applications. The proprietary solutions cost more than Unix OLTP solutions. Vendors say the years of tuning and optimization that have gone into the proprietary systems are the reasons for their higher costs. Prices also reflect what the market will bear, given limited competition.

FOCUS ON TP MONITORS. The typical Unix OLTP system today comprises a Unix hot box, a version of the Unix kernel that's been extended to support redundant disks, multiprocessing, and other performance and/or availability features, and an RDBMS. The RDBMS usually provides support for the atomicity, consistency, isolation, and durability (the ACID properties) required by OLTP applications.

The central role of RDBMSs in Unix OLTP makes a lot of people uncomfortable. Users committed to open systems strategies are keenly aware that their dependence on an RDBMS limits the openness of their systems. RDBMS software locks in users with a variety of proprietary elements, from their custom file systems to transaction management features like queue management to performance features like stored procedures. If I use Sybase, for example, I write my applications in a proprietary version of SQL, Transact SQL, and use a private service applications programming interface (API). Even if other RDBMS vendors support the same functions Sybase does, they usually implement these functions in different ways. One way to increase the openness of Unix OLTP systems is to separate transaction management services from data management facilities, making transaction management services available through a separate API.

OLTP professionals trained in CICS don't like assigning transaction management to an RDBMS. They'd rather have a separate piece of software—a transaction monitor—to coordinate the execution of transactions. A transaction monitor is responsible for managing the processes, queues, and system states involved in processing and committing or aborting online transactions. Transaction monitors generally have the following components:

Name Service. The name service in a transaction monitor stores the names and addresses of data management and I/O services. When an application requests a service, the name service fields the request and delivers it appropriately.

Queue Manager. If the name service fields a request for a service that is currently occupied in another transaction, it places the request in a queue. The effect of queueing is to bypass Unix's requirement that each service request have a dedicated proccess. Transaction managers allow multiple applications to use a single process, and can dramatically raise the number of users supported on a system.

Scheduling Service. The scheduling service in a transaction monitor schedules processes to handle competing requests for transaction services.

Communications Service. The communications service allows the transaction monitor to deliver service requests to appropriate services.

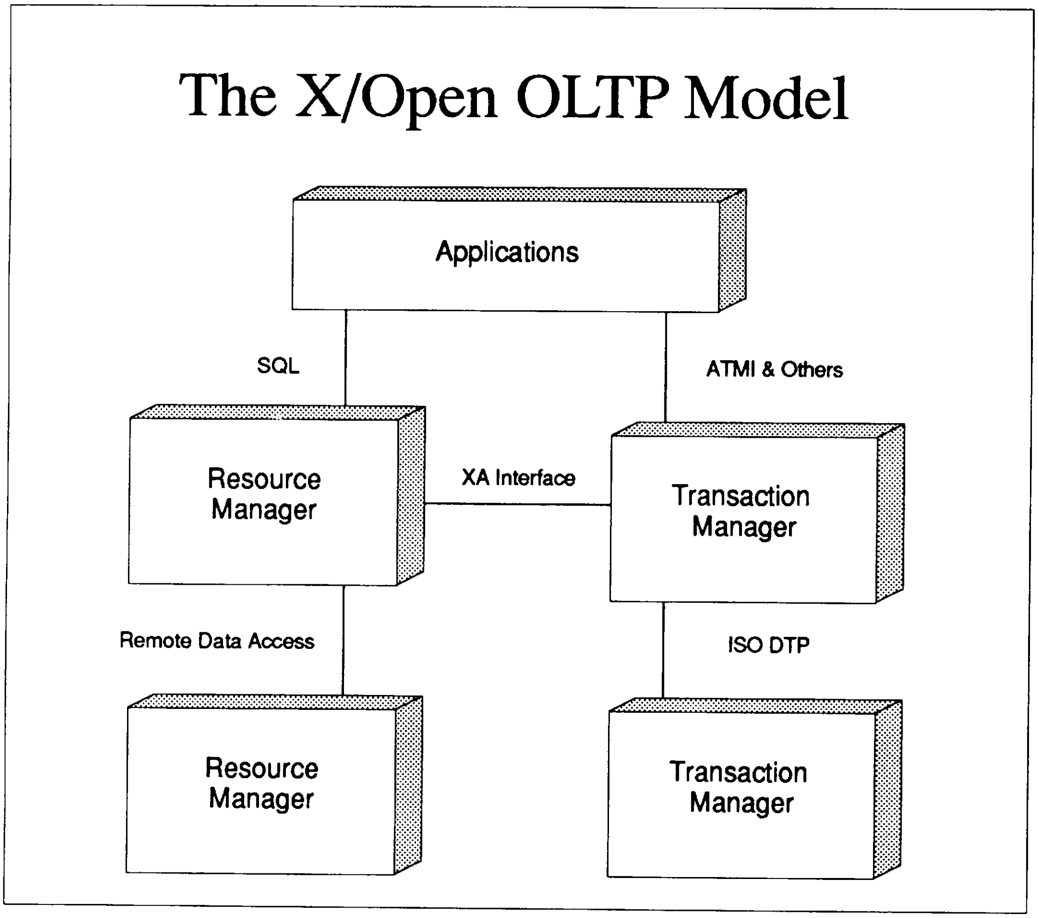


Illustration 3. The overriding concern in the debate over Unix OLTP standards is which interfaces can be defined today without stifling innovation that may deliver greater function later. X/Open's DTP Model identifies several key interfaces. The ISO RDA (OLTP context) and Distributed Transaction Processing (DTP) protocol standard proposal are other important parts in the picture.

The XA Interface

APPLICATION (AP)	TRANSACTION MGR (TM).	RESOURCE MGR. (RM)
tmopen0	Gather openstring, xa_open(openstring)	Prepare RM for use(connect to DBMS).
tmbegin(flags, timeout)	Generate gtrid, xa_start(gtrid, flags)	Associate gtrid with calling thread of control.
EXEC SQL UPDATE...WHERE...		Perform work on behalf of global transaction.
if (sqlca.sqlcode== SQLROLLBACK) tmabort(flags)	xa_end(gtrid, TMFAIL)	Dissociate thread from transaction and mark abort only.
	xa_abort(gtrid)	Rollback work done on behalf of transaction.
/* *Do work at other RMs *either local or remote */	Ensure that remote TMs know about gtrid.	
tmcommit(flags)	xa_end(gtrid, TMSUCCESS)	Dissociate thread from transaction.
	xa-precom(gtrid) Tell remote TMs to prepare to commit if(ALL_PRECOM_OK){ Stably record commit point;	Prepare to commit work performed on behalf of transaction.
	xa_commit(gtrid); Tell remote TMs to Commit; } else {	Commit work.
	xa_abort(gtrid) Tell remote TMs to abort; }	Rollback work.
tmclose()	Gather closestring, xa_close(closestring)	Tear down association with RM (disconnect from DBMS)

Illustration 4. This example shows how an application would define a global transaction using a transaction manager in conjunction with an RDBMS and hows how transaction manager primitives map to XA services.

Log Service. As it coordinates the execution of transactions, the transaction monitor stores information about the transaction's progress and the state of the systems involved in executing it in a log. The information must be current and detailed enough to roll back the transaction if necessary.

In our red-dress example, a transaction monitor would identify the steps needed to complete the transaction as a discrete unit of work. It would track the correct completion of data-gathering and the updating of shared data. It would store in a log reference information about the steps of the transaction and the state of the systems involved. The information in the transaction manager's log would then be used to roll back the transaction if aborting it became necessary.

Transaction monitors offer another important benefit to Unix OLTP: support for more users. Currently, most Unix OLTP systems top out at about 200 users. A transaction monitor, by scheduling and managing user requests for transaction services more efficiently than current RDBMSs do, could raise the user threshold of Unix to 1,000 or more. Independence Technologies Incorporated (Fremont, California), a systems integrator-turned-Unix OLTP vendor, used a version of AT&T's Tuxedo transaction monitor for Unix to build a health care claims system that supports 3,000 users.

In this case, the proprietary-system users and Unix users agree: Unix OLTP will benefit if the transaction management is separated from the RDBMS.

Unix has never had a standard transaction monitor—RDBMSs have filled this role, or users have written their own monitors. Users and vendors alike agree that transaction monitors are necessary to make Unix a serious contender in OLTP. But what's the best way to fill the need for better transaction monitor software in a standard—even better, an open—way? This is, after all, Unix OLTP. It should be open.

Like the promise of Posix itself, the X/Open OLTP model will take years to become a working reality. Nevertheless, it's crucial to the development of open OLTP systems.

THE X/OPEN MODEL. If Unix OLTP is going to be open, applications must be able to call on databases or transaction monitors through a set of common APIs. What's needed is a set of Posix-like interfaces for OLTP. X/Open Limited is defining a series of APIs to allow, for example, an OLTP application with components that execute on an Oracle RDBMS or an Ingres RDBMS using the same calls. A different API would allow an application to call transaction monitors from different vendors using common commands.

Like the promise of Posix itself, the X/Open OLTP model will take years to become a working reality. Nevertheless, it's crucial to the development of open OLTP systems.

X/Open is defining its APIs by breaking OLTP into three functional components—resource managers, transaction managers, and applications—and defining open APIs to each component. (See Illustration 3.)

