KBN

Patricia Seybold's

UNIX IN THE OFFICE

VOL. 4, NO. 4
ISSN: 0887-3054
APRIL 1989

INSIDE

PRODUCTS • TRENDS • ISSUES • ANALYSIS

CMU's Andrew Project

A Prototype for Distributed, Object-Oriented Computing

By Laure Brown

YOU'VE HEARD IT before, how Distributed Network Computing (DNC) and object orientation are shaping the next generation of computer systems. And while there are a lot of people who proselytize the advantages of these system paradigms, few would come up with the same definition for a distributed—and especially an object-oriented—*(continued on page 3)*

• E D I T O R I A L •

Unix: The Desktop X-Factor

By Judith S. Hurwitz

THE DESKTOP computer market is in disarray. On one hand, PC applications developers are clamoring for a 32-bit version of OS/2. On the other hand, they are still figuring out how to deal with the complexities of plain ol' 16-bit OS/2. They've begun by porting their existing DOS applications to OS/2 to gain some experience. Many are still in shock. OS/2 looks a lot more like Unix than DOS. It is, therefore, not surprising that these developers are looking longingly backwards and demanding that something be done to extend the life of DOS—such as breaking the 640K barrier and adding more multitasking features. Suddenly, DOS doesn't seem so bad after all.

A PLOT AGAINST OS/2. Things would probably continue to travel along this convoluted path if not for the X-factor—in this case, Unix/386. There are so many similarities between OS/2 and Unix that it was natural for those trying to win the desktop market to try to exploit the X-factor. Key among these vendors is Digital Equipment. Digital is savvy enough to realize that whoever owns the desktop has the opportunity to capture the hearts and minds of the users. Therefore, behind the scenes, Digital has been plotting the overthrow of OS/2. (We expect that Digital would vehemently deny this.) Yes, of course, Digital intends to fully support OS/2, if and when it becomes popular. It has its strategic partnership with Tandy ready to provide that piece of the pie.

OPEN DESKTOP. What's the tactic? Open Desktop. Open Desktop promises a robust desktop computing environment based on the Intel 80386 microprocessor, which will incorporate a graphical user interface, provide access to DOS applications, and include a relational database engine.

Those companies intent on supporting Open Desktop will point to its 32-bit operating system, its graphical user interface, its built-in relational database management system, and its support of DOS through Locus as evidence that Unix already has what OS/2 will not have for another year.

BUT WILL IT WORK? Maybe. The possibility that OS/2 might be knocked off center stage has certainly jolted that enfant terrible, Bill Gates. We were amused by an interview with Gates in a recent issue of *Computer Decisions* magazine in which he tried to clarify the difference between OS/2 and Unix this way: "The difference between OS/2 and Unix is that you can buy an OS/2 application off the shelf. Unix is just a word...." (How many letters are there in Unix?)

The difference between other Unix environments and Open Desktop is that, despite what our friend Gates says, we could begin to see shrink-wrapped Unix applications on the shelves. The pricing of Open Desktop and the number of vendors that intend to port their applications to it may indeed lead in this direction. There will be Application Binary Interfaces (ABIs) for Unix that could have the effect that the DOS ABI had some five years ago.

Ironically, OS/2 and Unix suffer from the same problem. Users mistrust both of these complex operating systems. Both require a large amount of memory and power on the desktop, and both require that traditional PC developers make a major transition. So technical issues may not be at the core of the battle for the desktop. It may well be that the politics of control, in the end, will decide if, indeed, Unix is a four-letter word.

Patricia Seybold's Office Computing Group

Publisher PATRICIA B. SEYBOLD
Editor-in-Chief JUDITH S. HURWITZ
Managing Editor RONNI T. MARSHAK
News Editor DAVID S. MARSHAK
Senior Editors MICHAEL D. MILLIKIN, JUDITH R. DAVIS Telephone: (617) 861-3926
Contributing Editor GARY J. NUTT Telephone: (303) 444-0341
Associate Editors JOHN N. KELLY, JOHN RYMER, LAURE BROWN
Sales Director RICHARD ALLSBROOK JR.
Circulation Manager DEBORAH A. HAY
Customer Service Manager ANNE L. WORKMAN

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

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

• ANDREW •

(continued from page 1) environment (hence our recent forum: "Object Orientation: Defining the End-User Platform for the 1990s").

At Carnegie-Mellon University (CMU), people have been defining, developing, refining, and using these technologies—in the form of the Andrew project—for seven years. Although Andrew is sponsored by IBM, its life in the university has given its developers freedom (though maybe not carte blanche) for creativity. Thus, Andrew has been a prelude of sorts to object orientation and distributed networks.

Andrew's use of Unix as its foundation makes sense, since Unix built its roots in academic communities. Given its scientific affiliations, it's no small wonder that Unix has opened the way to some pioneer technologies. Now those technologies are making their way into the office (witness, for example, MIT's X-Window). Although it remains to be seen to what extent the solutions found through the Andrew project will translate into commercial products, university research often points the way for the industry. CMU has already developed a ripening technology. We can only assume that the commercial world will take its cue.

If the toolkit and the file system are the meat of the Andrew project, the message system is the gravy.

Background

THE ITC. The Information Technology Center (ITC), an IBM-funded research and development laboratory at CMU, opened its doors in 1982. The 30-member center pooled both IBM and Carnegie-Mellon employees to design and develop computing technology for the university. So far, IBM has invested about $30 million in the center's activities.

Enter Andrew. The foundation of the work being done at the ITC is the Andrew Project. (The project, incidentally, was named after CMU benefactors Andrew Carnegie and Andrew Mellon). In 1982, the center set out with some assumptions about the kind of system that universities—as well as businesses—would need in the next five years: a consistent, graphical user interface; a distributed file system; and multimedia mail (and bulletin boards).

Among the initial thrusts of the ITC were the developments of the Andrew Toolkit (ATK) and the Andrew File System (AFS). The toolkit is a graphical programming environment with an embedded structure, and ITC pours substantial resources into its refinement.

When the ITC began work on AFS, no large-scale distributed file systems were yet in place (or even under development). The center began with the notion of a single, shared file system with a single interface. Today, AFS supports close to 9,000 users and 1,000 workstations at CMU.

If the toolkit and the file system are the meat of the Andrew project, the message system is the gravy. The Andrew Message System (AMS) was initially intended as a showcase to demonstrate the usefulness of the toolkit and file system. The system inherited the toolkit's multimedia characteristics. You can send a multimedia document that retains its life: The recipient of an AMS document can edit it, create graphics, scan images, rework spreadsheets—in sum, use every option the author had. It also caters to the file system's distributed nature as well as its support for non-Unix machines.

A System by Any Other Name. You might be expecting to hear a lot about Vice, Venus, and Virtue in this article. Well, don't. There's been a shuffling of terms at the ITC. What once was Vice is now the Andrew File System, or AFS. (We suppose the acronym does sound more file system-ish. Besides, we were reminded by a systems designer at the ITC, AFS comes before NFS alphabetically, and who's to say that Andrew won't benefit by subliminal influences?) Venus always played a strategic role in Vice. It still plays a strategic role in AFS, but now it goes by a name more functional, if less seductive: the cache manager. Likewise, Virtue is now just referred to as Andrew's user interface. So we'll be talking about AFS, the cache manager, and the user interface throughout this article, not Vice, Venus, and Virtue.

