Patricia Seybold Group

Editor-in-Chief
Michael A. Goulde

OPEN INFORMATION SYSTEMS

Guide to Unix and Other Open Systems

Vol. 8, No. 2 • ISSN: 1058-4161 • February 1993

Highly Available Open Systems

Expanding Today's Definition

By Kimberly Patch

IN BRIEF: As open systems move into the commercial arena, more and more users are looking for more reliable systems and software in order to keep planned and unplanned downtime to a minimum. A steadily increasing variety of high-availability options for open systems is available for a reasonable premium, which usually ranges from about 10 percent to 50 percent of the cost of the system. The caveat, however, is that users cannot mix hardware from different vendors when using the more sophisticated high-availability options such as system fail-over. *Report begins on page 3.*

Low Tech—Hi Value

Screen Scrapers Play a Key Role in Migration

ALTHOUGH THERE ARE approximately 35 million users of desktop computers or terminals with graphical user interfaces (GUIs) worldwide (roughly 20 million Windows; 10 million Macintosh; and 5 million OS/2, Motif, OpenLook or other), there are many times that number of desktops with character-based interfaces, either DOS PCs, traditional terminals, or X terminals running Xterm.

Those of us who live our days at the leading edge of technology often take it for granted that graphical user interfaces are universally accepted and universally used. This is far from the truth. The vast majority of users have no experience with mice, dialog boxes, list boxes, buttons, and all the other interface elements that are second nature to GUI users. This discrepancy poses a real dilemma for developers, particularly in-house developers who are faced with the task of re-engineering 20-year-old mainframe Cobol applications into distributed, client/server applications. The new development methodology says that MIS should be working side-by-side with users as they develop the next generation of applications. Rapid prototyping and iterative development processes are de rigeur. The vast majority of the next generation of applications being developed are graphical. But the people who will be users of those applications are unfamiliar with graphical interfaces. What kind of input can they give to the design of the "screens" if they don't know the difference between a radio button and push button? Even the concept of a screen changes in a graphical environment, with its overlapping windows and pop-up dialog boxes.

What can be done? The set of products that have been categorized by the unglamorous term *screen-scrapers* can play a valuable role. Without changing the back-end application that runs on the host, an existing character-cell application can be given a new, graphical front end. The screen coordinates of fields and labels are determined, usually automatically, and graphical elements are placed on the screen. Some of these tools are clever enough to introduce GUI-type widgets without modifying the host application by having the desktop machine manage the interface instead of the host. As a result, a legacy application can have an interface that is similar to what users are comfortable with but that contains some of the new GUI capabilities that the next generation will use.

It is possible to make the roll-out of these first stage distributed, client/server applications more gradual by first replacing the terminal with PCs doing terminal emulation. The next step can be a windows-based terminal emulator, followed by the new screen-scraped version of the application. This can be modified to include more GUI features as users become comfortable with it.

With this approach, users can move at their own speed. Then they will be able to give feedback on issues such as when keystroke navigation is preferable to mouse-clicks, and when list boxes are preferable to pop-ups. With this methodology, the design of the next generation of graphical, distributed, client/server applications can benefit from informed, experienced input from the users. After all, in this age of empowerment, empowered users should have a say about the way they will interact with an application. ◐

OPEN
INFORMATION
SYSTEMS

Editor-in-Chief
Michael A. Goulde

MCI:
MGoulde
Internet:
mgoulde@mcimail.com

Publisher
PATRICIA B. SEYBOLD

Analysts and Editors
JUDITH R. DAVIS
STANLEY H. DOLBERG
DAVID S. MARSHAK
RONNI T. MARSHAK
JOHN R. RYMER
ANDREW D. WOLFE, JR.

News Editor
DAVID S. MARSHAK

Art Director
LAURINDA P. O'CONNOR

Sales Director
PHYLLIS GUILIANO

Circulation Manager
DEBORAH A. HAY

Customer Service Manager
DONALD K. BAILLARGEON

Patricia Seybold Group
148 State Street, 7th Floor,
Boston, Massachusetts 02109

Telephone: (617) 742-5200 or
(800) 826-2424
Fax: (617) 742-1028
MCI: PSOCG
Internet: psocg@mcimail.com
TELEX: 6503122583

Open Information Systems (ISSN 0890-4685) is published monthly for $495 (US), $507 (Canada), and $519 (Foreign) per year by Patricia Seybold Group, 148 State Street, 7th Floor, Boston, MA 02109. Second-class postage permit at Boston, MA and additional mailing offices.

POSTMASTER: Send address changes to *Open Information Systems*, 148 State Street, 7th Floor, Boston, MA 02109.

Highly Available Open Systems

Expanding Today's Definition

Directions in High Availability for Open Systems

As companies re-engineer their business information systems to take advantage of the high price/performance of RISC computing and the scalability of client/server computing, users accustomed to the high availability of proprietary systems are looking for the same on their RISC machines.

Today's high-availability offerings for open systems include several different schemes for protecting data storage, along with a small but quickly growing slate of rudimentary and more sophisticated clustering options. Clustering two or more computers to back up critical applications is a cost-effective high-availability option; under the more sophisticated clustering schemes, users can use most of their compute power while ensuring that their most critical applications remain running after a short interruption in the event of a hardware or software failure. And, like client/server computing, clustering is highly scalable, allowing companies to pay for compute power more evenly over time by simply adding computers to existing clusters.

High Availability Opens Markets

High-availability options give open systems a better entree into commercial markets, where downtime means dollars lost. Companies tend to use high-availability systems for applications that would hurt the business if they went down for long, but that can withstand a short interruption in the event of a failure. Applications like debit/credit transactions in banking, and routing in transportation tend to be so time-dependent they cannot withstand even a few seconds of downtime. But many other industries, such as retail stores that depend on databases to take phone orders, can withstand small periods of downtime. High-availability systems ensure that a company can keep its most critical applications—or even all its applications—running most of the time.

Clustering Lagging in Interoperability

The major drawback of clusters of open systems that are either shipping or announced is that they only allow users to cluster computers made by the same company. As the high-availability market for open systems matures and high-availability standards develop, however, vendors should start offering solutions that can be used across heterogeneous systems. This would allow users more flexibility to mix and match open systems in heterogeneous networks.