Resource Manager. Resource managers include RDBMSs, file systems, and print services. A resource manager is responsible for managing a shared resource according to the ACID properties. Transaction managers and applications work with resource managers by asking them to perform services on their behalf. Applications and transaction managers don't have to know how the requested service is performed. As long as all elements support the same service interface, users should be able to plug in new components as they need to.

X/Open has circulated for industry comment a formal proposal for an interface—the XA interface—to resource managers. (See Illustration 4.)

Transaction Manager. A transaction manager provides the transaction management services that today's transaction monitors provide. It also gives applications access to communications services. Transaction managers use the XA interface to obtain the services of resource managers to complete a transaction. X/Open has reserved a place in its model for the emerging ISO Distributed Transaction Processing (DTP) protocol as the interface between two transaction monitors cooperating in a single transaction.

Application. Applications define transactions as having a beginning, a sequence of operations, and an end. The transaction manager coordinates and tracks the execution by resource managers of the operations defined in an application. They ask transaction managers to complete transactions on their behalf through a still-to-be-specified interface.

In our red-dress example, an application defines the operations that must occur during any sale to a customer, from the collection of data at the cash register to the updating of the inventory and revenue files on a server. A transaction manager directs the steps in the process, making sure that all of them are performed properly. A resource manager updates the inventory and revenue files at the direction of the transaction manager.

Distributed OLTP. The X/Open's model assumes its interfaces must allow different parts of a transaction to be executed on different, heterogeneous systems linked by a network. This is known as distributed transaction processing.

Distributed transaction processing is the future of OLTP. By allowing transactions to be executed across a network, users can provide additional capacity to existing applications without swapping in new systems, and can leverage older OLTP systems in new applications. For instance, the store in our red-dress example might want to add a dynamic pricing application to its systems that allows a regional office to monitor how merchandise is moving and to transmit price changes to stores to help stimulate demand. If the retailer's pricing database is on an IBM mainframe and its in-store systems are based on Unix, a distributed OLTP system might allow it to implement the new application without rewriting its pricing database.

The X/Open model incorporates two concepts to accomplish distributed transactions. First, it distinguishes between local transactions and global transactions. A local transaction is a set of operations that a resource manager executes as a local unit of work under its own control. A global transaction is a set of operations that is under the control of transaction managers and that includes local transactions.

Second, the X/Open model assumes that the transaction manager and resource managers in an OLTP system use a two-phase commit (2PC) protocol. A 2PC protocol builds an extra step into the process of committing a transaction. It allows the transaction manager to tell all involved systems to prepare to

Client-Server Two-phase Commit

TIME	CLIENT	ACTION	SERVER	LOCATION
Begin	tmbegin	Request	[TM awaits request]	Transaction Manager
	[prepare request]			
			[TM receives request]	
	tmacall		[TM dispatches service]	
	tmgetreply	Reply	Service0	Application
			[does requested work]	
	[waits]			
			tmreturn0	
End	[Determines whether to commit]	2PC	[TM returns to await the next request	Transaction Manager
	tmcommit			

Illustration 5. This is how two applications would interact using a two-phase commit (2PC) protocol under the client-server model. Bracketed statements describe events on the client and server sides of the connection. Nonbracketed statements indicate commands. TM, in the chart, refers to a transaction manager.

commit their parts of the transaction. If all of the involved systems respond to the "prepare to commit" message by saying they are ready to commit, the transaction manager then issues a Commit command. (See Illustration 5.)

The X/Open transaction processing model gives OLTP vendors an important target. X/Open has a long road to travel before its work on OLTP is complete. The X/Open OLTP group released an interim model a year ago. We expect it to introduce another interim version soon that will move communications management from the transaction manager to a new communications manager. (See "X/Open Struggles with RPCs page 5.") Beyond fine-tuning the model, X/Open faces other issues as it seeks to define open distributed OLTP. System management, for example, is a gaping hole. No standards organization or vendor has crafted a unified way to administer and manage distributed networks. And there are other issues X/Open hasn't addressed at all. For example, what is the standard way to identify a set of distributed operations as a single transaction? And how can sophisticated transaction structures, such as nested transactions, be performed in a standard way across distributed systems?

In the meantime, the Unix OLTP vendors are pushing to expand beyond their base in OLTP Light applications by offering transaction monitors and a variety of performance enhancements. IBM, Unisys, and systems vendors are building integrated OLTP environments for Unix. (See "The OLTP Heavies Weigh In," page 14.) All but two of the major vendors are proceeding with commitments to abide by the X/Open model and its interfaces. However, it remains to be seen how quickly X/Open's work will be implemented across a variety of products. The X/Open dissenters—VISystems Incorporated (Dallas), which provides a CICS-like environment under Unix, and Dharma Systems—believe the X/Open model is impractical because it doesn't address the need to optimize the components of an OLTP system.

What Unix Offers Now

Users building Unix OLTP solutions today have three basic choices: They can rely on the facilities of a general purpose RDBMS, they can implement one of several independent transaction monitors, or they can build atop an optimized OLTP operating environment that includes a database built for OLTP and a transaction monitor.

There has been much activity in each of these three areas as Unix OLTP vendors seek better performance and support for bigger systems than are feasible today.

FROM DSS TO OLTP. The RDBMS cut its teeth in information processing as a decision support engine. As we've seen, users like Tootsie Roll have pushed the RDBMS vendors to support OLTP in addition to easy data access. To keep pace with demands for bigger, faster Unix OLTP systems, Oracle, Ingres, Informix, and Sybase have each announced "OLTP releases" during the last year. Among these vendors, only Sybase was designed originally to support OLTP.

The OLTP releases of the major RDBMSs are either retuned or wholly redesigned to better support OLTP's requirements, which would allow their RDBMS engines to function as resource managers in heterogeneous OLTP environments under the control of transaction monitors. This report covers the highlights of these efforts.

Tuning the Engines. Each of the major RDBMS vendors has tuned its engine in essentially the same ways. They all allow databases to be locked by the row page during a transaction, rather than requiring entire tables to be locked. And they all employ the same techniques to reduce the amount of I/O required to service transactions.

Architectural Adjustments. Each of the major RDBMS vendors has also changed the architecture of its software to support more users and higher throughput in OLTP applications. There are two possible approaches. The first is to build a multi-threaded OLTP operating system on top of Unix. The second approach is to layer a transaction monitor between a conventional RDBMS and applications to give the appearance of multithreading.

To speed the proc-essing of transactions, RDBMSs can simultaneously process two or more transactions by using multiple server processes. This approach, called the multiclient-multiserver, or multithreaded, architecture, processes transactions in parallel, usually on multiprocessor hardware. An alternative to the multiclient-multiserver architecture is integrating a transaction monitor and a multiclient-single server RDBMS. In that configuration, the transaction monitor can achieve the same benefits as a multithreaded RDBMS engine.

Sybase and Ingres both have multiclient-multiserver architectures. Sybase added this feature with a virtual server architecture that allows an SQL Server process to two or more of the processors in a multiprocessing system. Ingres provides multiserver support with a multithreaded RDBMS kernel. Both of these database vendors provide their own transaction management software.

The Multiclient Server option of Oracle Release 6, due out this fall, and Informix Online, an OLTP version of Informix, both use the transaction monitor approach.

Oracle and Informix both plan to rely on their own transaction managers as well as on AT&T's Tuxedo transaction monitor technology. Both vendors are working with systems ven-