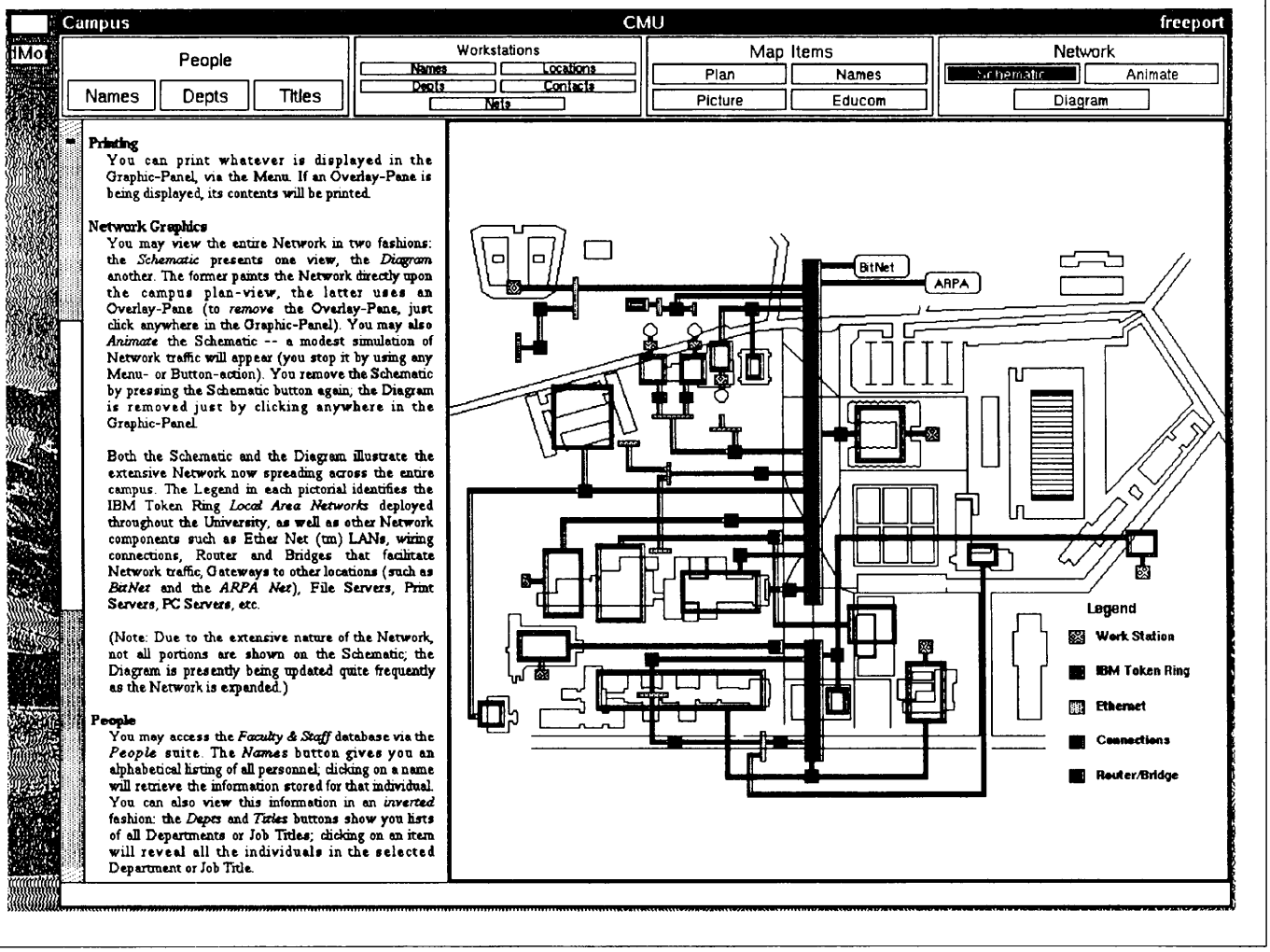
The Toolkit

The Andrew Toolkit (ATK) is an object-oriented, window-based environment for developing applications with flexible user interfaces. It's intended to let developers take advantage of the graphics capabilities of workstations. And that it does. ATK includes a set of sophisticated components, such as multifont text, raster images, and even animation (albeit simple animation).

The toolkit is extendible. Developers can create new objects that pop into those that already exist. Likewise, they can embed existing objects to create new applications without a detailed knowledge of what's being embedded.

Two window systems are supported by ATK: X.11 and the original Andrew window system with tiled windows. But ATK is easily ported to others (an OS/2 and Presentation Manager port are rumored to be in the works). Actually, it is conceptually window-system independent and, for that matter, operating-system independent. That way, applications are consistent, no matter what they're running on.

The ITC has also implemented a well-defined ASCII data stream for objects so that they can be saved as files and—this is



Sample Andrew screen.

a biggie—transmitted through the mail. Andrew's message system uses this ability for its multimedia mail.

IMPLICATIONS. Clearly, IBM and the ITC have commercial hopes for ATK. It's still considered an experimental technology, but that doesn't mean it won't impact the commercial market. Vendors may not be banging down doors trying to get it, but they are excited about it.

The Open Software Foundation (OSF) is courting the idea of including ATK in its research program. The toolkit would serve as a research vehicle for the development of future commercial systems. To OSF, the ITC fills the familiar university role of developer and forecaster in the evolution of new technology from which the vendor community can learn. OSF anticipates that parts of ATK will be absorbed into the commercial market—particularly its multimedia capabilities and its ability to transmit live objects (e.g., when you send a document to somebody, you transmit not just the page, but the underlying structure, too). While we've begun to see products with multimedia elements (e.g., IBM's MO:DCA and Digital's CDA), a set of interchange standards needs to evolve before we'll see live objects being zapped through many message systems (see "Life for Compound Documents in a Multivendor World," page 6). Funding from OSF would go into distribution and support of ATK or into providing it to OSF members. Meanwhile, the toolkit is available on MIT's X tape with the same licensing status as X-Window.

OBJECTS AS BUILDING BLOCKS. The Andrew Toolkit is a framework for building and combining objects. It has an embedded structure that, in addition to a set of basic components such as buttons, sliders, and composition components, includes six higher-level components: multifont text, tables (spreadsheets), equations, drawings, rasters, and animation. These can be embedded into each other in an assortment of combinations. It's a little like the picture on the Morton Salt box: a girl carrying a Morton Salt box that shows a picture of a girl carrying a Morton Salt box, and so on. With ATK, you can

create a table object that contains a text object that contains an image object... (you get the picture).

ATK objects are cooperative. The toolkit is based on a minimal protocol that lets objects talk to each other about user interface issues, so developers can devote themselves to the actual interaction between objects.

EXTENDING THE TOOLKIT. At the heart of ATK is the Andrew Class System (Class), which invests the toolkit with its object-oriented, embedded nature. Herein lies ATK's extendibility. Class has a dynamic load-and-link facility that allows developers to build new objects out of existing ones. Objects are strung together instead of being rebuilt. In addition, the ITC has come up with an interpretive language called "Ness" that lets even patient nontechies extend the toolkit without using C code. And that's the point. The ITC didn't want to be at the beck and call of every department at CMU. If a member of the mechanical engineering department wants to include circuit diagrams in an object, he can do it himself.

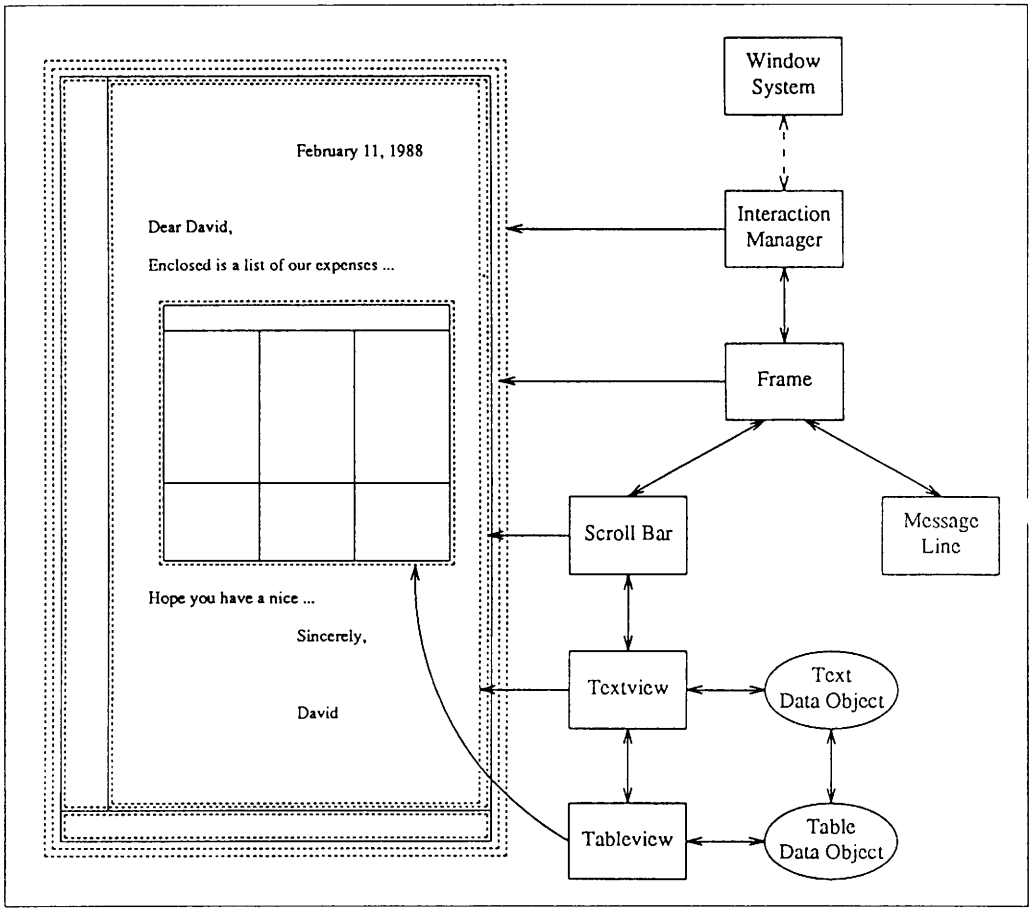
The Class System is a C-based, object-oriented preprocessor. It's modeled after C++. Apparently, C++ didn't quite offer the capabilities the ITC was looking for, like dynamic loading and sufficient object/source similarity to make debugging easy. Besides, C++ requires a licensing agreement, and the ITC didn't want to develop a product that required another licensed product.