Goals for Open High Availability

The eventual goal of open systems high availability, which will come only after clustering develops further to include heterogeneous groups of computers, is true open, distributed high availability so that any system in a network can take over for any other system that goes out of service.

Today's High-Availability Trends

There are two major trends in high availability today. One raises the floor on availability, and the other raises the ceiling. Work is being done to make off-the-shelf systems more reliable, increasing mean time between failure (MTBF) ratings for systems, disks, operating systems and even applications. Reduced parts counts and higher levels of integration are enabling vendors to deliver more reliable systems. Even smaller disk drive form factors make for more reliable systems. This means that all systems will increase in their overall availability.

At the other end of the spectrum, the capability of delivering very high availability is being implemented with features like clustering, sophisticated disk arrays that include dual-ported disks, mirrored and parity disks, and system and application software architectures designed to provide better data access and protection in the event of system failure.

Proprietary Systems Have Had the Edge

Features for delivering very high availability have historically been more obtainable on proprietary systems. And even now that high-availability features have increasingly become a part of open systems, those features generally work only within one vendor's offerings. For example, a system usually must fail-over to either an identical system or at least one from the same product line. One area where heterogeneous high availability is somewhat obtainable is database software, where some software products will write to two different brands of disk drives to provide mirrored databases.

Unix Vendors Enter the Fray

As the market for highly available open systems heats up, most major Unix vendors are mapping out high-availability strategies. The major goals in high availability are flexibilty in the configuration of storage arrays, true system clusters, and versions of Unix that do a better job directing hardware fail-overs and protecting data. Working in conjunction with the system vendors, database vendors are providing better and faster data recovery in the case of system failure as well.

These efforts promise to provide the user with an increasing amount of computer uptime by limiting unplanned downtime and—with online backups, online application software upgrades, and eventually online system software upgrades—by decreasing planned downtime as well.

Terminology Leads to Confusion

Like many computer terms, the term *high availability* tends to mean slightly different things to different vendors. By the same token, words associated with high availability, like *clustering,* also sport different definitions depending on the vendor. It is important to have a standard set of definitions so the offerings of different vendors can be compared on equal terms.

Definitions Based on Availability

Following are standard definitions we shall use for different types of systems that feature some form of protection against planned and unplanned downtime:

- **High Availability.** True high-availability designs use conventional computer technology, often in crash recovery configurations, to minimize unplanned and planned downtime. The length of delays when a program, component, or system goes down ranges from a few seconds to a few hours, but most often is between 2 and 20 minutes. High-availability systems tend to be very scalable, offering users more flexibility than other types of redundancy.

- **Fault Resiliency**. The vendors that claim a difference between high availability and fault resiliency are cutting the high-availability category in half and defining the high end as fault resilient. The key to fault resiliency is a shorter fail-over time that allows a system to bounce back quickly.

Terminology Leads to Confusion

- **Fault Tolerance.** Fault-tolerant systems run redundant hardware for all parts—including processors, power supplies, I/O subsytems, and storage subsystems—so, if a part malfunctions, there is always a hot backup. Today's more sophisticated systems allow users to use the extra hardware when it's not needed for backup in order to gain I/O or larger caches. Fail-over times for redundant components to take over for failed components are usually less than one second.

- **Continuous Availability.** The top-of-the-line in fault-tolerant products are those that offer continuous availability. A continuously available product, if it works correctly, eliminates all downtime, both planned and unplanned. The design covers hardware and software and allows for online upgrades and service. In addition, there should be no performance degradation in case of failure. Fail-over times are less than one second.

- **Disaster Tolerance.** A broad range of products and services relate to disaster tolerance, and, in some contexts, disaster tolerance is equated with high availability. All the term really means is the ability to restart or continue operations off site if the site where the computer is running is rendered inoperable due to a calamity like flood, fire, or earthquake. At the low end, disaster-tolerant products are simply off-site computers configured to a user's specifications which are available in case of disaster at the main site. At the high end, "disaster tolerant" can mean a mirrored setup off-site that can take over immediately if computers at the main site fail.

Planned vs. Unplanned Downtime

The various types of highly available systems all have a common objective—to minimize downtime. There are two types of computer downtime: planned and unplanned. Minimizing each requires a different strategy and technology.

Planned downtime is that which an MIS manager sets aside for upgrades to the system and for system maintenance. Unplanned downtime is the result of a system or component failure. Although high-availability systems are perhaps associated more with minimizing unplanned downtime, they are increasingly useful in decreasing planned downtime as well.

SOURCES OF PLANNED DOWNTIME. Probably the largest planned downtime culprit is data backup. Some high-availability disk configurations will allow for online data backup, especially with mirrored disks. The next place to look for reducing planned downtime is software upgrades. Today, some fault-tolerant systems and all continuous-availability systems allow software to be upgraded on line in order to reduce planned downtime. Some of the high-availability vendors promise that same ability within the next couple of years as well.

ATTACKING UNPLANNED DOWNTIME. In general, unplanned downtime is first reduced by using reliable parts, using parts with a fail-over path, or using parts in redundant configurations. Even with these strategies, however, a system may require a lot of planned downtime.

Software Makes Key Contribution

High-availability software acts like a combination traffic cop/maintenance person, directing hardware and software switchover and/or startup procedures when a component fails, and rebuilding data structures where needed.

High Availability Isn't Free

High availability comes with a price. Total costs can be broken down into three categories:

- Initial purchase premium
- Planning and implementation costs
- System overhead

High Availability Isn't Free

Initially, users must purchase extra software, disk space, and, depending on the setup, CPU power in order to implement a high-availablity system.

A good deal of planning is required to implement a high-availability configuration in the most effective way, especially when using the emerging clustering schemes. In order to have critical applications fail-over to another CPU, for instance, users must identify which applications are most critical and plot fail-over plans based on possible failures.

High-availability system overhead includes having to support fairly complex high-availability software products, and, for disk drives without dedicated system processors, the extra storage support required to provide double-writes for mirrored disks.

Dimensions of Cost Differences

The costs of high-availability solutions vary considerably, depending on configuration and features. Following are some very broad ranges for different types of redundancy.

HIGH AVAILABILITY. The cost premium for high-availability systems tends to range from 10 percent to 100 percent, usually falling toward the middle of that spectrum. A great deal of the extra cost of high availability depends on the extent to which the user is able to make use of the backup system for application processing. The cost of a highly available system can be as high as just over 100 percent for software and setup in the case of a 1:1 backup system that is not being used for anything else. Usually, however, the backup system can be used for noncritical tasks, cutting costs considerably.