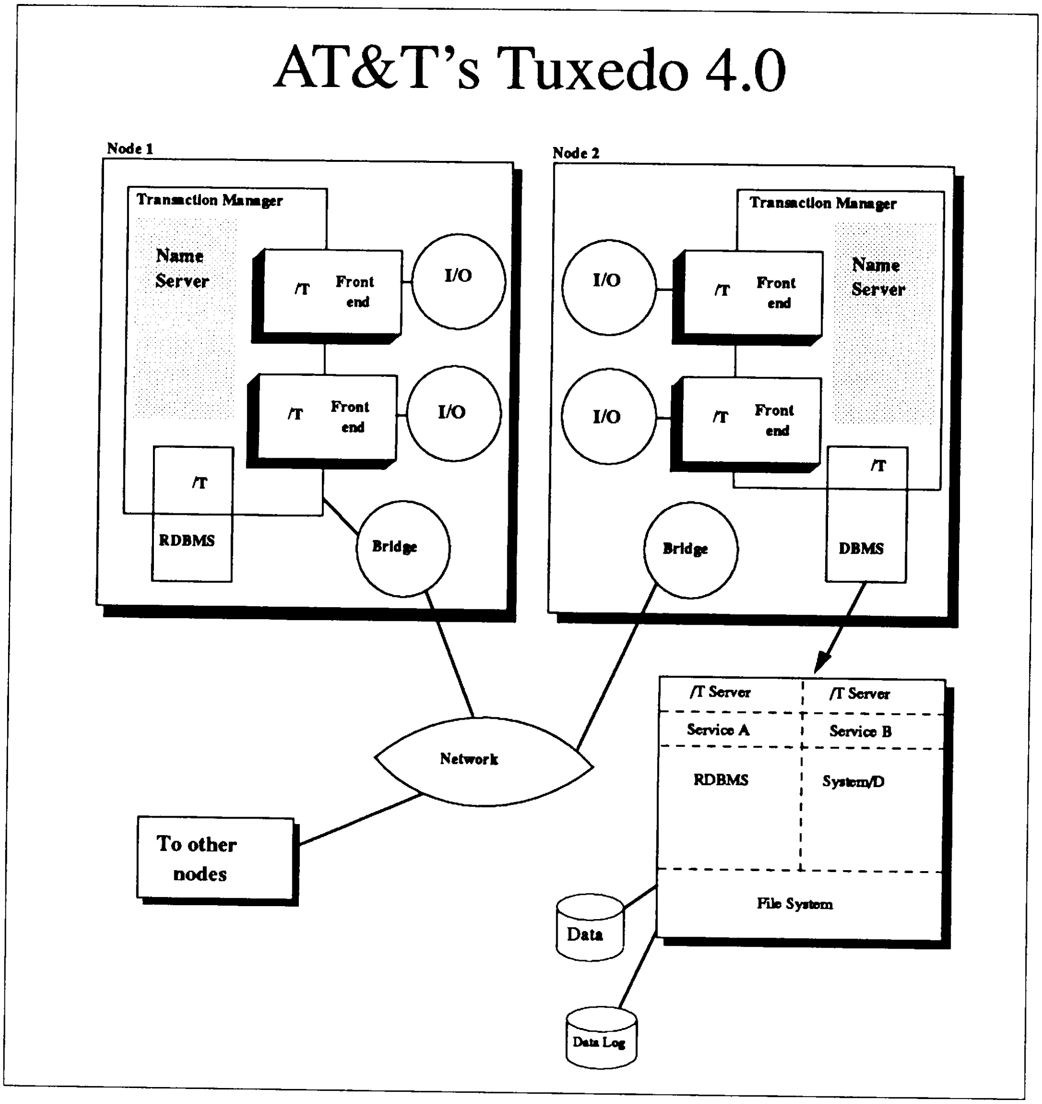


Illustration 6. AT&T's Tuxedo 4.0 is built on a client-server architecture and support for distributed execution of transactions. Tuxedo's Transaction Monitor is called System/T, or /T for short. It works with a name server to match client requests with data management services, like RDBMS. Tuxedo has its own, optional network DBMS, called System? D, as well as an SQL interface.

dors such as Unisys and Hewlett-Packard to provide support for Tuxedo within their DBMS software.

Optimization vs. Openness. It is too early to tell which pair of vendors has the architectural advantage. Both the multiserver architecture and the transaction monitor approaches will yield systems that can support larger number of users. And so users of all of these major RDBMSs will be able to do more with the Unix OLTP systems. We believe the winner in the competition for ever-better performance will be the vendor that can best optimize its transaction management components and RDBMS kernel, while leveraging standards.

We believe Sybase, Ingres, and Dharma Systems may have a short-term advantage in optimizing their systems. Each of these vendors provides what we call OLTP operating environments—data management and transaction management software created, optimized, and controlled by one vendor. Dharma's TPE* combines a multithreaded SQL RDBMS kernel and a Network Transaction Kernel based on Hewlett-Packard/Apollo's Network Computing System (NCS) RPC that manages transactions.

In contrast, Oracle and Informix appear more interested in moving to available standards by embracing Tuxedo. In the short term, both vendors may be at a disadvantage in optimizing their systems because Tuxedo is a source-level product over which neither has total control. Systems vendors will actually be handling the integration of the DBMS and transaction management software. We expect these vendors to need time to learn what it takes to make the combination of Tuxedo and their RDBMSs really hum.

Users talk about openness but we doubt many commercial will move to XA-based open OLTP architectures soon.

The wild card in the database market will be user demand for openness. By integrating AT&T's Tuxedo, Oracle and Informix will be early adopters of what looks like an important standard in Unix OLTP. Tuxedo is the first product to support X/Open's XA interface. As RDBMS vendors implement support for the XA interface in their products, applications written to Tuxedo should be able to access heterogeneous DBMSs without requiring modification.

Users talk about openness—witness David Sherr's earlier comment about the Holy Grail of software independence—but we doubt many commercial will move to XA-based open OLTP architectures soon. There are very practical reasons for this conclusion. First, none of the big RDBMS vendors are likely to offer XA interfaces until early 1991. Second, users say they are reluctant to support more than one RDBMS until they see a greater payoff for the additional effort required.

Still, both Sybase and Ingres are hedging their bets by moving to open their own APIs and committing to support the X/Opens XA interface at a future date.

TP MONITOR OPTIONS. And what of the transaction monitors the industry agrees are so critical to the future of Unix OLTP? At this point, users have two major options: a version of AT&T's Tuxedo Release 4.0 or VIS/TP, a CICS-like transaction monitor for Unix from VISystems.

Tuxedo and VIS/TP provide basic transaction-monitoring capabilities. But they are targeted at different groups of users. VIS/TP offers IBM CICS shops a practical approach to Unix OLTP. VISystems provides tools to port CICS applications to run under its own transaction system, and consulting services to optimize ported applications.

The one question about VIS/TP is its future position in IBM's Unix OLTP strategy. VISystems has a cooperative marketing agreement with IBM. But IBM will port a future version of CICS under Unix, and VISystems says it is not working with IBM on the project.

AT&T's Tuxedo, in contrast, doesn't provide tools to migrate proprietary OLTP applications to Unix. AT&T's strategy is to provide interoperability facilities, not migration facilities. Tuxedo is a new API aimed at users who are willing to start with a clean slate to develop applications. There are very few third-party applications for Tuxedo 4.0 at this point.

AT&T's "Wanna Be" Standard. AT&T announced Tuxedo 4.0 in February 1989 and has been pushing it hard as the standard Unix transaction monitor. Some vendors are taking the bait. At press time, Unisys, Amdahl, AT&T Computer Systems (no surprise there), and Sequent had licensed Tuxedo technology. Oracle and Informix have announced they will support Tuxedo. Hewlett-Packard appeared ready to license an enhanced version of Tuxedo from Independence Technologies Incorporated (ITI). ITI has added software layers to Tuxedo that make it easier to support additional communications and industry-specific protocols in the product. The first versions of Tuxedo should start appearing this fall.

Tuxedo appears to be catching on because it works, because the market is ready to consider a Unix-based transaction manager, and because it implements emerging OLTP standards. Licensees like the client-server architecture and support for distributed transaction processing of Release 4. (See Illustration 6.) There are even second sources for Tuxedo software (ITI). Licensees also like the opportunities AT&T has given them to add value to the base product.

Predictions that Tuxedo will dominate the market, however, are premature. Even big licensees hedge their bets on the technology. "Tuxedo is there and it works," said one licensee privately. "It gives us a foundation for added value, but we're not making a strategic commitment to it yet."

Vendors are reluctant to make strategic commitments to

OLTP Heavies Weigh In

The OLTP heavyweights are moving into Unix. Beginning in the last quarter of 1990, Tandem, Stratus, Unisys, and Hewlett-Packard will launch either Unix online transaction processing (OLTP) strategies or actual product lines. IBM and Digital won't be far behind.

This is good news for users. They'll soon have more choices in Unix OLTP and be able to work with vendors they already have relationships with. Also, IBM, Unisys, and HP are each getting into Unix OLTP with integrated environments that include not only an OLTP-ready Unix kernel and extensions such as disk-mirroring, but also transaction management software, a DBMS, and application tools.

The following positions the major vendors' new Unix OLTP strategies and products.

IBM. Consider IBM's various positions in Unix OLTP.

- IBM cooperatively markets, with VISystems of Dallas, VIS/TP, an environment transaction monitor for Unix that can host applications written under IBM's CICS.

- IBM officials have promised a version of CICS for AIX at an unspecified future date.

- IBM holds a stake in Transarc Corporation (Pittsburgh), which is building a distributed OLTP environment based on the Open Software Foundation (OSF) Distributed Computing Environment (DCE). IBM is also pledging to use OSF DCE technology.

It's difficult to see the strategy in these three elements, but IBM does have one. The company is working within standards organizations such as X/Open and ISO to create an open OLTP environment spanning its AIX and proprietary platforms. CICS will be the centerpiece of this environment. IBM will use the OSF DCE technology, probably working with Transarc, to implement this strategy. VIS/TP is a tactical solution only, although IBM expects to sell it in cooperation with VISystems for some time to come.

CICS is widely used for OLTP applications, principally IBM mainframes under MVS or VM. IBM also has a CICS version for OS/2. Each of these versions provides transaction management functions and an application programming interface (API) to those functions. IBM's strategy is to make the CICS API available across its hardware and operating system platforms, but not necessarily the transaction management functions provided by mainframe CICS. IBM will map CICS calls to equivalent functions on other platforms.

The result of this strategy will be a consistent application programming interface across multiple platforms. IBM will be offering its many mainframe CICS customers access to Unix and OS/2 for new applications, including distributed OLTP applications. CICS supports distributed transaction processing.

CICS is vital to IBM's OLTP strategy. Ninety-nine of IBM's largest 100 customers use it, and IBM has some 30,000 CICS licensees overall in 22,000 sites. The product has generated $2 billion in revenue during its 16-year lifetime. However, IBM supports standards, so CICS isn't the only transaction processing API that IBM will offer.