With Class, The ITC has concocted a number of features for Andrew's user interface. Andrew's multimedia editor, for one, was built using the dynamic load facility. It's actually a meta-editor and loads the appropriate object editor code when needed. When a music component, for instance, is embedded into text, it is dynamically loaded into the application, and the user is virtually unaware that the music component was not part of the original editor. The Class system also had a hand in other Andrew interface applications, such as the Help system (Andrew offers both context-sensitive Help and a general Help index); a console, or status window, that displays things like date and time, CPU load, mail notification, and file system information; and the Andrew message system.

POINT OF VIEW. ATK offers developers some versatility by providing the ability to have multiple, simultaneous views of a single object. For instance, you can look at a single text object as a document, an outline, or a table of contents. You can even have two views of one object in a single window. Tables and charts are the obvious examples here. When you tinker with a spreadsheet, you can have a corresponding chart view that automatically reflects your changes.

This multiple-view scenario calls for a special design. There must be multiple ways to present a component with the same underlying data. To do this, ATK separates components into data objects and views. The data object contains the actual information (e.g., the very characters) of an object, while the view contains information about how the data is displayed and how the user manipulates it.

Separating the data from the view allows data objects to be used differently from the way they were originally conceived. Developers can create new views of existing data objects, and



A view tree for a window that contains a scrollable text view that contains a table view. The text view is surrounded by a scrollbar, which is surrounded by a frame (which also provides a message line). Thus, this particular tree has five generations: the interaction manager, the frame, the scrollbar (along with its sibling, the message line), the text view, and the table view.

Life for Compound Documents in a Multivendor World

THINK FOR A MINUTE about the concept of a living document—not merely a compound document, mind you, but a compound document that keeps its functionality even when it's sent across a multivendor system. Interesting prospect. In it is the idea of compound documents as works in progress. When you're working on an important business report, you usually go through a few stages: gathering information, collaborating ideas with other people, getting the report reviewed and approved, revising it, prettying it up, etc. As compound documents and computer-supported cooperative work continue to flourish, users will want to exchange multimedia objects, even among diverse systems.

YOU CAN'T GET EVERYTHING YOU WANT. However, the industry isn't ready to accommodate those users. In its way is a lack of standards. ODA is one of the industry's most promising vehicles for compound document exchange, but, as yet, it's not suitable. ODA supports only a subset of functionality for most compound document/object architectures. For instance, it doesn't handle spreadsheets or equations, and its style-sheet mechanism isn't as flexible as it could be.

LOOKING FOR SOLUTIONS: THE EXPRES PROJECT. In 1986, the National Science Foundation (NSF) began the Expres project, which sought to enable the exchange of complex information across multivendor systems. The players: the Andrew system at Carnegie-Mellon's Information Technology Center and the Diamond system at the University of Michigan (UM). (The Diamond system was developed by BBN Systems and Technologies and is now marketed under the name "Slate." For a review, see Vol. 3, No. 8.) Andrew and Diamond generate similar-looking compound documents, but their underlying structures don't come close. Hence, the universities began work on a common exchange solution.

The ODA Toolkit. A by-product of Expres is the ODA toolkit. Despite its limitations, UM and CMU have a lot of faith in ODA's layout semantics. (Cited as a major drawback is ODA's weak style-sheet device. Hopefully, the work going on with the toolkit will catalyze some improvements there.)

The toolkit makes it easier to work with ODA by providing a machine-independent collection of functions for conversion. Translators map specific document representations to and from ODA, which becomes an intermediate format. So, if you're sending a Diamond document (.DMD) to an Andrew (.ATK) user, it's converted first to ODA, and then from ODA to .ATK. Some attributes are slightly readjusted in the process (ODA is less flexible than either .DMD or .ATK). So far, the toolkit has been used to convert Interleaf, troff, Diamond, and Andrew documents. It will be interesting to see exactly how much information gets lost—or readjusted—once more translators are developed.

Translation Overload. If the ODA toolkit is as successful as its developers hope and a whole spray of translators evolve, what might happen to a document that's been run through several systems with varying capabilities? Two things. One is that it will be reduced to a common document that all the systems could deal with. The extreme in this case is pretty bleak; the file could be reduced to nothing as each system trashes the information it can't support. The opposite extreme is scary, too. The document could become completely overladen with redundant information.

Time will tell.

Next Steps. The ITC would like to see the ODA toolkit become public domain. One result the ITC would really like is the formation of an ODA Consortium with the double purpose of continuing to extend ODA so it supports more kinds of documents and making the toolkit available to consortium members.

existing data objects can be used as building blocks for more complex objects.

The View Tree. An ATK component relies on its relationships to other components, both the ones it contains and the ones that contain it. The toolkit structures these relationships in a view tree. At the top of the tree is the interaction manager, which can have a long line of descendants. It all depends on the depth of the object in question. You can consider the tree as a chain of responsibility, and the interaction manager as a delegator. The interaction manager never deals with an event itself; it merely accepts events (e.g., keystrokes or mouse clicks), then passes them off to the child view. The child view may be able to process an event itself. If not, it passes the event down to one of its own children, which may hand it down further.

Here's a scenario: Someone is manipulating an image that's contained in a text object that also contains a table. The interaction manager says, "Aha! Data is being manipulated in

that text object; its view needs changing," and alerts its child, the text view. The text view, in turn, looks at its two children, the table view and the image view, and decides to let the image view process the event.

The Graphics Layer. The ITC created a separate object that provides the graphic interface for views. It also provides a window system-independent layer and default printer support. However, the current graphics layer won't let you get to the underlying window system.

APPLICATIONS. We thought it a good idea to give you a peek at the kinds of applications ATK produces. The ITC seemed particularly proud of the creation of a piano, which was built with sub-objects such as switches, sliders, buttons, and text. On screen, it looks like a scale of piano keys. As you slide the mouse pointer over the keys, they'll make music. The piano captures the notes of any tune you might compose and replays it at your bidding. Or you can put it in another application, or even send the piano and tune electronically to be modified by someone else. (The piano is posted on a CMU bulletin board.) Pretty neat, huh?

On the more practical side are the user interface tools (some of which we've already mentioned, such as the editor). Among them is an experimental Unix shell called "Bush" that gives you a tree-like view of Unix directory structures.

The applications aren't all ITC-developed. For instance, members of the history department came up with a program that contains loads of historical data and various ways of viewing it. You can, for example, look up the teen-age, female population of Cork, Ireland, before the potato famine. Once you get it, you can view the demography on a map: scale it, size it, zoom in, and compare it to the demography after the famine.

AFS doesn't support diskless machines. This will be a stumbling block in AFS's route to commercial viability.

The File System

When CMU started the Andrew project, the campus had lots of computing going on with different machines and an incredible need for communication (sound familiar?). Thus, the Andrew File System (AFS) was designed to support a number of diverse machines on a single file system.

To the user, AFS looks a lot like the Unix file system. It also attaches simpler file systems for simpler machines (like Macintoshes or PCs); to these, AFS looks like an extension of the native system. The user views the same set of files no matter what machine he or she is working on—the MicroVax on the desk or the RT down the hall. AFS is a single shared file tree that is distributed across a group of servers. But the user is spared the intricacies of its distributed implementation (e.g., the user doesn't have to keep track of file locations).

The AFS network is based on TCP/IP and made up of Ethernets and Token-Rings tied together by bridges. Although other protocols are used on various network segments at CMU, the routers operate at the IP packet level. Any workstation on the network has direct access to AFS. Lower-powered machines, such as PCs and dumb terminals, hook up to the system via a set of machines called Unix servers, which are reserved explicitly for this task.