HIGH RESILIENCY. Highly resilient systems at the high end of highly available systems pack a cost premium of 20 to 100 percent, again usually falling near the middle of that range. High-resiliency schemes are more complicated and tend to incur higher costs for planning and more overhead costs than lower-end high-availability systems. In some cases, however, the user is able to make more use of total CPU resources, reducing the overall cost.

CONTINUOUS AVAILABILITY. The cost premium for continuous availability systems tends to run from 20 to 100 percent or more and is usually nearer the high end of that range. Software for continuous availability tends to be more complicated than for high resiliency. More system components, such as CPUs, power supplies, controllers, and cables, tend to be duplicated and sometimes triplicated. Like users of the high-resiliency half of high-availability systems, continuous-availability users are able to make use of full resources more of the time than those with low-end high-availability systems.

DISASTER TOLERANCE. The premium for disaster-tolerant systems can range from a small percentage of system cost to reserve emergency computer time at an off-campus site to several times the system cost for a system that actually fails over to a second site via high-speed networking. Most disaster-tolerance offerings inherently require a substantial amount of planning.

In order to cut costs, users can provide varying measures of high availability, fault tolerance, and/or disaster tolerance for their systems, paying the highest premiums for only the company's most mission-critical applications and data.

The High-Availability Challenge

Computer industry vendors and users agree that high-availability options are one of the top requests made by Unix customers today. The chief characteristic of a highly available system is that it is not down as often as a standard system, and, when it does go down, it comes back up faster than a standard system, quickly restoring applications and sessions to the same state they were in when the system went down. Many user sites can afford to absorb a little downtime in exchange for a lower premium for high availability versus the much higher cost of providing continuous availability.

High-Availability Storage Protects Data

The first step in providing high availability is protecting the most important part of a system: the data. Several different types of redundant storage configurations provide varying types of data protection and carry different price premiums.

RAID Provides Data Protection

There are three main types of highly available storage subsystems, all of which use the Redundant Arrays of Inexpensive Disks (RAID) approach. These are: RAID level 1, or mirrored disks; RAID level 3, with parity; and RAID level 5, with distributed parity. All three types of storage generally have a seamless and nearly instantaneous fail-over time. In addition, some vendors allow users to mix and match types of RAID within a disk array. Disk arrays generally appear to applications as a single disk.

RAID LEVEL 1. RAID level 1, or disk-mirroring, uses two disks instead of one so that, if one goes down, the extra copy can take over. Disk-mirroring software transmits all data to two standard disks instead of one. The drawbacks of mirroring are that the hardware costs are doubled and write times increase slightly since data is being written to two drives. The advantages include the ability to provide online backup as well as online disk replacement, both of which cut down on planned downtime. Mirroring also eliminates several single points of failure because, with two disks, there are two cables and two I/O cards.

RAID LEVEL 3. A RAID level 3 array provides an extra disk drive that acts as a parity drive for every two or four disks. Parity information, consisting of a check sum calculated from the contents of the other drives, is stored on the parity drive. If one disk goes down, the parity disk steps in to replace it, using the parity information in conjunction with the data on the other remaining disks to reconstruct the data on the inoperable drive.

RAID LEVEL 5. A RAID level 5 array is a combination of RAID level 0, where data is striped across several drives, and RAID level 3, where there is an extra parity disk. In a RAID level 5 setup, four drives' worth of data and parity information is striped across five drives, so that, if any one drive fails, any of the other four can provide all needed data. Disk-striping also increases disk I/O for large data transfers.

OTHER TYPES OF DISK ARRAYS. Disk arrays range from simple subsystems without many extra features to sophisticated subsystems that allow users to mix and match RAID levels. The more sophisticated disk subsystems may also contain processors that offload I/O processing, disk-formatting, error protection, and RAID algorithms from the system CPU. Most arrays are dual ported, allowing users to attach the arrays to two different systems. Pyramid makes a drive that has four I/O connections.

High-Availability Storage Costs

Cost premiums for disk-mirroring tend to be 100 percent, since disk needs are doubled in a 1:1 redundant configuration. Extra disk costs for RAID levels 3 and 5 are either 33 percent, when there is a parity drive for every two drives, or, more commonly, 20 percent, when there is a parity drive for every four drives. In addition, there are software and/or disk array costs. Some companies offer disk-mirroring software that can be used with off-the-shelf-disk subsystems. Others offer mirroring abilities in RAID subsystems made for that purpose. Subsystems using RAID levels 3 and 5 vary in cost depending on features.

High-availability storage may incur system overhead. For example, the system CPU must process two writes for everything saved on mirrored disks that are not part of a mirrored disk array which has its own processing power. The more sophisticated disk arrays, however, offload overhead onto I/O processors designed as a part of the drive array.

With storage in general, however, there is a continuing downward trend toward costs approaching one dollar per megabyte, a trend that makes today's redundant storage solutions less costly compared to regular disk subsystem costs of just a year ago. Therefore, RAID is rapidly becoming a basic system requirement rather than a special availability feature.

High-Availability System Configurations

Several companies offer high-availability system configurations that range from a one-on-one, hot-standby system to very loose clustering arrangements that allow one system to take over for any of several different systems in the cluster.

High Availability through Clustering

The term *clustering* has a variety of meanings in the computer industry today. The strict definition is "to make collections of machines appear as one to the operating system, system software, applications, and users." Machines clustered together in this way would be able to very quickly distribute work from a failed processor around the cluster. This is an eventual goal of many of the high-availability vendors. There are several vendors that call their highly available system setups "clusters" or "loose clusters," however there are few clusters available today that match the strict definition.

High Availability through Fail-Over

With current high-availability system designs, fail-over involves switching accounts and applications to another machine and ensuring that there was no data loss or corruption during the failure and changeover. Depending on the system, some or all of this process may be automated.

FAIL-OVER MECHANISMS. Highly available systems communicate with their backup systems via a very small daemon "heartbeat" program that allows the backup system to monitor the primary system or systems it is backing up. If a heartbeat is missed, the cluster goes into fail-over mode.