IBM believes it can build on the OSF DCE technology to create consistent system and transaction services across a variety of APIs on heterogeneous systems. IBM plans to support the transaction processing APIs offered by major relational DBMS (RDBMS) vendors—such as Sybase's Transact SQL—as well as transactional RPCs such as the one under development at Transarc. AT&T's Tuxedo API should be able to play in IBM's world as well.

We expect to see a formal statement of this strategy along with some AIX OLTP products in early 1991. OLTP, after all, is vital to IBM's future.

UNISYS. A lot of Unisys mainframes run OLTP applications, and the company has been active in Unix OLTP as a reseller of Sequent hardware. Now, Unisys is lashing these platforms together under a strategic umbrella called Open/OLTP.

Open/OLTP might better be called Standards-based/OLTP. Unisys plans to implement support for all and any OLTP standards in a Unix-based environment that integrates high-availability/reliability features, data management, and transaction management. In Unisys's view, the available standards today are System V Unix, the XA open DBMS interface, the OSI Distributed Transaction Processing (DTP) protocol, and the OLTP API defined by X/Open and implemented by AT&T's Tuxedo transaction monitor.

Unisys also plans to allow its Unix OLTP platforms to work in cooperation with OLTP applications running on its 1100/2200 and A Series mainframes, as well as OLTP applications on IBM mainframes and other non-Unisys environments. Standards such as XA and OSI DTP support cooperative processing between Unix and proprietary systems. Eventually, Unisys will add support for OLTP standards to its mainframe systems.

The first Open/OLTP products, due this year, will be based on Unisys U6000 processors. (Unisys has already made these products available through an early-developer program.) They will feature an OLTP-tuned System V kernel, an Oracle or Informix DBMS, AT&T's Tuxedo transaction management software, and Ally, a cross-DBMS application-generation tool that supports Tuxedo's API. Support for standards guided Unisys in its choices of partners. Oracle and Informix have committed to support the XA interface; Tuxedo was chosen because it supports XA and is based on X/Open's model for open distributed transaction processing.

The value of Unisys's Open/OLTP strategy is obvious to existing Unisys mainframe users. The strategy allows them to start distributing applications off their mainframes to lower-cost Unix systems. To the larger market, Unisys is offering a standards-based, integrated solution and a very useful applications-building tool. Ally is Unisys's trump card in Unix OLTP. None of Unisys's competitors have an equivalent tool.

DIGITAL EQUIPMENT. During the last two years, Digital Equipment has made a big play for OLTP with VAX/VMS products. It offers a VMS fault-tolerant system, with support for disk-mirroring and related features. However, Digital offers no comparable offering for its Ultrix users.

Digital's sole Unix OLTP offering is a Unix System V version sold only to telecommunications companies that require System V compliance. VAX System V is a tactical product that will be replaced by a strategic Unix OLTP product based on the OSF/1 kernel.

In the meantime, Digital isn't saying much about its strategy in Unix OLTP. The company says it does plan to implement fault tolerance for Ultrix in the future, but it has no transaction management software for Ultrix. And it won't say how it will accommodate existing VMS OLTP applications under Ultrix.

HEWLETT-PACKARD. Hewlett-Packard is pursuing Unix OLTP more aggressively than is Digital. HP will begin offering OLTP prerequisites such as disk-mirroring and automatic process recovery in a cluster of minicomputers this fall. And HP has decided to offer AT&T's Tuxedo transaction manager on its Unix platforms.

Still, HP's strategic platform for OLTP applications is its MPE proprietary operating system. Its Unix OLTP offerings will always lag behind MPE in functionality and innovation. Also, HP hasn't said how OLTP applications based on MPE will interoperate with HP-UX systems, and how HP will approach migration of MPE applications to Unix systems.

TANDEM AND STRATUS. Having built multimillion dollar companies on proprietary fault-tolerant systems, Tandem and Stratus will both release Unix-based, fault-tolerant systems during the fall of 1990. Tandem's Integrity S2 is based on the R2000 RISC chip from MIPS Computer Systems, running Nonstop-UX, a System V-compliant (System V Release 3.2) Unix. Stratus has ported Unix to its existing XA2000 hardware, which is based on the Motorola 68000 family. Stratus's FTX is also System V Release 3.2-compliant. We expect both to offer support for AT&T's Tuxedo transaction management software in the future.

Both Tandem and Stratus have replicated the full range of benefits they offer with proprietary technology on their Unix systems. These are:

- Fault tolerance. Each vendor provides fault tolerance with its own architecture. Stratus has the same hardware-based architecture for both its FTX and proprietary systems, while Tandem is introducing a hardware-based architecture with Integrity S2; its proprietary NonStop/Guardian system has software-based fault tolerance.

- Online changes. Both vendors allow system components to be swapped out without disrupting operations. Also, both allow system reconfigurations on the fly.

- Remote diagnostics and support. Both vendors continuously monitor their customers' systems from remote support centers. The remote centers can troubleshoot system problems. Tandem's remote support capability is new; Stratus has always offered remote support.

Stratus hasn't changed its basic approach to fault tolerance in building its Unix products. Tandem, as previously noted, has switched from software-based fault tolerance in its proprietary systems to a hardware-based architecture. It also has introduced a new hardware architecture with its Integrity S2—an architecture based on third-party RISC chips. Tandem obviously hopes to cash in with Integrity S2 on the superior price/performance available with RISC technology. Stratus will have to make the move to RISC at some point in the future as well, but, for the moment, it is sticking with its existing hardware base. The implication for Tandem is that it won't have a full range of Unix OLTP boxes—from low-end departmental systems to mainframe-class systems—to offer to customers for some time. Stratus will. However, Tandem's choice of a generally available RISC chip should allow it to build a full line of processors fairly rapidly.

CONFLICTING INTERESTS. Each of the OLTP heavyweights has a long-standing investment in proprietary OLTP technology. When they are given a choice between Unix and their proprietary systems, proprietary always wins. Margins on proprietary systems are higher, and sales forces know their proprietary products best. Every vendor with proprietary interests faces these same conflicts over Unix.

But the market is pushing vendors hard to offer good and economical Unix solutions. The OLTP vendors who come to grips with Unix will be the big winners as Unix OLTP takes off.

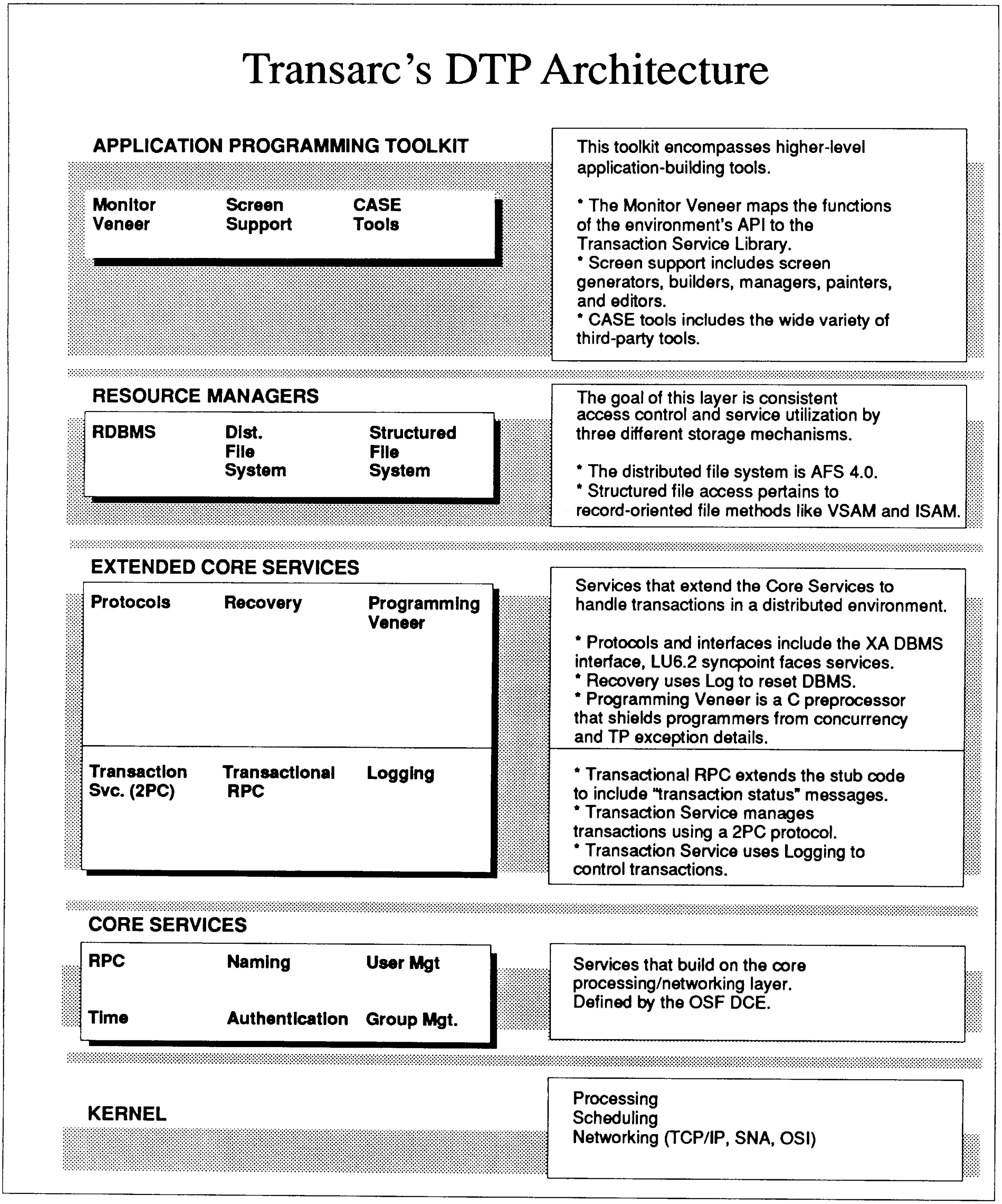


Illustration 7. Transarc's architecture for distributed, open-interface OLTP assumes kernel-independent Core Services defined by the Open Software Foundation in its Distributed Computing Environment. Transarc then adds extensions like Transaction Processing Services and an OLTP-enhanced RPC, as well as a programming veneer, to round out the environment.