THE DESIGN. Crucial to AFS is workstation-caching. When you retrieve a file from the server, it is cached—in its entirety—on your hard disk. If you find yourself with a huge file, AFS will automatically read it in chunks. This disk-caching principle sets AFS apart from the usual record-based schemes. File server traffic is reduced because, since users have most of their data cached locally, they don't have to keep nagging the central file system. Furthermore, the design gives the computer on your desk a high degree of autonomy because files are cached and reused locally.

But not if you have a diskless computer on your desk. AFS doesn't support diskless machines, and lots of networked computer users in the corporate world have them. This will be a stumbling block in AFS's route to commercial viability.

At CMU, AFS is distributed among approximately 30 servers with about 1,000 client machines. The servers are relatively small: Sun3/260s and RTs. Each is equipped with three to six 400-megabyte disks. The distributed, small-server design has advantages for the university's computing needs: It's easily expandable; it's dependable (a single server might crash, but it won't affect the others); and it simplifies the development process by using the same hardware and operating system as the workstations. However, no one is dismissing the use of larger, more centralized servers. In fact, AFS is being adapted for the MVS operating system (see "Commercial Aspirations" below).

Communication. AFS uses remote procedure calls (RPCs) for all interprocess communications. The RPC mechanism, called Rx, uses streams to transfer entire files and 64KB file chunks between file servers and workstations.

CACHE CONSISTENCY. AFS's cache-based, distributed design demands a solid, file-consistency solution. Again, AFS developers were concerned with reducing server and network loads. To do this, the system uses callbacks. Instead of constant version-checking messages being sent to the server, the server lets you know when a cached file has been modified. AFS offers open/close granularity for shared files. So a user on one

workstation doesn't see the changes written at another workstation until the other file is closed and an open system call is executed.

As far as overwriting files goes, you're working with Unix; the last writer wins. However, file locks are available to keep disreputable (or just plain half-witted) coworkers from overwriting files.

CACHE MANAGER. The Andrew cache manager, the crux of AFS/client activities, lives on client workstations and interfaces to the Unix kernel via Sun's vnode interface. Basically, the cache manager is in charge of finding and retrieving files from the server. Thus, users don't need to know the location of a file; they just need to ask for it. The cache manager takes care of server communications and stores the file locally. Likewise, it routes updated files back to the server.

Users don't need to know the location of a file; they just need to ask for it. The cache manager takes care of server communications and stores the file locally.

The cache manager recognizes a file on servers by its fid (file identifier) and version number. In order to cut down on processor time, Andrew developers decided to free file servers from translating pathnames to fids. Instead, the cache manager keeps track of pathnames. Additional responsibilities of the cache manager include:

- Keeping copies of recently used files on the local disk
- Reproducing Unix system-call semantics
- Freeing cache space when it becomes full
- Performing RPCs to the file servers
- Caching directories and file status information
- Maintaining user authentication tickets (see "Security" below)
- Keeping track of the user's various IDs (see "Wide Area File System" below)

SECURITY. Security is a critical issue at CMU. You can't really trust your workstation (especially in a university setting where, more often than not, you're sharing it). Furthermore, a good many workstations and PCs at CMU have access to the network, so sending passwords over the network wasn't an option.

Authentication. The ITC scrapped its original authentication system and adopted one produced at MIT called Kerberos (developed for Project Athena). In essence, the system won't recognize you unless you answer its encrypted message. (The messages, incidentally, are unique. They're never repeated—an added measure against would-be data thieves.) Once you answer, the system checks your identification and hands off a ticket to the cache manager, which provides access to the system for 25 hours. After that, you log in again.

Access Lists. AFS uses access control lists to specify the class of operation for each user (or group of users). The seven classes are: read, write, insert, delete, lookup, lock, and administer (i.e., change the access list).

LOGICAL VOLUMES. To make life easier for the systems administrator, AFS groups files into "logical volumes." Typically, a volume would be a single user's files (although the end user is really not aware of it). That way, systems administrators deal with volumes rather than individual files for operations like backup and load-balancing between file servers. The cache manager is responsible for gathering volumes into a coherent file tree.

Clones. One luxury of distributed architectures is resiliency. It's okay if a server fails because files are copied on multiple servers. Read-only versions of volumes, called clones, are replicated throughout the Andrew file system. Clones serve a number of purposes. For example, new software releases are made by cloning the system binaries. At CMU, each user's logical volume is cloned every midnight for backup the next day. Users can find these volumes under the name "OldFiles"; thus, the previous day's work is available without having to resort to backup tapes.

PERFORMANCE. Allow us an anecdote here. Our trip to Pittsburgh—the actual plane ride—was fun ... and revealing. One mention of the word "Andrew" to the passenger in the window seat, and the whole planeful of passengers (well, okay, maybe just the first four rows or so) was buzzing about it. Anybody even remotely connected with CMU (even someone associated with Pitt) had heard of Andrew and had something to say about it—good and bad. We were intrigued. Anything that generated so much energy must be worthwhile.

One grievance repeated now and again during the flight was that Andrew was slow, especially at peak hours—say between 1:00 and 2:00 p.m. Well, we didn't see any applications until about 4:00, but, yes, the system was sluggish. Caching puts reins on network congestion, but there will still be periods of heavy traffic on a system that supports 1,000 client machines on a very large number of internetworked LANs.

WIDE AREA FILE SYSTEM. The ITC is in the midst of an

AFS-based, wide-area file system experiment. A wide area file system is just what it sounds like: a large-scale, shared, distributed file system. It's intended to let individual organizations share files while letting them also maintain independent control and data protection. Transparency is the goal. Ideally, you could be in New Jersey sharing files with a site in Montana (or even Hawaii, for that matter), and be completely oblivious to the network's existence. In other words, it isn't necessary to explicitly name the computer on which a particular file is stored (as it is with DECnet, for instance).

A wide area file system seems a logical next step from the advances that have been made in distributed network computing (DNC) and single-organization systems (such as NFS, Locus, DS, and, obviously, AFS). It's also a natural extension of the network services around today, such as remote login, file transfer, and E-mail.

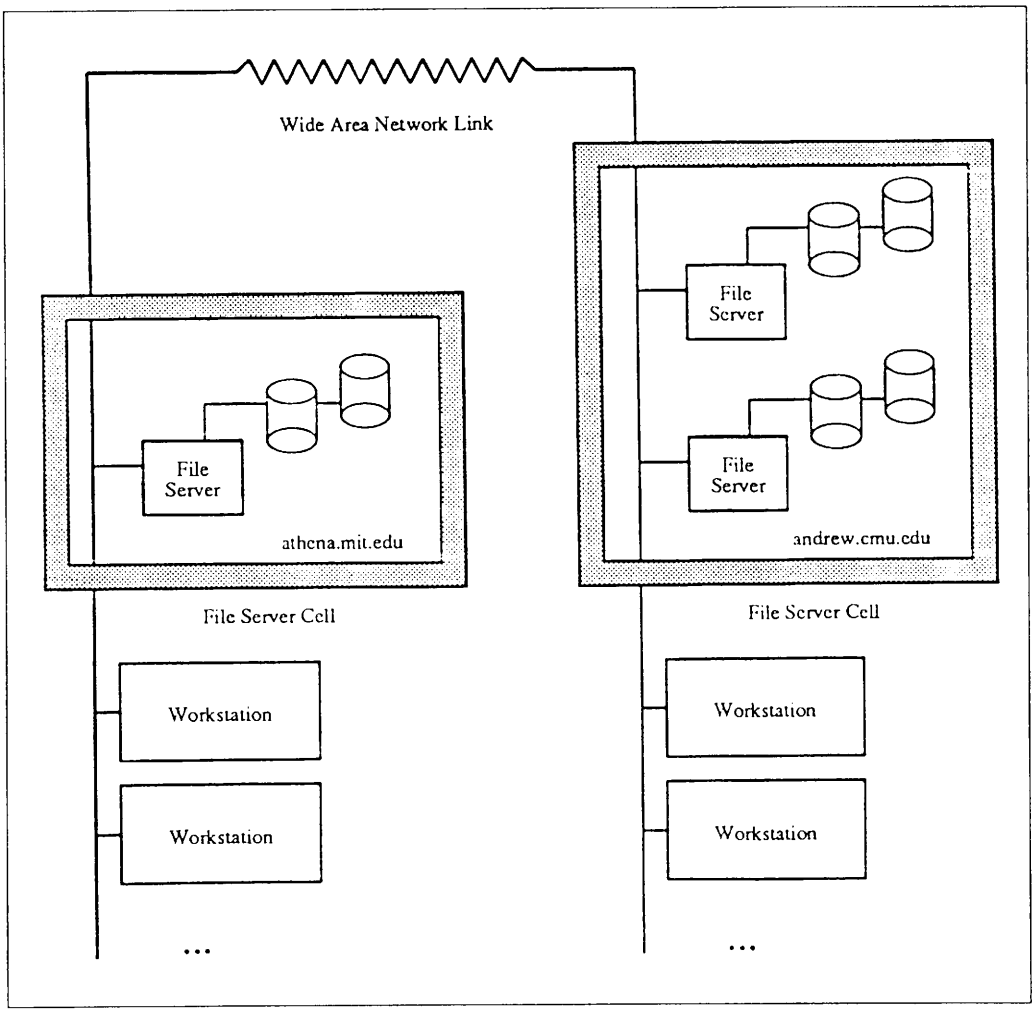
The project has two main purposes: to determine the advantages of a wide area file system over existing network services and to put the power of AFS to the test. We're talking about a system that is capable of supporting thousands of workstations across thousands of miles. In the end, the ITC expects as many as 20 sites will be using the file system. Right now, there are half a dozen experimental sites: Dartmouth, the National Institutes of Health, Mt. Xinu (Berkeley, California), MIT, the University of Michigan, and CMU.