Fail-overs can be manual or automatic, and there are several degrees of automation. For instance, in some cases, users are instructed to log off and then log on again. In other cases, the fail-over is more transparent; the user does not have to do anything but wait a brief period. Sometimes the user can choose between manual or automatic fail-over. In some implementations, users will be able to resume exactly where they left off after a fail-over. In other cases, they may be prompted to reenter their last transaction.

SIZING BACKUP SYSTEMS. A backup system does not necessarily have to be the same size as the system it is backing up, and a user can save money by backing up a larger system or systems with a smaller one. This means either taking a performance hit in case of failure or having only mission-critical applications fail-over to the second system.

In addition, some users prefer not to run applications on the standby machine, although many run a light application load on the standby server in a cluster. Users also say the setup allows them more flexibility to grow more smoothly.

High Availability in Operation

System Setup Requirements

Most highly available systems today require a fairly involved setup in order to configure the cluster to properly carry out procedures required in the event of a failure. Users can program the scripts themselves or have a system integrator or the vendor write them. Depending on how complicated the system setup is and the type of systems the user is migrating from, it may take anywhere from one or two days up to weeks or several months for experienced programmers to write scripts directing highly available system actions in the event of system failure. Many vendors provide some standard scripts with the setup. In addition, several provide setup services that include programming script fail-over scenarios, or they will work with the customer to write or modify scripts. Users can customize the scripts, but it takes a programmer who is proficient in Unix and in C to do the configuration properly.

Fail-Over Time Considerations

Highly-available system fail-over downtime can range from a few seconds to 20 or 40 minutes or more. A fail-over procedure involves the following: The backup machine senses a failure, and then follows a script that will most likely include rebooting, transferring user addresses, getting the proper applications up and running, and taking steps to ensure that data has not been corrupted or lost. Fail-over time depends mostly on how quickly the second machine can get applications up and running and how quickly the operating system and applications such as databases or transaction monitors can get data straightened out.

In general, hardware fail-over takes on the order of a minute or two, and the system is rebooted in another minute or two. In most cases, it takes 5 to 20 minutes to get applications up and running with data either fully recovered or users instructed to reenter the last transaction.

System Overhead Variables

System overhead depends on the system and setup used. For straight fail-over setups, it is very low—on the order of a fraction of a percentage point to 1.5 percent. However, users must add to that overhead any unused CPU power in a standby system to get a true overhead cost. Although customers tend to use the backup system for noncritical applications, it tends to be used more lightly than a regular system. True clustering systems, such as Digital Equipment Corporation's VAXclusters and the new LifeKeeper Cluster from AT&T's NCR Division, are much more sophisticated than simple fail-over setups and use all available CPUs fully, but they also entail higher overhead, which increases with the number of nodes.

System Software Requirements

When vendors provide fail-over systems, they usually provide additional high-availability features in their versions of Unix in order to allow a backup system to boot quickly and/or recover quickly in case of fail-over.

File System Journaling

The ability to journal changes to the file system ensures that files are always in a usable state. When a system fails, a journaled file system ensures that files are saved in the last consistent state. This allows systems to fail-over without any data corruption and with either no data loss or the loss of only the last transaction. This is contrasted with systems that only journal the file systems' metadata, a procedure that helps maintain file system integrity but not data integrity.

Process Fault Isolation

Process fault-isolation technology is also sometimes employed for heavily used software components, such as the file system, to make certain that errors in one system are isolated and do not propagate throughout the system.

Transaction Processing (TP) Monitors

Transaction monitors are sometimes used to direct fail-overs and ensure that there is no data loss. Uncommitted transactions can be rolled back and databases returned to a known state. Transaction monitors for Unix include USL's Tuxedo, Transarc's Encina, IBM's CICS/6000, and NCR's Top End.

Other Software Functions

In addition, online diagnostics are playing an increasing role in allowing users to forestall problems that could lead to downtime. Currently, these are system specific. In the future, diagnostics will become part of distributed management.

Application Software Requirements for High Availability

The first open systems applications built with high availability in mind were databases and communications applications. Highly available databases ensure ready access to information—the lifeblood of many corporations and the underpinning of today's information economy. Highly available communications software strengthens fail-over schemes and lays the groundwork for distributed high-availability and disaster-tolerant systems.

DATABASE HIGH AVAILABILITY. Several database companies, like Oracle and Sybase, include software with their databases that allows for quick file reconstruction in case of system failure. This reduces downtime for mirrored server and clustering solutions.

The Oracle7 Parallel Server allows multiple systems to read and write across a single Oracle database image. Load-balancing and a distributed lock manager, which allows multiple systems to access the database at the same time, improve potential database availability.

The Oracle7 Parallel Server was developed for Digital's VAXclusters. Oracle also worked with Sequent to port a version to Sequent computers. The database has also been optimized for Pyramid clusters, but Pyramid has developed its own fault-tolerant distributed lock manager for the setup, ensuring that the system will keep running even if a fault occurs in the distributed lock manager software. NCR is also working with Oracle on applications for the parallel server.

In addition, several third-party database add-on products provide users with increased availability. Afic Computer Incorporated (New York, New York), for instance, sells a product called Multi Server Option (MSO), which allows users to concurrently write to any number of mirrored Sybase databases at distributed sites over a TCP/IP network. Information is broadcast to all databases simultaneously. MSO also allows for load-balancing over multiple servers. If one server goes down, the software will direct a user to a duplicate database within 30 seconds. MSO costs $12,000 to $26,000 per server depending on the size of the server. Afic is currently readying a version for Oracle, which will be ready by the end of the quarter. The company is also planning a version for Informix, which should be available by the end of this year.

Networking and Communications Requirements

Highly available systems require highly available communications as well. The faster the communications between machines, the quicker the fail-over. Some highly available configurations feature dual communications links in order to eliminate the link as a single point of failure. The trend toward higher bandwidth LAN and WAN networks gives both local and remote computers faster communications, which translates to faster fail-over times for distributed systems. Digital, for instance, is slated to announce a Fiber Distributed Data Interface (FDDI)-based LAN interconnect for its Alpha-AXP clusters in the second half of 1994.