Tuxedo for two reasons. The first is key gaps in Tuxedo's functionality. Tuxedo does not support PCs or Macintoshes as clients. It only supports character-based Unix clients. In early 1991, AT&T will begin fixing this problem by releasing a generic DOS client. In addition, Tuxedo relies on standard Unix password security. Even AT&T concedes that it needs a validation service to check the passwords users submit against access control lists.

The second reason for vendors' hesitation about Tuxedo is their doubts about its openness to new technologies. Vendors who believe remote procedure calls are the simplest method for building network applications, for example, point out that Tuxedo doesn't support an RPC yet. Others see Tuxedo as a monolithic product that won't be easy for AT&T to open up. AT&T has not, for example, provided an open interface to its logging service. If a licensee wants to substitute a new logging service, it is very difficult to do so. These vendors would rather see a modular architecture akin to the Open Software Foundation's Distributed Computing Environment (DCE). Others worry that the industry doesn't know enough yet about transaction monitors in open, distributed OLTP to intelligently choose a standard.

DCE goes further than previous technologies supporting distributed OLTP by insisting that any system must be totally modular and be portable to a variety of operating systems, not just Unix.

Users, on the other hand, appear much less interested in debating Tuxedo's merits than in evaluating it. The big users we talked to were all either using Tuxedo, evaluating it, or planning to evaluate it in the very near future. Tuxedo is the most immediately available alternative among Unix transaction monitors today, they say.

The Distributed Future

Vendors and users alike believe the future of Unix OLTP lies in distributed transaction processing. This is why distributed OLTP is at the heart of X/Open's project to define open OLTP interfaces in addition to ISO's effort to define transaction processing protocols.

Distributing the execution of transactions across a network is a difficult technical feat, particularly in an environment that is not under the control of one vendor. Users are skeptical that reliable distributed OLTP systems will be available anytime soon. "We haven't built enough distributed systems to know what the proper constructs for distributed OLTP are," says Shearson's David Sherr. Still, Sherr is one of many who are closely watching the efforts of the OSF and others to deliver on the promise of flexibility and scalability of distributed systems.

Earlier this year, assumptions about distributed transaction processing were challenged by the OSF's DCE technology initiative. DCE goes further than most previous technologies supporting distributed OLTP by insisting that any system must be totally modular in structure and be portable to a variety of operating system kernels, not just Unix. (See *Network Monitor*, July 1990.) Modularity promises to ensure that systems, resource management, and communications software will evolve to higher function and greater efficiency without blowing away applications built on it. Portability to a variety of kernels promises application portability. Ultimately, the user's investment in applications is protected and extended. No Environment/toolkits or RDBMSs discussed above meet both conditions.

Transarc Corporation (Pittsburgh), vendor of the Andrew File System selected as part of the DCE, has become a leading proponent of the DCE approach for OLTP. Transarc expects to begin announcing the first products that conform to its DCE-aligned architecture for OLTP in late 1990, with products becoming available in 1991. Transarc also has a strong IBM connection; IBM has an equity investment in the company. We expect IBM to work with Transarc to deliver a future CICS version for AIX.

TRANSARC SOFTWARE ARCHITECTURE. Transarc's distributed OLTP environment can be layered atop a variety of operating system kernels. It incorporates the services provided with OSF's DCE: threading services, RPC, distributed time service, naming services, authentication services, and user and group management. Transarc calls these Core Services, and they support a variety of distributed applications. (See Illustration 7.)

Extended Services. Atop this distributed services platform, Transarc adds the following Extended Distributed Systems Services that support OLTP application development and management:

- Distributed Transaction Service, which coordinates and manages transactions involving multiple processors and shared resources using a 2PC protocol

- Transactional RPC, which is Hewlett-Packard/Apollo's NCS with additional semantics that describe transactions

- Logging Services, which information needed to recover after a transaction aborts for any reason

- Recovery and Locking Services, which are available for use by a variety of resource managers

- Protocol and Interface Translators, which allow interoperability of OLTP systems using different protocols (LU6.2, etc.) and interfaces (XA, etc.)

- Programming Veneer, which uses C procedures and macros to shield developers from the details of transaction management as they build their applications

Taken together, these services define the components of a transaction manager. Transarc plans to call them, collectively, the Transarc Toolkit.

Resource Managers, Management, Development Tools. Resource Managers in the Transarc architecture conform to the X/Open model by accommodating RDBMSs, file systems like C-ISAM and VSAM (as well as Transarc's own Structured File System), and distributed Unix file systems. Systems management is a big question mark in the architecture because so much work is still to be done to define utilities in distributed environments. Transarc hopes the OSF will provide help in this area with an upcoming technology initiative.

At the highest level of the architecture are high-level tools like screen builders and CASE environments. Transarc is developing an application development and run-time environment called the Transarc Application Programmers Environment. The environment will provide a screen generator, a transaction monitor library, a security library, and basic system administration facilities. In addition, Transarc hopes to attract third-party tool vendors to its environment.

Conclusions

There are a lot of reasons to be excited about Unix OLTP. It clearly comprises a set of technologies that can help businesses solve problems at attractive costs today. At the same time, however, Unix OLTP technology is very much in flux. Vendors and users are struggling to understand how they can build bigger systems, more powerful systems, and distributed Unix OLTP systems. As they do, they're redefining OLTP. IBM mainframe veterans are horrified by the thought of exposing OLTP systems to the inquiries of users. Unix OLTP users reply: "Sorry, we need it that way. Change the rules."

We believe the number of Unix OLTP applications will continue to grow. The rate of that growth, however, will be determined by the success of vendors and users in dealing with four issues: system management, integration and optimization, and standards.

System Management. Transarc is about as bullish as a company can be about open and distributed OLTP. Yet even Transarc concedes that it will be years before users have the kind of system management tools they'll need to manage large distributed OLTP systems. OSF's next big technology initiative will be in distributed systems management. Everyone in Unix OLTP hopes OSF finds a robust basket of system management technologies to back—and fast.

Integration and Optimization. Tight integration and optimized interactions between the elements in a system are the keys to reliable performance. Integration and optimization are real challenges in Unix OLTP systems because of the number of vendors involved in the typical system. Today, most systems are the products of a hardware (and operating system kernel) supplier and a database vendor. Add to this mix a transaction monitor, and the number of potential problems goes up.

Standards vs. Innovation. There's a tug-of-war in Unix OLTP between those who believe Unix OLTP needs standards now and those who believe Unix OLTP needs innovation now and standards later. The standards agreements needed to open up Unix OLTP are complex and intertwined. The fact that the industry has been able to agree on the XA interface for RDBMS services is a positive sign. However, we don't expect a full suite of standards to be complete for two years.

We expect the industry will need three years to sort out these issues. In the meantime, Unix OLTP will appear to be a fertile field to some, an unstable landscape to others. We don't believe users should preoccupy themselves with the standards fights in Unix OLTP. Unix OLTP presents real opportunities to deliver quality applications at low costs today, and functionality is advancing rapidly. If you don't take advantage of the technology where you can, your competitors will.

Next month's issue is a review of the Progress relational database management system and applications development environment.

For reprint information on articles appearing in this issue, contact Anne Workman at 617-742-5200.

NEWS ANALYSIS

PRODUCTS • TRENDS • ISSUES • ANALYSIS

• INSIDE •

• GUI TOOLS •

A More Complete UIMS

Porting Unix applications to multiple X Window-based toolkits has been too time-consuming and expensive to make it an option for most developers. Thus, the need for user interface management systems (UIMSs) is greater than ever. A UIMS is a tool for creating and testing user interfaces quickly and interactively instead of programming complex toolkit components. The technology is nothing new; UIMSs have been around for some time. But, because of the recent trend toward graphical interfaces, they've gained a great deal of attention lately—and a great deal of refinement. TeleUse by TeleSoft provides an example of the advances being currently made in UIMS technology.

TELEUSE. Like most UIMSs, TeleUse has a screen layout editor, which TeleSoft has dubbed VIP, to create and modify user interfaces. The system includes the Motif and Athena widget sets. It lets you select from menus of widgets—buttons, scrollbars, menu bars, pull-down menus, etc.—and palettes of attributes—color, font, geometry, patterns, etc. In addition, TeleSoft has widget templates, a useful feature for viewing and implementing plug-and-play interface components. You can combine a collection of widget templates to form self-contained, reusable, logically related interface components. They also feature inheritance, so, when you change a widget attribute, it can be replicated throughout the entire template.