Implementation. The system is AFS. It has everything we've been describing: caches, callbacks, logical volumes, access control lists—you name it. The wide area AFS is a connection of cooperative cells. (Cells are groups of file servers and clients). The cache manager knows which volumes belong to which cell. It also remembers all the user's IDs (which may vary according to cell) and gives the user appropriate connections. As long as the user has permission, a workstation in one cell can access files—climb the file tree, so to speak—of another (see illustration).

The system inherently requires a speedy WAN. NSFNet is the backbone, which runs at 1.5 Mbps, and RPCs should also speed things up. Speed will be an issue. AFS is already fairly slow at fetching files from servers. If retrieving a remote file is annoyingly delayed, no one will bother.

COMMERCIAL ASPIRATIONS. IBM would certainly like to make a product out of AFS, which isn't to say that the company has no other options. IBM supports Sun's NFS (Network File System) and its own DS (Distributed Services) system. DS is impressive (see Vol. 3, No. 4), but it's proprietary (at least at the moment. OSF could make it part of its operating system). NFS is open, but it belongs to Sun. Enter AFS. It's open, it has benefits over NFS (e.g., the AFS authentication and disk-caching mechanisms), and IBM can leverage it as an IBM product. AIX will be the next implementation.

An MVS version is also in the offing. Granted, we've been talking a lot about AFS's distributed nature. But there's no reason its caching and callback mechanisms can't be applied to big, centralized machines. In fact, that's how it's being implemented for a project at the University of Michigan. UM plans



AFS cells on a wide area network. If access is permitted, the workstations on the Andrew cell can retrieve files on the MIT cell.

on using 3090 mainframe servers to create a system that supports 30,000 workstations. How well will a Unix-conceived file system adapt to MVS systems? The environments are worlds apart. We'll wait and see.

In Light of NFS. The basic premises behind AFS and NFS don't exactly mesh. The goal of AFS is to reduce communication, which must seem almost sacrilegious to a company that advertises "the network is the computer." All the same, NFS is the de facto standard, and Andrew developers aren't ignoring it.

The AFS cache manager can coexist with an NFS implementation so workstations can access both NFS and AFS servers. Additionally, the CMU computer science department has written a server that exports AFS files using NFS protocols. Thus, NFS clients can have wide area file service via an intermediate machine in their LAN (although the security provided in this scenario will be the least common denominator).

But NFS isn't the only file system out there. Somewhere down the line, the ITC would like to construct an export server to let arbitrary distributed file systems export files to AFS.

Multimedia Messages

The Andrew Message System (AMS), an end-user application, the fruit of the Andrew toolkit and file system, is definitely not your average E-mail. Users can send even complex compound documents—with all the elements that distinguish the toolkit—electronically. How much of that document gets lost in transition depends on the sophistication of the machine at the other end. If you're sending it to another workstation, it's great. The document actually stays live. When you send it to a text-only destination, AMS creates a no-frills version with text descriptions of the other media elements (see illustration below).

AMS also supports multimedia bulletin boards. At CMU, it supports over 1,800 bulletin boards, including netnews, the Dow Jones information service broadtape, and all sorts of messages (fun, informational, off-beat) from the user community. Bulletin board magazines can be set up to weed out the tangle of bulletin board messages on specific topics. The mail and bulletin boards share a common interface. A user who wants to keep a message from a bulletin board simply puts it in his or her own mail folder.

AMS STRUCTURE. The Andrew file system and toolkit both shaped the design of AMS. Without the toolkit, the system simply wouldn't generate multimedia messages. But those multimedia messages take up a lot of room. The message databases of AMS are hefty (gigabytes of message text), but the distributed file system accommodates them and simplifies accessing them. The message database is in one place—the file system. Furthermore, AFS's authentication feature combats message forgeries and permits private bulletin boards and shared mailboxes.

AMS is divided into three layers of functionality: the application layer, the client library layer, and the message server layer (see illustration on page 11). Users deal only with AMS at the application level; client library and message server activities take place in the background.

Thus, where a Messages user might see something like this:

Date: Wed, 25 Nov 87 10:52:30 -0500 (EST)
From: Nathaniel Borenstein <nsb+@andrew.cmu.edu>
Subject: Animated logo!

Here's the CMU logo:

But the *best* part is that if you click on it and choose the **Animate** menu, it will turn into a cube and tumble around on your screen!

Isn't that neat?

A user of a lower-functionality interface would instead see something like this:

Date: Wed, 25 Nov 87 10:52:30 -0500 (EST)
From: Nathaniel Borenstein <nsb+@andrew.cmu.edu>
Subject: Animated logo!

Here's the CMU logo:

[An Andrew/BE2 view (an animated drawing) was include here, but could not be displayed.]

But the best part is that if you click on it and choose the Animate menu, it will turn into a cube and tumble around on your screen!

Isn't that neat?

Application Layer. The application layer is used to create and retrieve mail messages. The interface of mail applications varies (naturally) according to machine type. AMS has three user interface levels. The most basic, CUI (common user interface), doesn't require anything more sophisticated than a teletype display. Visual user interface (VUI) is a notch above that—at PC level. Andrew's fancy multimedia interface, called Messages, is confined to workstations.

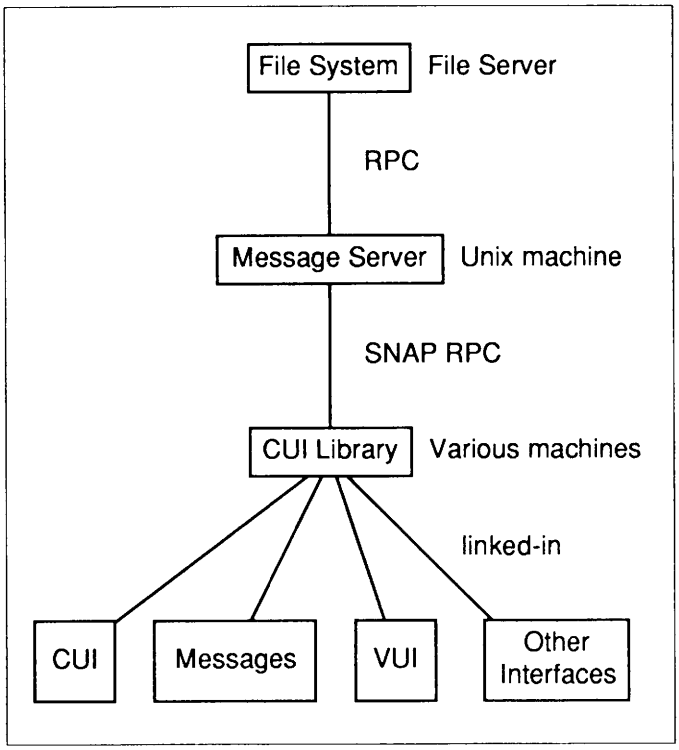
CUI Library. The client library, or CUI library, is the middleman between the application and the message

server. It implements various abstractions of client interfaces as well as communications with the message server. It's written in C and runs in Unix, VMS, DOS, and Macintosh.

Message Server. The message server actually accesses the message database (which is stored in AFS and holds all AMS messages and bulletin boards). Clients access the message server via an RPC called SNAP.