Options Available for Available Networks

Where the networks themselves are concerned, users' choices are currently limited to making sure that networks are inherently reliable by reducing single points of failure and using more sophisticated networking devices from various vendors, such as routers and network hubs. Routers, which determine data paths around networks, can calculate new paths in the event of a link failure. Hubs can be configured with redundant parts and can isolate faults in a physical network to prevent the network from going down. Network analyzers are also playing an important role in keeping systems up by keeping them functioning optimally and alerting system managers of any symptoms that might eventually lead to downtime.

In addition, high availability in networking is always highly dependent on the size of the network. As a network gets larger, it tends to gain built-in redundancy: the larger the network, the more likely the use of an alternate route if a component failure blocks a portion of a network.

Vendors Address the Requirement

Vendors are starting to address high-availability networking. Data General, for example, is working on an automatic LAN fail-over network that is slated to be available in the second half of 1993. In addition, the Asynchronous Transfer Mode (ATM) network switching scheme promises to eventually bring users high-bandwidth wide area networking (WAN). The first ATM products are just coming onto the market today, however, and the technology is not likely to be widely used until the mid to late 1990s. ATM will, in theory, allow enough information to be passed quickly enough over WANs to allow for high-speed fail-over between remote CPUs.

High-Availability Networks Add Costs

Prices for highly available networks are very dependent on network configuration. Here are some ballpark figures for highly available peer-to-peer networks:

- For a small network with a single site and less than 5200 nodes, a user should expect to pay about 50 percent extra for high availability.
- A medium network with multiple sites and up to 5000 nodes commands a 30 to 40 percent premium.
- For a very large site with more than 5000 nodes, the premium drops to around 20 to 30 percent.

The higher availability of fault tolerance adds addtional costs. The premium for a basic fault-tolerant network tends to be about 80 to 100 percent or more and varies less with network size than with high-availability premiums.

The illustration on page 13 shows the relationship between increased availability and the cost of providing that availability. It demonstrates the cost premiums associated with different approaches, beginning with more reliable parts and progressing through fully duplexed systems.

Vendor Profiles: Making Availability Pay

Today, a handful of vendors offer highly available open systems. They include NCR , Data General (DG), Digital Equipment Corporation, Hewlett-Packard (HP), IBM, Pyramid Computing, Sequent Computer, Stratus, Sun Microsystems, and Tandem. (See Table, page 12.)

In general, Pyramid and Sequent offer large, sophisticated high-end systems that range in cost from six figures to more than a million dollars. Midrange and workstation-level Unix systems from DG, HP, NCR, IBM, and Sun offer varying degrees of high availability, with NCR and DG probably leading the way in current solutions that are shipping today. NCR and IBM are gearing up to ship true clustering software. Digital has announced "open clustering" abilities for its computers, including its new AXP lines, but any cluster must contain at least two OpenVMS systems. Tandem and Stratus specialize in high-end fault-tolerant and continuous-availability Unix products that range in price from a half million to several million dollars.

Company Comparison of High-Availability Features

Features	DG	Digital*	HP	IBM	NCR	Pyramid	Sequent	Stratus	Sun	Tandem
Automatic system fail-over	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Possible number of systems	Two	96 of one to six CPUs	Two to eight	Two in Modes 1, 3; four in Mode 2	Two or three	Two to four; 2-24 CPUs each	One system of up to six CPUs		Two	Two to 255; 2-16 CPU's each
RAID levels supported	0, 1, 3, 5	0, 1, 3, 5	0, 3	Mirrored disks	0, 1, 3, 5	0, 1 (5 by spring 1993)	1, 3 (5 by end of 1993)	1	0, 1	Mirrored disks
Multiple RAID level support within one storage subsystem	Yes	Yes	Yes	——	Yes	Yes	Yes	——	Yes	——
Dual-ported storage subsystem	Two	Two (can use a star coupler to gain more connections)	One	Two	Two	Four ports	Two	One	One	Two
Online disk backup and restore	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Online disk swap	Yes	Yes	Yes	Yes (for 9333)	Yes	Yes	Yes	Yes	Yes	Yes
File-journaling	Yes	Yes	No	Yes	Yes	Yes	2H 1993	No	No	Yes
Software fault isolation	Yes	Yes	No	Yes	No	Yes	No	Yes	Yes (with Fusion)	Yes
Distributed lock manager	Yes	Yes	No	Yes	3Q93	Yes	Yes	Yes	No	Yes
Load-balancing software		Yes	No	Later in 1993	3Q93	Yes	No	Yes	Yes (with Fusion)	Yes
System overhead to run high-availability software	>1%	1-2%	>1%	1.5%	5% (2-node) 7.5% (3-node) 3Q93 25% (2-node) 38% (4-node)	3-7%	>1%		>1% (with Fusion)	1-2%
Multidirectional fail-over	No	No	No	No	Yes	No	No	No	Yes (with Fusion)	No
Average system fail-over time (according to vendor)	5 min.	3-15 sec.	10-20 min.	30 sec. to 5 min.	2-10 min. >1 min. 3Q93	AMS, 7-10 min.; RM 3 min.; VCs immediate, depending on configuration	1-2 min.	Immediate	>5 min. (with Fusion) 10-15 min. without.	>1 min.

Table. Features comparison for vendors offering open systems that offer high availability.

*Digital's clusters require that two nodes be running its OpenVMS operating system.

Vendors Target Services toward High Availability

High-availability products are extremely complex, requiring extensive integration and support for proper implementation. As a result, with demand for high availability increasing, computer companies are doing more consulting and systems integration work for their customers.

In addition to computer company system integration, a lot of third-party companies offer sophisticated system integration services that include providing various levels of high availability on standard platforms. Because there are many possibilities in the way a system or application can fail-over, vendors and their third-party companies are offering more services to customers trying to script different failure scenarios to ensure the best return on their high-availability investments. The cost of these services is usually well over $1,000 a day, although it is sometimes bundled into the hardware deal. Many vendors also offer one-week user training sessions to teach users who already know Unix and C to learn to write and alter fail-over scripts.

Finally, disaster-tolerant solutions require extensive planning and are, therefore, another area where services are being offered. Disaster tolerance includes making systems, applications, and technical staff available at an outside site for users who want to make sure their applications will continue to run even if disaster strikes their place, or places, of business.