Another useful feature is the widget browser, which gives you a broader perspective of the interface-in-progress. TeleSoft interfaces are hierarchical in nature, and the browser lets you view and manipulate the hierarchical tree structure of a screen layout. From the widget browser, you can create, move, copy, delete, customize, and name specific widgets. And you can jump into the browser window while the interface remains active. Therefore, changes made using the browser are immediately reflected in the screen layout in the work area. Very nicely implemented. Very sophisticated.

Once the layout is complete, the screen description is stored as an ASCII file that can be modified or reused without recompiling.

A Language for Dialogue Management. However, there's more to designing an interface than merely laying out the screen. The other critical component of a UIMS is dialogue control—i.e., dealing with the way users interact with the interface and the way the interface and the application communicate. Most UIMSs rely on C to implement interface functionality. TeleSoft has implemented a special dialogue language, called the D language, for specifying and controlling the interaction between the user and the computer. TeleSoft claims that D looks very much like C. However, it is used specifically to create events (commonly known as *callbacks*) that describe specific dynamic behavior. Events in TeleUse can either be external, where the function is the result of a user interaction (e.g., a mouse click causes a dialog box to appear), or internal, where the function is sent from within a D module (e.g., updating a widget with a specific string). D is also rules based: A single event can have several rules depending on its circumstances (or conditions).

TeleUse also includes a set of tools to work with the D language:

- A graphical debugger, which allows you to set breakpoints, view variable contents, and step through actions
- A D interpreter, which is especially advantageous for testing and prototyping, making it possible to execute D modules without recompiling after each modification
- A D compiler, which translates D code into C code

Integrating Interface and Application. Once the interface is designed and the dialogue is written, TeleUse provides tools for integrating them with the application code. The first is the run-time library, a collection of functions needed to convert description files into an executable program. It contains functions to control events, routines to integrate screen description files, bindings between TeleUse and X Window intrinsics, and functions for modifying the interface at run-time.

The user interface (UI) builder then uses the features of the run-time library to bind the interface to application code. The UI builder handles all the details involved in the integration process (e.g., information about paths, filenames, and libraries). It also uses the D compiler for translating the interface code into C code.

Availability. TeleUse is available for $9,900. TeleSoft is marketing the product heavily to vendors and has already made a distribution agreement with Digital Equipment. TeleUse will run on DECstations, Data General AViiON, and Sun (3 and 4) workstations.

CONCLUSION. TeleSoft has made some worthy gains in UIMS technology. Its dialogue language is perhaps the most significant, but it has other features we haven't seen in other UIMSs, such as the widget templates, the interactive attribute palettes, and the widget browser.

However, we're still looking forward to the time when the dialog component of interface design becomes more user oriented. We're not complaining about TeleUse specifically here, but about UIMSs in general. Today, UIMSs are programmers' tools, and that's too bad. Interface designers are just that—designers, not programmers (or at least they should be). They know about the cosmetics, the composition, and the psychological implications of user interface design. UIMSs should be geared for them. End users are another force to be dealt with. Eventually, users should be able to customize an existing application for the way they want to interact with it. And users will be able to design their own applications, or overlap them, or reuse functions they like from other programs and libraries.

One hole in TeleUse is its lack of support for OpenLook. The industry hasn't come to an X toolkit conclusion yet, so TeleSoft's selection of Motif may be somewhat shortsighted. TeleSoft representatives say that they have been looking into the idea of support for both toolkits (and we are finally beginning to see interface technology that supports both). However, they maintain that the technical differences between Motif and OpenLook are enough to get in the way of merging both toolkits smoothly. Furthermore, OpenLook has a few different versions to contend with. Nonetheless, until a true standard emerges, toolkit independence may be the best—albeit difficult—solution.

—L. Brown

•PLAN 9•

Ideas for Distributed Computing

Plan 9 is the latest research operating system to emerge from AT&T's Bell Laboratories. For you trivia buffs, it is named after an obscure, cult science fiction movie (*Plan 9 from Outer Space*) that never made it to the big time. Plan 9, according to Peter Weinberger, chief scientist at AT&T's Software Operation, is a Unix-like operating system because it has been developed by the creators of Unix, including Ken Thompson. For example, Plan 9 still uses commands like fork, exec, read, and write. In Plan 9, these system calls are the same no matter if the system is dealing with a file, a process or a directory. One of the major differences is that these pioneers have learned a lot since they developed Unix more than 20 years ago. Being able to start from scratch they have been able to write an operating system that takes distributed processing into account.

How does Plan 9 differ from Unix?Primarily, in the way it allocates files to a name space. In Unix, the name space consists of files on a local disk. Plan 9 assumes that all files are remote, implying that a user can put all kinds of objects into this name space. Even though these objects are at a level above the operating system kernel, they are treated as though they were resident in the operating system. As a result of the object orientation of Plan 9, a process can be fooled into thinking that it is dealing with the entire screen. However, it is writing not to a physical I/O device, but simply to a process. In fact, Plan 9 makes a process look like a directory that contains files. While there aren't specific files, Plan 9 puts a structure on a process that makes controlling objects easier. This is especially important in a distributed environment. In essence, Plan 9 takes the concept of a name space and uses it throughout as a tool for distributed computing.

The key implication is that the application and the operating system are location independent since they don't have to know whether objects are local or remote. So, if I were to start up a window manager in this environment, it would not have to take up the entire screen. This technology could have implications for running a multitasking operating system like OS/2 as a process, much the way DOS is run as a process under Unix today.

How close is Plan 9 to becoming a commercial reality? Not very. It will probably be a long time before it finds its way to a user environment. It is primarily a research tool that permits Bell Laboratory researchers to continue to develop and understand distributed computing. It is, in fact, in the same league as other distributed research operating systems, such as Amoeba, Chorus, and Clouds—all microkernel operating systems that are targeted to the distributed environment. Mach, another distributed microkernel operating system, has taken center stage since the Open Software Foundation (OSF) selected it as the kernel for OSF/1.

However, because Plan 9 comes out of Bell Laboratories, we expect some of the concepts to find their way into System V within the next few years. AT&T is under pressure from groups like Unix International to keep the operating system moving towards a modern base. Some of that pressure is coming from the desire to compete with OSF's directions, and some is coming from software developers and leading end users looking for a good way to move to distributed computing in the coming decade. *—J. Hurwitz*

•WORKSTATIONS•

Merger of HP and Apollo Workstations Begins

With the acquisition of Apollo in May 1989, Hewlett-Packard (HP) gained impressive technology and a much stronger presence in the workstation market. On the downside, the company faced a difficult task—that of creating a single, comprehensive workstation line while, at the same time, preserving customer investments in both HP and Apollo systems *and* maintaining competitive price/performance offerings in an increasingly cutthroat market.

HP is rising to the challenge. One year after acquiring Apollo, the company has taken a significant first step toward converging the HP and Apollo product lines. The recently-announced HP Apollo 9000 Series 400 workstations, based on the Motorola 68030 and

HP Apollo 9000 Series 400 Workstations

	Model 400dl Desktop 2-D mono graphics	Model 400t/425t Desktop 2-D/3-D color	Model 400s/433s Deskside 3-D color/high-end graphics
CPU	50 Mhz 68030	50 Mhz 68030/25 Mhz 68040	50 Mhz 68030/33 Mhz 68040
PERFORMANCE	12 MIPS 0.5 Mflops	12 MIPS/20 MIPS 0.5 Mflops/3.5 Mflops	12 MIPS/26 MIPS 0.5 Mflops/4.5 Mflops
GRAPHICS	VRX mono	VRX mono/color Personal VRX	VRX mono/color Personal VRX Turbo VRX
MEMORY	8 or 16 MB ECC RAM	8 to 64 MB ECC RAM	8 to 128 MB ECC RAM
I/O	RS232	SCSI, Centronics/ parallel, RS232, HP-IB (optional)	SCSI, Centronics/ parallel, RS232, HP-IB plus options for DIO, DIO II, ISA, and EISA (433s only)
MASS STORAGE	up to 4.6 GB external	up to 400 MB internal up to 4.6 GB external	up to 1.2 GB internal up to 4.6 GB external
NETWORK	Ethernet	Ethernet, Apollo Token Ring, IBM Token-Ring	Ethernet, Apollo Token Ring, IBM Token-Ring
PRICE*	from $4,990	from $6,990/$8,990	from $13,990/$15,990

* All prices include VRX monochrome graphics

68040 chips, provide a single hardware platform that runs either the HP-UX or the Apollo Domain/OS operating system. From a compatibility/migration perspective, the new workstations are object-code compatible with existing HP and Apollo workstations; current applications can run unchanged on the merged hardware.

HP also continues to push on the price/performance and functionality fronts. The low end of the Series 400 provides 12 MIPS for under $5,000—performance in the $400/MIP range. At the high end, HP is the first vendor to introduce a workstation based on the 68040 chip, providing major performance enhancements over the 68030. According to HP, floating-point performance is ten times better and integer performance is up to four times better. Along with the workstations, HP has also introduced a new high-performance family of graphics subsystems, the HP Apollo 9000 VRX products.