DELIVERY SYSTEM. The delivery system, logically, sends mail messages. Originally, the ITC hoped to simply use the standard Berkeley sendmail program, but the characteristics of AFS got in the way. Therefore, the AMS delivery system was entirely rewritten to take advantage of the file system's authentication system, its support of non-Unix machines, and its distributed nature.

Aside from the fact that it copes with AFS, the AMS delivery system has other advantages over standard Unix systems. For instance, it provides location independence. Senders don't have to know machine locations, just user names—and they don't even need to be sure of those. The delivery system has a user name look-up facility called "white pages" that makes allowances for misspelled names. So if you're not sure how to spell the name "Breitmeyer," the system will come up with some probabilities. The white pages database is stored in AFS and is accessed by the message server, which, in turn, exports it to the client interface.



AMS structure.

Commercial Ramifications: What Can We Learn?

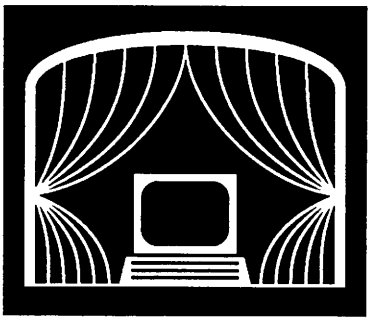
For starters, we know we need industry standards that will achieve the same goals as the Expres project: the exchange of processable objects among dissimilar hardware and software environments. The industry is starting to put out compound document/object architectures, but can they be transmitted among various systems? Nope. Before long, users will demand a common mechanism for exchange.

Interoperability is also a plug for the Andrew file system. Very well, Sun's NFS is open, too, and it's also the industry standard. But even Sun could learn a trick or two from AFS. AFS's disk-caching scheme makes the system a far better network decongestant than NFS. AFS's logical volumes also make things easier for administrators to configure and reconfigure the system. Moreover, AFS's authentication system provides better security than NFS, which depends on Unix access rights.

The commercial world will also be eyeing this year's AFS projects, namely the wide area file system and the MVS port.

The toolkit may be Andrew's first direct impact on the marketplace. Recently, there's been a lot of hoopla surrounding user interfaces. Tomorrow's standard user interface will be like Andrew's: graphical and object oriented. The recent spurt of compound document/object architectures and applications with graphical user interfaces makes the Andrew toolkit seem particularly pertinent—especially for the proliferation of Unix. The ITC maintains that, if Unix is to gain an edge on the PC world, it needs to provide a higher-quality user interface than PCs do. Furthermore, it needs to coexist with the office environments already in place. PCs and Macintoshes aren't going anywhere. The toolkit's flexibility in regard to window and operating-system independence can help bridge that PC/Unix gap.

•TRADE SHOW COVERAGE•



UniForum '89: Getting Serious

By Judith S. Hurwitz

With all of the momentum behind Unix this past year, we expected this year's UniForum to be a good show. We were not disappointed. We were also very impressed that almost 25,000 attendees showed up for this annual pilgrimage to Unix land.

In many ways, UniForum '89 was much like those of the past few years: It was filled with the promise of a vastly expanding base of Unix devotees and the continued increase of the business-suit count (only 4.2 ponytails were spotted). Each year, we've seen more and more traditional proprietary vendors showing signs of turning to Unix. And again, we were not disappointed. Almost all the traditional proprietary vendors were out in force.

What set this year's UniForum apart? For the first time, it did not have the feeling of a religious gathering, where preachers stand up to see how many converts are in the audience. And it did not have that naive sense of wonder ("look how many new friends we've made"). UniForum was all business—big business. Somewhat sadly, we noted that the spirit was different; it had that hard edge of sophistication that comes when a marketplace turns commercial.

SETTING THE FOUNDATIONS. While we keep hoping that next-generation applications will start to flood the floors of Unix conferences, we expect that we will have to wait until next year. This is not to denigrate this year's show. At UniForum '89, the foundation was set. You couldn't step anywhere without seeing graphical user interfaces, networked workstations, X-terminals, servers, and diagrams of complex networking schemes. It was an exhibition dominated by hardware. Intel proudly touted its new i860 microprocessor (formerly called the N-10), which, if all the claims are correct, will make the 80386 seem as slow as an 8086. Data General (DG) proudly announced that its new Aviion family of RISC systems had broken the $500-per-MIPS barrier. DG was so intent on showing the industry that it was dead serious about the Unix operating system that DeCastro himself could be seen on the show floor talking to customers.

IBM had its 3090s in its booth again, as it had at Unix Expo last November. The IBM booth was chock-full of applications running on RT/PCs. It was also demo-ing its Transparent Computing Facility (TCF), which will allow 370 family members to participate in a distributed network computing environment (see Vol. 3, No. 4 for a full description of TCF).

CONSORTIA ABOUND. Unix International (UI) was very visible at UniForum. In fact, these guys won the prize for the best conference giveaway (beach balls of the globe of the world—international, get it?). Each day, UI put up on a blackboard the names of companies that had joined. UI used the occasion to announce (to no one's surprise) that it had selected Open Look as its user interface. It did leave room for other user interfaces, however.

In contrast, the Open Software Foundation (OSF) was uncharacteristically quiet. It held no events at the show. OSF seemed to be laying back and enjoying the positive reception of its Motif user interface.

UniForum '89 was an event surrounded by consortia of all sorts and shapes. In order to pick up some momentum for its SPARC RISC chip, Sun formed a consortium called SPARC International. SPARC International intends to publish binary compatibility specifications and institute conformance-testing. As part of the announcement, Interactive Systems and Phoenix Technologies were chosen to direct the "open licensing of SunOS and the complete SPARC software development environment." Therefore, the licensing of SunOS will be handled by a neutral body that will work with AT&T's software development environment to develop an Application Binary Interface (ABI) for SPARC. The 88Open Consortium held an event to drum up support for the new Motorola RISC chip. 88Open also announced that it would work with AT&T to develop an ABI for the 88K chip.

USER INTERFACES. User interface was hot. The Open Desktop consortium was promising inexpensive and wide availability of applications based on a combination of OSF/Motif, Locus's DOS merge products, and the Ingres Relational Database Management System. At the same time, Sun Microsystems was making loud noises about the

Open Look user interface. To build momentum for Open Look, Sun promised to provide users with the Open Look toolkit at no cost. The company is hopeful that the more than 2,000 applications that exist within the SunView environment will be ported to Open Look.

(For more information on Open Desktop, see pages 2 and 16.)

APPLICATION SOFTWARE: HELP WANTED. There were very few new applications at UniForum. We were hoping to find lots of interesting applications tucked away in corners. We did see a few DOS packages that were ported to Unix and a lot of the old faithfuls like Uniplex and Applix (which we hear is in serious trouble). In fact, several Pick operating system-based applications and hardware vendors took advantage of this applications drought to offer Unix vendors and users access to the more than 3,000 existing Pick applications. Companies like The Ultimate Corporation are offering Pick-to-Unix software bridges that allow Pick applications to run under Unix System V.3. There were some new faces, as well. Among these were HCL America, the new American subsidiary of the largest computer vendor in India. HCL is hoping to take advantage of its extensive commercial Unix experience in the Indian market to penetrate the U.S. market.

CONCLUSION. The Unix industry is on a roll. This has been a turbulent year, and it doesn't look like the pace of change is going to stop. We do believe that the beginnings of commercial Unix are taking hold. If application developers begin to flock to Unix during this coming year, then the good times may continue into the new decade.

Patricia Seybold's Office Computing Group
Special Research Reports

Available
April, 1989

Compound Documents

VENDOR STRATEGIES, ARCHITECTURES, AND APPLICATIONS

By Michael D. Millikin

This special report explores the emergence of what will be a major application enabler in the coming years: compound documents.

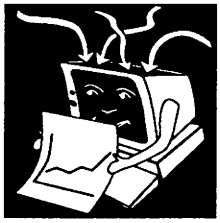
Compound documents are documents consisting of a variety of data types: text, image, and graphics, for example. In the context of a distributed network computing environment, compound documents become critical as a method of providing users with a transparent means to access, integrate, manipulate, and disseminate a variety of information stored on heterogeneous processors.

This report examines the four major defined document architectures:

- Digital's Compound Document Architecture (CDA)
- IBM's Mixed Object: Document Content Architecture (MO:DCA)
- ISO's Office Document Architecture (ODA)
- Wang's Wang Information Transfer Architecture (WITA)

We explore the functionality defined in each specification, and, where possible, give examples of sample applications.

Additionally, we will touch on the strategy for document architectures of two other vendors: Hewlett-Packard and Microsoft.



Available
June, 1989

Executive Support Systems

By Mickey Williamson

An Executive Support System is an important business tool that can integrate and summarize key information about a corporation and its competitive environment.