Availability Trade-offs: Reduced Downtime Costs Money

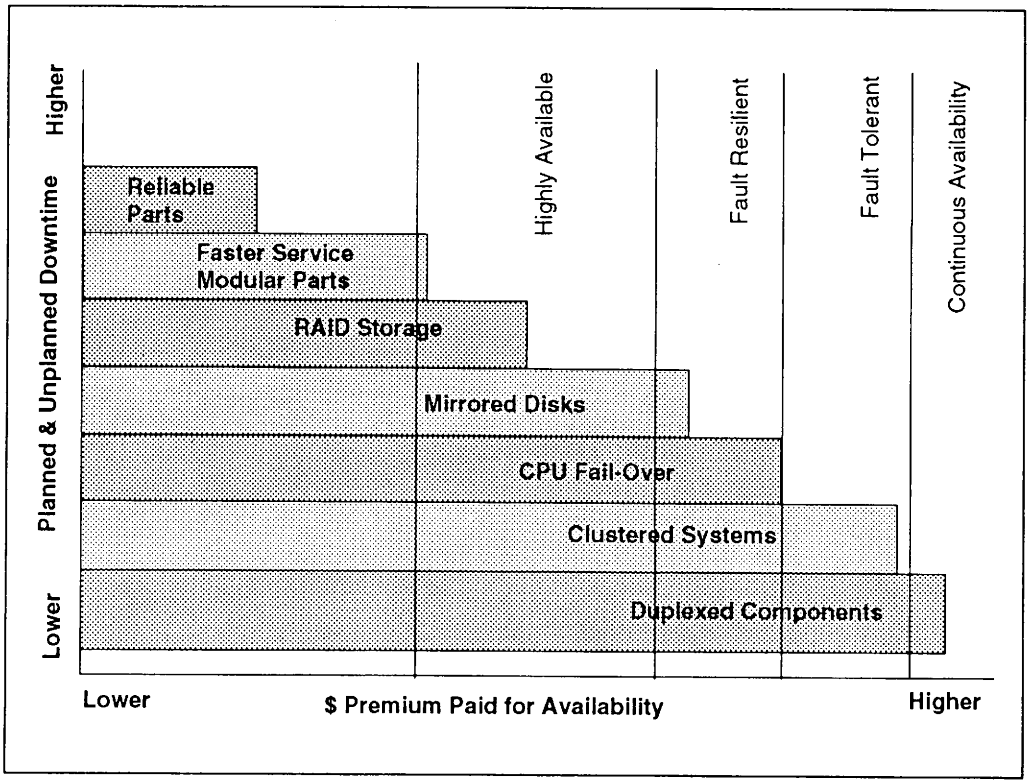

Illustration. The cost of providing availability with different design approaches.

Data General Offers Cost Effectiveness

DG's line of high-availability products includes systems that feature fully automatic recovery without loss of data, highly configurable disk arrays, and high-availability features built into system software.

STORAGE ENHANCEMENTS. DG's CLARiiON disk subsystems offer the user any combination of RAID levels 0, 1, 3, and 5. They include processors that offload from the main CPU the tasks of disk-formatting, error protection, and RAID algorithm processing. The subsystems allow for online backup and restore. In addition, subsystem cooling modules, power supplies, and disk drives can be swapped in and out without taking the subsystem down.

According to DG, system overhead is negligible for both system fail-over setups and for disk arrays.

SYSTEM SUPPORT FOR AVAILABILITY. DG's AViiON line of systems includes redundant controllers, power supplies, and fans in order to eliminate some common points of failure. Users can configure one system to fail-over to one other system. The systems fail-over automatically without any data loss; users do not have to log on or off and do not have to reenter any transactions after a fail-over.

SYSTEM SOFTWARE SUPPORT FOR AVAILABILITY. DG's DG-UX operating system includes file-journaling and fault isolation. The operating system also includes AV/Alert, which monitors and reports on system status and error conditions.

DG systems use the Tuxedo transaction processing monitor to reroute transactions during a fail-over. Tuxedo also handles the roll-back of transactions in process when the system has failed.

CUSTOMER SERVICES. DG provides system fail-over scripts that users can modify. DG's Implementation Services group will custom-write fail-over scripts for the customer. Implementation Services' costs are negotiated in conjunction with the original purchase contract; the cost of these services for a typical full installation, including software, disk array, and system, is about $10,000. About half DG's customers opt for services from the Implementation Services Group.

FUTURE DIRECTIONS. DG is expected to announce 2GB disks for its RAID arrays in 1993, which will allow the company to provide CLARiiON arrays with up to 40GB of disk capacity. In the second half of 1993, Data General is expected to announce hardware isolation features and LAN fail-over. Hardware isolation will cut downtime by allowing AViiON multiprocessing systems to isolate and reconfigure around CPU and memory failures by going down to the boot shell, then coming back up. LAN fail-over will allow a user to fail-over to a standby LAN.

Digital Leverages Cluster Experience

Digital offers a wide range of high-availability products for its proprietary systems, which include the most sophisticated clustering technology on the market.

CLUSTERING CAPABILITIES. Digital invented clustering a decade ago using its proprietary VMS running on VAXs. Last February, Digital announced OpenVMS, which supports POSIX and is XPG3 compliant, making it, by most definitions, an open system. Digital's clusters, which use sophisticated load-balancing to distribute tasks throughout the cluster, are totally transparent to the user. If a system fails, the user's session is back up within 3 to 15 seconds.

Digital has announced that limited clustering abilities for its Alpha AXP-based RISC systems will be available near the end of the second quarter of 1993. Full clustering abilities for the AXP-based machines, including an FDDI-based interconnect, will be available late in the second half of 1994.

Digital's future clusters will be highly available open systems, with one caveat: At least two of the systems in a cluster must be running OpenVMS. The clusters will serve OpenVMS, OS/2, or, when it becomes available, the Windows NT operating systems.

STORAGE SUPPORT. Digital offers RAID 0, 1, 3, and 5 subsystems, and all Digital disks have dual porting capability, allowing two systems to access the disk at any given time. A modular storage system, code-named STORM, is being readied for shipment. It allows different levels of RAID disks and tape drives to be supported in one subsystem.

COST PREMIUM. According to Digital, the additional cost of hardware and software using high-availability clusters ranges from 2 to 25 percent, depending on the size of the cluster and the applications. The overhead associated with clustering is about 1 to 2 percent.