THE MERGER STRATEGY. The first year following the merger marked a transition phase as HP worked to incorporate the Apollo folks organizationally and functionally. Apollo is now a division within HP responsible for RISC workstation research and development. During this time, HP/Apollo also introduced a number of point products, such as the Apollo DN 2500 (the first workstation to break the $4,000 barrier), the RISC-based Apollo DN10000 providing up to 44 MIPS, and HP's 345 and 375 workstations based on the 50 Mhz Motorola 68030.

Now in year two, HP and Apollo are focusing on merging their two workstation hardware platforms into one. The first step, represented by the Series 400 and the VRX graphics, merges the company's Motorola 68xxx workstations, or CISC (complex instruction set computing) technology. Both HP and Apollo have a broad range of existing offerings based on the Motorola 68xxx, including the HP 9000 series 300 and the Apollo Series 2500/3500/4500. With the new workstations, HP continues its strong commitment to the Motorola 68xxx.

We expect the next step to be a convergence at the hardware level of the company's two RISC technologies—HP's PA-RISC and Apollo's Prism architecture, represented in the HP 9000 Series 800 workstations and Apollo's DN10000. (See *News and Analysis*, Vol. 5, No. 1 for more details on this strategy.)

Year three will most likely bring a merger at the operating system level, with HP's implementation of OSF absorbing both HP-UX and Domain/OS.

THE WORKSTATIONS. The entry-level desktop Model 400dl workstation is designed for markets such as computer-aided software engineering (CASE) and electronic publishing. Models 400t/425t, also desktops, provide additional power (up to 20 MIPS) needed for electronic and mechanical design as well as for CASE applications. The high-end deskside workstations, Models 400s and 433s, go as high as 26 MIPS and add expandability. These are designed for graphics-intensive applications such as mechanical engineering, industrial design, simulation, and scientific visualization, and can support the high-end TurboVRX graphics system. The chart on page 21 summarizes the features and functionality of the Series 400 workstations.

Servers. The 400s and 433s can also be configured as servers, starting at $12,990 (without the display).

Upgrades. In line with its longstanding policy of providing upgrade paths for its workstations, HP offers chip upgrades from the 68030-based 400s and 400t to the 68040-based 433s and 425t, respectively, for $2,000. The 400dl is upgradeable to the 425t for the same price. The 68040 workstations and the upgrades will be available in the fourth quarter of 1990.

HP has also promised what it calls "40+" upgrades for the future—that is, up to 40+ MIPS—to either future 68xxx chips (at a guaranteed price of less than $5,000 for workstations purchased by the end of 1990) or to RISC (with an attractive trade-in value).

Ease of Use. The Series 400 supports HP VUE (Visual User Environment) 2.0, HP's newest version of its X11 and OSF/Motif-compliant graphical user interface. HP VUE is the first step toward incorporating HP NewWave into the Unix environment. HP VUE costs $550 and is currently a separate product for Domain and HP-UX workstations. By the fourth quarter 1990, HP expects to bundle HP VUE with all workstations.

Series 400 customers can also get HP's Instant Ignition software package. This contains preloaded and preconfigured software (productivity tools, HP VUE 2.0, operating system, application demos); the user just plugs in the workstation, loads the disk, and goes.

VRX GRAPHICS. HP is stressing speed and realism in its HP Apollo 9000 VRX graphics family. VRX also offers software compatibility across HP/Apollo and other vendors' graphics products through the implementation of the PHIGS standard.

VRX. Entry-level VRX for 2-D graphics comes in either monochrome (19-inch display, 40K 2-D vectors/second, from $1,200) or color (16 or 19-inch display, eight color planes with two overlay planes, 256 simultaneous colors from a palette of 16.7 million, 130K 2-D vectors/second, from $4,200). Standard resolution on all of the VRX graphics products is 1280 x 1024. HP also expended a lot of effort to improve the X Window performance of the VRX mono/color products. This is done with a device-dependent X Window system driver with an R4 server.

PersonalVRX. PersonalVRX provides accelerated 3-D color graphics and is designed to meet the needs of the mechanical computer-aided design (CAD) market. It uses an Intel i860 RISC processor as the graphics transform engine and comes in three configurations. System features include 8 or 16 color planes (the 16-planes version has a 16-bit Z buffer); 4 overlay planes; up to 270K 3-D vectors/second; up to 50K polygons/second; many speed features including sectioning (the ability to look inside an object), capping (the ability to make a sectioned object look solid), inference-checking, and contour-mapping; realism features such as the Personal Visualizer ray-tracing, and radiosity through progressive refinement.

Ray-tracing provides photorealistic rendering with reflections and sharp shadows. Radiosity allows the user to model the diffuse qualities of light. Radiosity is already available in HP's existing TurboSRX graphics products. According to HP, it is the only vendor that currently provides radiosity with progressive refinement. The PersonalVRX subsystem starts at $14,000 in a system, or $15,000 standalone.

TurboVRX. At the high end, TurboVRX provides several impressive enhancements for complex 3-D graphics applications. Designed to meet the needs of the advanced design engineer, TurboVRX provides 20 times the performance of HP's existing SRX, and 5 times the performance of TurboSRX.

TurboVRX comes in three configurations with up to three i860s running in parallel as transform engines to boost graphics performance. TurboVRX adds the following capabilities to those of the PersonalVRX: up to 1 million 3-D vectors/second; up to 778K 3-D anti-aliased vectors/second; 24 color planes with a 24-bit Z buffer; texture-mapping with perspective interpolation (the ability to take a 2-D image and map it to a 3-D object); anti-aliased vectors and polygons; and alpha blending. TurboVRX starts at $40,000 in a system, or $41,000 standalone.

TurboVRX is only available for HP-UX Series 400 workstations. Since the Apollo Domain 10000 has its own high-end graphics imaging system (10000 VS), HP made the decision not to commit the significant resources required to implement TurboVRX on the Domain/OS. In addition, the VRX family will be supported in the future on the new HP/Apollo RISC systems when they are announced.

Graphics Software. The Personal Visualizer is licensed from Wavefront Technologies and bundled with the PersonalVRX and TurboVRX graphics systems. This menu-driven rendering application gives users access to VRX features, such as ray-tracing and radiosity, even if the features have not been programmed into specific applications.

HP-PHIGS provides source-code compatibility for graphics based on Domain/OS and HP-UX, operating at the same performance levels as that of the HP Starbase and Apollo G*R graphics libraries. The Series 400 also supports G*R on Domain/OS and Starbase on HP-UX. *—J. Davis*

•USENIX•

More Frustration Than Innovation

Usenix conferences have long been known for providing an exchange of innovative technologies under Unix. This summer's conference was no different, except that we sensed a great deal of dissatisfaction from developers about the inadequacies of the Unix operating system. Most sessions addressed problems associated with implementing a Unix system and contributed makeshift—but valuable—solutions. A few others offered alternative kernel technology. It became clear that, since Unix has been pivoted into the commercial mainstream, it needs to evolve into a more robust system. What we need are developments beyond the kernel.

WHAT'S WRONG WITH UNIX? Unix is over 20 years old, and it was never intended to handle the kinds of services that systems require today, like support for multiprocessor systems and distributed computing. While Unix has been tinkered with to let it play in newer paradigms, more often than not, the tinkering is nothing more than a band-aid solution. Take implementing Network File System (NFS) under Unix, for example. A number of inconsistencies arise when you try to retrofit a stateless file system on an operating system that has no notion of statelessness. And security? If you're looking to Kerberos, the OSF selection for authentication services, all your system programs that require network services will require modification. These examples are just two of many. Developers are simply spending too much time tweaking the system. As one Usenix attendee complained, "I'm writing bad programs for a bad operating system—but I'm really good at it."

To complicate matters, as each vendor has made extensions to Unix, it has lost its consistency. Each implementation has its own idiosyncrasies and winds up looking more like a proprietary system to the end user. (Actually, in many ways, Unix is a proprietary system. It is owned by a single vendor and is not developed through an "open" process.) Furthermore, Unix has picked up a lot of excess baggage over the years to handle new functionality. Yet, it still doesn't provide a proper model for multiprocessor architectures. The few vendors that have made significant advances in multiprocessor technology have done so by creating complex extensions to the Unix architecture. Unix has simply outgrown its model, and many maintain that it's time to either rewrite the system or turn to another.

NEW KERNEL TECHNOLOGY. Much of the excitement at Usenix centered on the new alternative kernels being developed at universities and research centers. Mach, of course, is the best-known example of the advances being made in operating system design. But it seemed that most of the attendees were interested in more obscure solutions: Chorus (Chorus Systems, France), Sprite (University of California Berkeley), Amoeba (Vrije University, Amsterdam), and the V kernel (Stanford University), as well as Plan 9—a research system being developed at Bell Labs which has been hailed by some as the next Unix. It was unveiled during the UK Unix User Group (UKUUG) Conference in London in July.

Such systems have a common goal

of efficiently supporting distributed computing—often with sophisticated techniques of memory management and process control via a small, modular kernel architecture. This "microkernel" design makes for a more streamlined, efficient system. The functionality necessary for implementing distribution and multiprocessors is the very essence of the operating system itself, not an exclusive set of extensions.