Unlike most other computer applications, these systems are adapted for justifiably impatient non-users who demand fast answers to questions about their companies and won't accept technical barriers to gathering those answers from different computers and databases.

This special report:

- Examines the opportunities and constraints faced by companies deciding to implement an ESS
- Compares in-house development, vendor-supplied software and consulting, and mixed development options
- Outlines the costs, time frames, and technologies, and the people who tend to be involved in successful ESS implementations
- Examines the leading ESS products and the technology trends that will determine the form and evolution of Executive Support Systems in the 1990s

OSF And UNIX International

SETTING THE OPEN SYSTEMS AGENDA

By Judith S. Hurwitz

The inception of the Open Software Foundation (OSF) and Unix International (UI) has changed the dynamics of the open systems industry for the future. These organizations are being influenced by the standards-setting work of X/Open.

This report will look at the origins, developments, and structure of these organizations as well as their plans for the future development of their environments. It will also examine the implications of each organization and the impact they will have on Unix and the standards movement as we move into the 1990s.

Among the highlights of this in-depth report are:

- User Interface: A close look at OSF's Motif and AT&T's Open Look
- Operating System Kernels: Design, implementation, and technologies
- Communications and Networking Foundations: How each group is building its networking underpinnings
- Applications Binary Interfaces: How OSF and UI will implement the ABIs and the implications for shrink-wrapped applications
- The Development Environment: Tools that will emerge for developing applications in these environments
- Administration, Backup, and Security: How these vital areas are being addressed
- X/Open and the Common Applications Environment: The impact that X/Open's portability guides will have on OSF and UI

Available May, 1989

ORDER FORM

☑ Please send me the research reports indicated below!

Qty:	Report:	Price:
____	Compound Documents	$495
____	Executive Support Systems	$495
____	OSF and Unix International	$495

Total Amount Due: _____

Form of Payment:

❑ Bill me P.O. # _____

❑ Check enclosed

❑ Charge to my: VISA/MasterCard/AMEX (circle one)

_____
Signature

_____
Card Number Exp. Date

Please print or type. Checks should be made payable to Patricia Seybold's Office Computing Group.

_____
Name

_____
Title

_____
Company/Dept.

_____
Address

_____
City, State, Zip

_____
Business Phone

UF89-SAMAD

Simply fill out the order form and send it to:

Patricia Seybold's
Office Computing Group
148 State St., Ste. 612
Boston, MA 02109

For information on multiple order discounts, contact Richard Allsbrook.

For *faster* service, call 1-800-826-2424 (in Mass. 617-742-5200) or fax 617-742-1028.

NEWS ANALYSIS

PRODUCTS • TRENDS • ISSUES • ANALYSIS

• INSIDE •

•OPEN DESKTOP•

An Assault on OS/2

A group of computer companies, including Digital, Relational Technology, The Santa Cruz Operation (SCO), Locus, and Tandy, has banded together to provide a desktop environment for low-end Unix. The result of this joint venture will be the packaging of a user-interface environment intended to propel Unix into the forefront of the battle for the desktop. If Open Desktop achieves its goals, it could change the dynamics of the low-end Unix market forever. Open Desktop is not a new technology; it is high-powered marketing and packaging.

Open Desktop bundles the following products:

- Unix System V/386 Release 3.2.
- OSF/Motif, the graphical user interface based on Digital's XUI user Application Programming Interface (API) that incorporates the behavior of Presentation Manager as developed under X-Window by Microsoft and Hewlett-Packard
- The X-Window system (Version 11 Release 3), based on SCO's XSight developed by Locus Computer.
- Ingres/386 (Release 6), the distributed SQL relational database from Relational Technology. Ingres/386 includes the standard structured query language, ANSI Level I SQL. Relational Technology will publish an open database communications interface called the General Communication Architect (GCA) so that third party developers can write to the API. With the GCA, database applications will be able to access the back-end database engine, enabling database developers to port their products to the open desktop environment.
- Communications, including TCP/IP and NFS. In addition, Open Desktop will be compatible with Microsoft's OS/2 LAN Manager and will support Ingres/Net so that networked database applications can operate transparently in a heterogeneous environment.
- DOS access, provided by Locus's Merge 386 which works with SCO's Xsight and Locus's PC-Interface server to access DOS applications inside the Open Desktop environment.

To help lure government users and contracts, Open Desktop will meet the C2 level of security as set by the National Computer Security Center. In addition, the product will conform to X/Open's Common Applications Environment (CAE), the IEEE portable operating system interface for Computing Environments, and the Federal Information Processing Standard 151 (POSIX FIPS). It will, of course, conform to the System V Interface Definition (SVID) and the ANSI X3J11 C programming language (ANSI C).

The core product, which will include the operating system, database, and OSF/Motif, will be sold at $995 for one or two users. Developer versions should ship in April, while end users will be able to buy Open Desktop during the third quarter of 1989. In addition, there will be an optional upgrade that includes the development system and API set as well as interactive development and debugging tools. Purchasing the individual modules that make up the core of Open Desktop separately would cost about four times as much. The pricing is clearly intended to make Open Desktop an attractive sale to a large number of developers and users.

The primary mover behind the Open Desktop product is Digital. It got the parties together and shaped the foundation. We see Open Desktop as a clear swipe at OS/2. Given the problems that have plagued OS/2, Unix looms larger and larger as an alternative. After all, isn't it a multitasking

operating system built for the 386 chip with a graphical user interface and a built in relational database engine? Might Open Desktop be an alternative to IBM's SAA? We have a hunch that this possibility could have been on Digital's mind.

Open Desktop could be a watershed event for commercial, low-end Unix. However, much of its success will depend on how quickly developers can create innovative applications to run in this environment. Open Desktop offers a legitimate alternative to OS/2 in the short run, while OS/2 struggles to move from its current 16-bit operating system status into one that takes advantage of the power of the 386. Over the next six months, we plan to watch the progress of this new marketing venture. A lot depends on how quickly Tandy, one of the sponsors of Open Desktop and the first to license it, can begin using it in its products. We think some interesting times could be ahead for Open Desktop. —*J. Hurwitz*

•INTERNATIONAL OUTLOOK•

Unix in Canada

The Canadian market lags a bit behind the U.S. market in its perception of Unix and open systems. In Canada, the term "open systems" usually conjures up a limited and partial image of networks—specifically, ISO's open systems interconnect model. However, that image is just starting to change, and, to provide a better angle on the status of Unix in Canada, the Canadian network of Unix systems users, /usr/group/cdn (Toronto, Ontario), sponsored a research project to investigate the country's Unix and open systems market. The project was conducted by the DMR Emerging Technologies Consulting Group, and its findings were published in a report entitled "Open Systems: The Unix in Canada Study."

The study has discerned "a fundamental change in the use of, and the market for, information technology. A new paradigm of computing is becoming apparent, based, in part, on open systems." In 1988, only 6 percent of industry hardware revenue was generated by Unix systems. The study identified an incredible lack of knowledge about Unix and open systems. Nevertheless, the study also found a widespread interest among information systems executives in learning more about the technology. DMR forecasts a 41 percent growth rate to 1992. That's significant, especially when you compare it to the 8 percent projected growth rate for the computer industry as a whole. In four years, Unix-based hardware should reach $880 million in revenues, or 19 percent of the market share. Clearly, Unix and open systems are gaining attention in Canada.

These statistics point to the Canadian market as a real gold mine for Unix vendors in the next few years. The "Unix in Canada" report has highlighted government, manufacturing, services, and education as the prime Unix targets. What makes these sectors so ripe for the picking? They are all facing similar system management situations and are considering adopting Unix to realign their information systems. The symptoms of these organizations' difficulties include:

- Increased diffusion of MIS functions
- More multivendor acquisitions
- Problems with systems compatibility
- Plans to replace existing information systems during the next three years

The study concludes with a number of recommendations both for IS executives, who will be implementing them, and for vendors, who are hoping to penetrate the Unix market. In order to monitor and influence developments in the Unix market, /usr/group/cdn plans to repeat the "Unix in Canada" study in 1989. The report is available to Unix in Canada subscribers. Subscriptions are priced at $12,500 (Canadian) and include the 550-page report, a private half-day seminar, access to the study, and copies of the "Unix in Canada Summary Report." —*L. Brown*

•NOVELL•

Unix Vendors Embrace NetWare

The integration of Unix systems and PC networks took a large step forward last month when Novell and a group of systems vendors announced the porting of Novell's popular NetWare operating system to Unix servers. The announcement, obviously a direct response to the planned Unix versions of LAN Manager being developed by Hewlett-Packard and AT&T, is part of a larger strategy by Novell to open up NetWare and license it to vendors on virtually all popular platforms.