HP: Getting up the Learning Curve

HP's offerings for higher availability include disk arrays, mirrored server software called Switchover/UX, and custom disaster-tolerance services. HP's "loose cluster" scheme involves one standby system supporting from one to seven primary systems, all of which must be HP 9000 Series 800 computers. The primary systems send out periodic "heartbeats" to tell the standby they are working fine. When the standby senses a missed heartbeat, it shuts its processes down, takes control of the faulty processor's disks, reboots, assumes the faulty processor's network address, then restarts applications. The entire fail-over process usually takes between 10 and 20 minutes but could take longer, depending on the application.

SWITCHOVER REQUIRES MANUAL INTERVENTON. The SwitchOver/UX software that directs HP's loose clustering setup requires that users manually log off, then log on again and restart their applications. Users usually have to reenter the transactions they were working on when the system went down. HP obviously has some catching up to do in automatic fail-over.

RELIABILITY AND STORAGE AN HP STRENGTH. HP's systems and disk drives are renowned for their reliability and high MTBF. Its HP-FL disk arrays use RAID levels 0 and 3, and feature online disk replacement. HP's mirrored disk software is called MirrorDisk/UX.

FUTURE DIRECTIONS. HP is expected to announce RAID level 5 in mid-1993. Version 10.0 of HP's HP-UX operating system, which will be released in late 1993, will support file journaling and distributed lock management. At that point, HP should be able to compete more effectively in high-availability applications.

IBM Offers Loose Clustering for Starters

IBM offers three modes of two- or three-system loose clustering using its AIX High-Availability Clustered Multiprocessor/6000 (HACMP/6000) software. The most sophisticated clustering configuration that IBM has announced will not be available until March 1993, but it approaches real clustering and offeres a single database image. The three modes of automatic system fail-over supported are:

- Mode 1, which allows a fail-over from a primary system to an idle backup system.
- Mode 2, which allows the backup processor to run noncritical applications that can be bumped off or run in a degraded mode if the primary processor fails.
- Mode 3, which will be available in March 1993, can be called a real cluster because the systems run concurrently and share logical and physical resources, taking advantage of the lock manager portion of HACMP/6000. Mode 3 also requires IBM's 9333 High-Performance DISC Drive Subsystem.

STORAGE ENHANCEMENTS. IBM's 9333 High-Performance DISC Drive Subsystem uses a serial disk interface that can support up to 17.6GB. The 9334 uses a SCSI-1 interface that supports up to 8.2GB. They are both dual ported and can be mirrored via an operating system software utility.

SYSTEM SUPPORT FOR HIGH AVAILABILITY. IBM offers high-availability configurations with the destkop, deskside, and rack-mounted models of its uniprocessor RISC System/6000 series of systems. The systems are essentially unmodified, which means that the HACMP option can be added later.

FLEXIBLE FAIL-OVER CONFIGURATION. For IBM's current products, the system, system software, and application fail-over process averages from 30 seconds to five minutes. More time may be needed for applications to recover their data, however. The systems can be configured so that users log onto the backup machine or are logged on automatically. The user may have to reenter the last transaction, however, depending on the database and application.

IBM supplies some scripts and provides hooks in the HACMP software so that users can customize scripts to direct what will happen during a fail-over.

SUPPORT SERVICES. IBM business partners ClaM Associates, Incorporated (Cambridge, Massachusetts) and Cluster Computing, Incorporated (Sunnyvale, California) provide installation and configuration services for the three modes of high-availability systems. A typical installation takes three to three-and-a-half days; more complex installations may take two to three weeks, including planning and system testing.

COST PREMIUMS. According to IBM, the average user spends an additional 30 percent in system and disk costs for a Mode 2 high-availability configuration. Users must purchase an extra system for Mode 1, but no extra disk, making total extra cost about 66 percent, depending on the complete configuration.

The system overhead needed to run the AIX HACMP/6000 software in any of IBM's three modes is about 1.5 percent.

FUTURE DIRECTIONS FOR HIGH AVAILABILITY. IBM is expected to release FDDI LAN support for its HACMP/6000 software in the first half of 1993. The company is also expected to include load-balancing software in the next version of its operating system. It is also readying an SCSI-2 version of its 9334 disk subsystem. Multiprocessor versions of the RS/6000 are being developed in conjunction with Groupe Bull and are expected to be announced later in 1993. These are likely to include high-availability features in their design.

NCR (AT&T)

NCR recently announced the availability of sophisticated LifeKeeper high-availability software which, along with the company's robust disk arrays, brings its systems to the edge of true clustering abilities. More advanced capabilities included in LifeKeeper FRS (Fault Resilient Systems) Clustering software, slated for release in July 1993, will give NCR systems true clustering abilities. NCR clusters can comprise any members of NCR's 3400 and 3500 lines of systems, each of which may contain from one to eight processors.

STORAGE SUPPORT. NCR's 6298 Disk Array Subsystem supports RAID levels 0, 1, 3, and 5 in any combination. Users can swap disk drives, fans, and power supplies out of the subsystem while it is powered up, running, and even while it is transferring information. It has three ports and is also available with redundant controllers. It holds up to 20 1GB drives.

SYSTEM SOFTWARE. NCR's LifeKeeper software allows for a two- or three-node high-availability system with multidirectional switchover, so that two or three machines can back up each other. This offers more protection because applications on either machine can be failed over in the event of a failure. Planned downtime for software installation can be reduced because a fail-over can be manually triggered, the software upgraded, and the fail-over reversed. LifeKeeper also permits recovery from system, application, and peripheral errors. It also provides for automatic switchover in case of failure and operator-initiated switch-back. LifeKeeper began shipping in January and will be generally available in the second quarter of this year.

LifeKeeper uses a fine-grained approach to software failures, failing over only those users associated with the application that has failed rather than all the users on a system. This ability, together with NCR's multidirectional switchover abilities, allows for software upgrades with two 2-to-10-minute interruptions to switch over and back. NCR provides standard recovery scripts that users can customize for their configurations within a few days. Recovery time for LifeKeeper software ranges from 2 to 10 minutes, depending on the application.

When using the Oracle7 Parallel Server database, LifeKeeper's distributed lock manager allows NCR's high-availability systems to address a single database image.

COST PREMIUM. LifeKeeper system overhead is about 5 percent for a two-node setup and 7.5 percent for a three-node setup.

SERVICES. NCR's Professional Services provides user training, support, and installation and configuration systems.