TODAY'S SOLUTIONS. Unfortunately, most of the "new" operating systems are just research projects. Chorus alone is marketed commercially. Like other so-called modern operating systems, Chorus has a microkernel architecture. It offers a Unix binary-compatible version of its nucleus (a.k.a. kernel), which adds to Unix the advantages of improved performance, distributed virtual memory, multithreading, real-time processing, and dynamic reconfiguration. Chorus has just recently set up a U.S. shop, but it has amassed a number of European customers, among them the French Ministry of Defense, the European Space Agency, and NATO.

Daily Drudgery. Though intrigued with the prospect of a new operating system, most Usenix attendees sought solutions to just get them through the day. Forget new development; forget application design. The session on System Administration was packed with people haggling over the limitations of *sendmail* and weak device drivers. Unfortunately, very few solutions surfaced. Vendors are shrugging their shoulders when it comes to system administration, leaving users to make their own mistakes. We did, however, talk to a new company that solely provides service and support—but only for freely distributable software (i.e., software that you can exchange without a licensing fee). Cygnus Support (Palo Alto, California) provides services—from installation to bug fixes, porting, extensions, and upgrades—on a yearly basis. Despite its very focused target audience and its recent entrance into the market (the company is only a few months old), Cygnus Support is profitable and has already garnered some impressive clients, including Sun Microsystems and NASA.

Security also seems to be an ever-intensifying problem. One presentation described a useful security tool called COPS (Computerized Oracle and Password System), developed at Purdue University, which allows administrators to check for possible security holes. Another paper described a way to implement a secure Internet gateway.

System Development, Not Applications. The COPS presentation was just one in an isolated session on Applications. The Usenix conference leaned more toward system development and architectural issues than application development. The conference focused particularly on distributed computing, with sessions on distributed file systems, X Window technology, and distributed operating systems. While many presentations offered new developments—such as a distributed programming language and an X Window manager—many others offered system remedies—such as improving cache management, memory bandwidth, and NFS file server performance.

CONCLUSION. Unix may hold promise, but it also has its problems. The concerns brought up at Usenix indicate an impasse in the growth of commercial Unix. The typical Unix user is no longer a techie Unix fanatic, and commercial users will not tolerate the system's inadequacies. *—L. Brown*

•SCO•

A Target Platform for OSF

SCO's various Unix versions are practically ubiquitous on the 386 platform. OSF needs to preserve its members' investments in 386 applications when moving to OSF/1. Thus, a marriage made in heaven.

OSF and SCO have announced a joint development design project that will result in the OSF/1's ability to run applications written for SCO Xenix System V, SCO Unix System V/386, Open Desktop, and other Unix System V/386 environments.

The goal of the project is to both provide a large library of existing applications that can run on OSF/1 and to set a clear target platform for application development. The Intel 386/486-based computers are fast becoming the leading Unix platforms. Both OSF and SCO are taking advantage of this.

WHAT'S IN IT FOR EACH? SCO is the leading vendor of 386-based Unix systems worldwide, capturing about 90 percent of that market. OSF obviously benefits from aligning with this phenomenal installed base. SCO also benefits by becoming identified with OSF, well-known for its work on developing standards in the open systems market.

OTHER PLAYERS. Several other vendors play a part in the project. Locus Computing Corporation will do the actual software engineering of the SCO-compatible OSF product. Locus already supplies X Window System products, MS-DOS emulation, and other system applications to SCO. All development will be done using SCO's Open Desktop's OSF/Motif environment.

Hewlett-Packard will provide hardware for the project. HP's Personal Computer Group already resells SCO Unix V/386 under the HP name. The company, therefore, has a strategic interest in ensuring that SCO's line of Unix versions continues to be regarded as meeting open system standards. The agreement with OSF certainly suggests that this will happen.

PROJECT STATUS. The project is already underway. SCO and OSF have promised that regular status and review will be provided to OSF members as part of the OSF open process. The goal is to ensure compatibility with SCO Unix Systems by the first half of 1991. *—R. Marshak*

Patricia Seybold's Office Computing Group

The Office Computing Group (OCG) is a rapidly growing research, publishing, and consulting firm focused on the evolution of information technology and its impact on organizations. It provides analysis, education, and direction concerning the development of technology and its relationship to organizational performance.

Members of the interdisciplinary team which comprises the OCG share a keen interest in the interplay between information systems technology and business design.

The OCG publishes three monthly technical reports:

• ***The Office Computing Report:***
For over twelve years, this monthly report has delivered research tracking the key issues, new products and evolving technologies in the corporate computing environment.

• ***Unix in the Office:***
This monthly report discusses applications and technology trends in the Unix industry. In-depth feature articles spotlight Unix databases, operating systems, innovative applications, and standards issues.

• ***Network Monitor:***
This publication tracks the evolution and development of Distributed Network Computing, the environment that will become the foundation for Information Systems in the 1990s and beyond.

In addition to the technical reports, the OCG also publishes a monthly AudioNewsletter titled ***Paradigm Shift—Patricia Seybold's Guide to the Information Revolution.*** This innovative service, both an audiocassette and a printed newsletter, highlights interviews with key industry leaders and analysts and provides Seybold's shrewd evaluations of their discussions.

The OCG also publishes comprehensive research studies on key industry trends, including: ***IBM RISC System/6000 Product Line Assessment; Hewlett-Packard 9000 Series 800 Product Line Assessment; Object Orientation—Concepts and Implementations; Executive Support Systems; Digital's Office Strategy;*** and ***AS/400: An Assessment.***

As an outgrowth of its publishing activities, the OCG sponsors a number of industry conferences, including:

• ***Patricia Seybold's Technology Forum:***
Launched four years ago, this annual event focuses on a particular area of technology and provides corporate end users with an opportunity to sample cutting edge applications and to share implementation strategies.

• ***Executive UniForum Symposium:***
Cosponsored with UniForum and X/Open, this annual seminar spotlights the applications development environment of the 1990s.

The OCG also provides a variety of consulting services to both computer industry vendors and users. The consultants excel in helping clients analyze the industry within the larger context of the emerging global economy and in developing strategies—strategies for company direction, products, purchases, marketing plans, announcements, alliances, and internal teambuilding

Patricia Seybold's Office Computing Group
148 State Street, Suite 612
Boston, Massachusetts 02109
Phone: 617-742-5200
Fax: 617-742-1028

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	$495	$507	$519
☐ *Paradigm Shift—Patricia Seybold's Guide to the Information Revolution*	12 issues & tapes per year	$395	$407	$419
☐ *Paradigm Shift—Patricia Seybold's Guide to the Information Revolution*	12 issues per year	$295	$307	$319

Please send me a sample of: ☐ *Network Monitor* ☐ *Office Computing Report* ☐ *Unix in the Office* ☐ *Paradigm Shift—Patricia Seybold's Guide to the Information Revolution*

Please send me information on: ☐ Consulting ☐ Special Reports ☐ Conferences

☐ **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 20-093-118-6. Please be sure to identify the name of the subscriber and nature of the order if funds are transferred bank-to-bank.

IU-0890

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 1989/1990:

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

Office Computing Report

1989—Volume 12

#	Date	Title
11	Nov.	Cease Fire!—PC Wars Confuse Customers Needlessly
12	Dec.	IBM's Data Management Design—The Future Is Distributed and Relational

1990—Volume 13

#	Date	Title
1	Jan.	Office Redefined—A New Model for a New Decade
2	Feb.	Visual Programming—Is a Picture Worth 1,000 Lines of Code?
3	Mar.	Word Processing in the '90s—How MS Word for Windows Stacks Up
4	Apr.	Multimedia—The Promise, the Players, the Issues
5	May	Is X the Window? X Window and distributed Applications
6	Jun.	HP's Office Strategy—Extending NewWave Beyond the Desktop
7	July	Lotus Notes—A Platform For Developing Workgroup Applications.

UNIX in the Office

1989—Volume 4

#	Date	Title
12	Dec	Targon Office—Nixdorf's Object-oriented Platform for Application Integration

1990—Volume 5

#	Date	Title
1	Jan.	Oracle's RDBMS—The Product behind the Marketing
2	Feb.	Multiprocessing—Towards Transparent Distributed Computing and Parallel Processing
3	Mar.	IBM's Second Generation RISC—With its new RISC System/6000, IBM claims center stage.
4	Apr.	Hewlett-Packard's NewWave Office—Distributed Object Management in an Open Environment
5	May	The Informix RDBMS—Exploring Paths Less Traveled
6	Jun.	Desktop Unix—Open Desktop Aims to Be the Rallying Point for ISVs
7	July	A/UX 2.0—Apple Targets the Unix Desktop

Network Monitor

1989—Volume 4

#	Date	Title
11	Nov.	A New Calling—IBM's Strategy to Integrate Voice and Data
12	Dec.	TCP/IP Network Management—What It Was, What It Is, What It Will Be

1990—Volume 5

#	Date	Title
1	Jan.	Distributed Computing Environment—Interoperability Now!
2	Feb.	LAN Manager 2.0—Jumping Up to the Major Leagues
3	Mar.	A New View of OpenView—HP's Network Management Strategy
4	Apr.	A Study in Contrasts—Unisys's Open Networking Strategy
5	May	Xerox Panos—Delivering on Distributed Networking
6	Jun.	Soft•Switch—Blazing the Trail for Mail in Distributed Computing Environments
7	July	OSF's DCE—Solutions for Heterogeneous Interoperability