Dubbed Portable NetWare, the open version can be licensed by hardware vendors for porting to their systems. Among the vendors announcing agreements to implement the product on their platforms were Prime Computer and NCR (who, with Novell, codeveloped Portable NetWare for the Intel 80X86 and Motorola 680X0 architectures respectively), and a list that included such companies as Data General, Northern Telecom, Unisys, Sun Microsystems, Acer Counterpoint, MIPS, Intel, and Hewlett-Packard.

Application vendors announcing support for Portable NetWare include Oracle, Word Perfect, Relational Technologies (Ingres), Informix, Access Technology (20/20), Unify, and Uniplex. Three companies—Prime, Lachman Associates, and Mortice Kern Systems—announced that they would supply porting services to customers who do not want to do their own.

PORTABLE NETWARE FEATURES. Portable NetWare is fully compatible with NetWare V.2.15, allowing existing NetWare-compatible applications

to run unmodified. It provides the full range of NetWare services, including File, Print, and Management Services. DOS, OS/2, and Macintosh workstations are supported. (It is interesting, in view of the fact that most of the initial ports will be to Unix-based platforms, that neither Novell nor any of its partners has announced a timetable for extending this support to Unix workstations.)

Portable NetWare will be made available during the second quarter of 1989.

Standards. Novell is very sensitive to the charge that it does not adhere to prevailing standards. In addition to publicly proclaiming itself to be *the* open systems vendor and renaming its architecture NetWare Open Systems, with Portable NetWare, Novell is putting its product where its mouth is. Portable NetWare complies with Posix (Portable Operating System—Unix) and will be available in the ANSI C format recommended by the IEEE.

Portable NetWare under Unix has also been presented to X/Open as the proposed standard interface for PC LAN-to-Unix connections.

Novell has also entered into an alliance with Sun to incorporate NFS/ONC (Sun's Network File System/Open Network Computing) protocols into future NetWare releases. This will allow any workstation running NFS/ONC protocols to access a NetWare Server, as well as enabling workstations on NetWare LANs to reach NFS servers.

PLATFORMS. The initial interest in Portable NetWare has come mostly from minicomputer vendors, particularly those currently pursuing Unix as their server operating system, though some are doing native implementations (see "Prime" below). Novell has shown an interest in porting Portable NetWare to the IBM AS/400, which the company views as a strong platform for departmental solutions. In addition, the next version of NetWare VMS will be built on Portable NetWare.

BENEFITS TO NOVELL. For Novell, the Portable NetWare announcement is a necessary step both in blunting some of the initiative seized by LAN Manager and in furthering its own enterprise-wide architecture. Novell can now add scalability and connectivity, particularly to the Unix environment, to its story, presenting customers an architecture upon which a distributed network computing environment can be built.

OPPORTUNITIES FOR MINI VENDORS. Embracing Portable NetWare provides three specific opportunities for Novell's partners. First, these vendors can begin to sell powerful computers into the large installed base of NetWare users. Many of them are in a position to provide both specialized (i.e., database, image, computational, etc.) and general servers to users who have heretofore been limited to desktop-level machines. This is a potentially vast market that can only increase as distributed applications which make use of the client/server model are developed.

The second set of opportunities lies in providing integrating services between the PC LAN and the enterprise-wide network. Services such as connectivity (NFS, X.400, X.25, SNA, EDI, etc.), directory (X.500), and network management are value-adds that can differentiate Novell's partners' offerings. The only question here is how long this window will be available if and when NetWare itself evolves to embrace these services as a native part of Novell's offering.

The third opportunity is for these vendors to provide a single point of service to both NetWare users and those in the larger corporate architecture. This approach is already being taken by HP in agreements with 3Com and Novell.

Some of Novell's partners have given an indication of how they intend to exploit these opportunities. Some of the most interesting are the two development partners, Prime and NCR, as well as Northern Telecom and HP.

Prime. Prime, which codeveloped the Intel architecture port, seems to be the most advanced in its own plans. Two weeks prior to Novell's announcement, Prime revealed that it will provide a native NetWare implementation on its full 32-bit, 80386-based, EXL series superminicomputers. The server, which will be available in the second half of this year, is targeted primarily at applications requiring high-performance back ends, such as database engines and CAD/CAM.

In the first quarter of 1990, Prime plans to introduce, also on the Intel architecture, a nondedicated, Unix-based NetWare implementation. Also designed as a high-performance server, the Unix-based server will be the platform for Prime's value-added software, including X.400 electronic mail products, NFS gateways, Unix client software, remote procedure calls (RPCs), and X.25 protocols for wide area networking.

Prime will also provide porting services for other vendors wishing to implement NetWare on the Intel architecture.

NCR. NCR, Novell's partner in porting NetWare to the Motorola environment, is somewhat more circumspect about its implementation plans. The company does not plan to publicize specific implementation plans until mid-1989. However, it is certain that NCR is looking to move its Tower Family into the PC LAN server role.

HP. HP finds itself in an interesting position. A prime mover in the LAN Manager camp, the company is also aware that there is a large installed base of NetWare users that it would like to call (or continue to call) HP customers. Accordingly, HP has stated that it will investigate putting NetWare on the HP3000 in order to satisfy its installed base requirements. However, the company is stressing that LAN Manager Unix (LM/X) is still its strategic focus for developing the distributed network computing environment.

RELIGION AND OPPORTUNITIES. The battle of the network operating systems has now moved from the world of MCA and EISA to the great religious wars of OSF and Unix International (interstingly, much of the alignment is the same, with the big exception of AT&T). The LAN Manager/NetWare wars may prove to be less bloody, as many of the vendors announcing Portable NetWare support gave rather strong indications that they would likely implement LAN Manager in the future.

Both Portable NetWare and the Unix LAN Manager implementations will lead to Unix-based servers playing key roles in what are now PC LANs. They will appear both as powerful specialized servers and as scalable general servers where applications can be moved as they grow. The opportunities for Unix vendors are great, and the potential benefits to users are virtually unlimited. —*D. Marshak*

Patricia Seybold's Computer Industry Reports

Topics covered in Patricia Seybold's Computer Industry Reports in 1988/1989:

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

Office Computing Report

1988—Volume 11

#	Date	Title
6	June	Service And Support *Second Feature:* LAN Support
7	July	AI Update—Expert Systems Flourish While the AI Industry Disappears
8	Aug.	Metaphor Computer Systems—A Quiet Revolution
9	Sep.	Technology That Supports Meetings
10	Oct.	Document Processing
11	Nov.	Upper CASE Tools—Getting Users to Help Design Systems
12	Dec.	Executive Information Systems

1989—Volume 12

#	Date	Title
1	Jan.	Digital's Office Strategy—Going After the Desktop
2	Feb	Compound Documents Take the Stage
3	Mar	Wang's Freestyle—Not the Paperless Office, But an Office with Less Paper

UNIX in the Office

1988—Volume 3

#	Date	Title
6	June	SmartWare from Informix Software—The Quiet Contender
7	July	Olivetti—Global Bridges on a Standard Foundation
8	Aug.	BBN's Slate System—Delivering Compound Documents
9	Sep.	Distributed Databases—Where Are We?
10	Oct.	DECwindows
11	Nov.	Data General's Unix—A New Beginnning
12	Dec.	Document Processing with Interleaf *Second Feature:* Apollo and Sun Service and Support

1989—Volume 4

#	Date	Title
1	Jan.	Unisys Ofis Ensemble—In Search of Cohesion
2	Feb.	A Tale of Two Operating Systems—System V.4 and OSF1
3	Mar.	FrameMaker 2.0—The Art and Science of Document Processing

Network Monitor

1988—Volume 3

#	Date	Title
6	June	Exploring Distributed Network Computing
7	July	Transparent File Systems—The Locus Model for DNC
8	Aug.	Novell—The Juggernaut Rolls On
9	Sep.	OSI Standards—A Promise Fullfilled?
10	Oct.	ISDN: We Still Don't Know
11	Nov.	Data General's PC Networking Strategy
12	Dec.	3+Open—3Com Makes Its Move

1989—Volume 4

#	Date	Title
1	Jan.	APPN—Peer-to-Peer Networking Arrives at IBM
2	Feb.	Bridges & Routers—Dividing to Conquer
3	Mar.	BBN Builds Big Networks—Designing Private Packet-Switched Networks