FUTURES. NCR will provide true clustering when it announces its LifeKeeper FRS Clustering software, which includes a distributed lock manager and load-balancing software, in July 1993. Clusters configured with this software will be able to contain up to four nodes and will have recovery in seconds rather than minutes. This setup will exact a higher system overhead similar to Digital's proprietary clusters. A two-node cluster will have an overhead of about 25 percent; a four-node cluster will have an overhead of about 38 percent. LifeKeeper FRS Clusters will also allow single-database images and Unix file systems to be addressed by multiple computers. NCR's grand plan for clustering is to provide campus-wide clusters within two years and, eventually, global clusters, which will allow for quick disaster recovery.

Pyramid Computer Offers Range of High-Availability Products

Pyramid offers midrange to high-end high-availability solutions with three levels of high availability.

HIGHLY AVAILABLE SYSTEMS. Pyramid servers contain between 2 and 24 CPUs each and include a built-in high-availability diagnostic subsystem called the Availability Management System (AMS). If a CPU in a Pyramid server fails, it sends a message to its AMS and shuts itself down. The AMS determines which CPU is responsible for the fault, takes it offline, and brings the rest of the system back up.

Pyramid offers two additional levels of high availability using its Reliant Monitor and Valence Clusters software. Reliant Monitor allows a primary system to fail-over to a second system. In the event of a fail-over, Reliant Monitor can log the users off the second system, or the system can keep its original users and take on the users from the second system as well. In the latter instance, however, system performance may be adversely affected.

Pyramid's Valence Cluster software, when used together with the Oracle7 Parallel Server, allows up to four systems to share a single Oracle database image. Pyramid has replaced the Oracle distributed lock manager with a fault-tolerant lock manager which can recover transparently from distributed lock manager errors. In addition, the Tuxedo network monitor can be used to direct a transparent fail-over.

It takes 7 to 10 minutes for the AMS system to get a server up and running after a failure. It takes about three minutes for a system using the Reliant File System (RxFS) software to fail-over. Valence Clusters that use front-end servers fail-over instantaneously. If no front-end server is used, users manually log onto another server if a failure occurs.

STORAGE. Pyramid's High Availability Array System (HAAS) disk subsystem supports any combination of RAID 0 and 1 and also includes a redundant power supply and four interchangeable controllers. It supports up to 37.2GB of storage.

Pyramid implements RAID in software that is independent of underlying hardware. This allows the company to offer a tape implementation of RAID as well. Its FastTRAC redundant tape arrays allow users to spread data transfer across several channels and configure large virtual tapes. A parity tape improves data integrity.

AVAILABILITY FOR USERS. Even when systems are functioning properly, the availability of resources may be skewed toward a particular process, job, or user, negatively affecting the availability of resources for other users. Pyramid Fair Share configuration software addresses this problem by allowing a system manager to assign certain amounts of system resources to different users or groups of users.

COST PREMIUMS. In general, users pay about a 25 percent premium for Reliant Monitor and Valence Cluster software and about a 5 percent premium for Fair Share software. System overhead ranges from 3 to 7 percent, depending on configuration and applications run.

FUTURES. Pyramid is expected to ship a RAID 5 subsystem this spring. The company is also expected to add to its AMS systems the ability to isolate and shut down bad memory boards.

Sequent Clusters for Availabiity

Sequent offers midrange to high-end high-availability solutions. Sequent's Hi-Av System provides duplexed systems that can share disks. Users can choose manual or automatic system fail-over. Sequent's ptx/CLUSTERS software includes a lock manager that allows for data-sharing among applications. ptx/NQS balances batch-job loads within a cluster, and ptx/LAT connects online users to the node best able to handle their applications. ptx/ARGUS provides centralized management; ptx/SVM is a mass storage management tool. Fail-over times for Sequent's clusters average one or two minutes.

Sequent systems running the Oracle7 Parallel Server give the two systems in a Hi-Av setup access to a single Oracle7 database image running on commonly shared disks.

STORAGE SUPPORT. Sequent's SCSI-2 disk subsystems support any combination of RAID levels 1 and 3.

SERVICES. Sequent provides extensive technical consulting and week-long training sessions on clustering.

FUTURE DIRECTIONS. Sequent is expected to announce a four-system cluster in fourth quarter 1993. The company is also expected to support RAID 5 by the end of 1993.

Sequent also intends to provide services similar to its Oracle7 Parallel Server for more databases. The company is also working on software that will provide faster fail-over and speed file-system recovery.

Stratus Leverages Continuous Availability

Stratus has carved out a significant niche for itself with its continuously available systems. Its systems, resold by IBM as well as by Stratus, are used for applications in which absolutely no unplanned downtime is acceptable. Stratus sells continuous-availability systems that use full hardware duplication and tightly coupled software. The Stratus XA/RISC (XA/R) Series are redundant machines with all components duplicated. Duplicate pairs of components are in constant cross-checking contact. If an error is detected, the pair shuts down and work is spread over the remaining components in the system. The systems take advantage of some of the duplication by using the double caches and faster disk I/O to provide better performance than a single, non-duplicated system.

Fail-over times are instantaneous. That means that users or processes experience no interruptions as the result of the failure of any component.

STORAGE. Stratus provides mirrored disks as its approach to assuring availability for its systems.

COST PREMIUM. The cost premium for Stratus's continuous availability runs from 20 to 80 percent, depending on the system.

Sun Microsystems: Newcomer to High-End Commercial Requirements

Sun's high-availability offerings are sketchy, perhaps because the company is still relatively new to the commercial market. Only with the SPARCcenter 2000 system announcements did Sun really have a system it could aim at the high end of the marketplace. Sun's most powerful server, the SPARCcenter 2000, is scheduled to ship in volume in April. It can be configured with two to eight CPUs, will have 1 Terabyte of disk storage, will support up to 3,000 users, and will contain two S-bus slots per CPU.

SYSTEM SOFTWARE. Sun's line of SPARCservers can be configured using software from third-party vendor Fusion Systems Group (New York, New York) to fail-over to a second system without rebooting the second system. This allows users who were logged on to the second system to stay logged on to the system. The Fusion product allows for system fail-over time of under five minutes. Fusion also provides multidirectional fail-over for Sun servers.

STORAGE SUPPORT. Sun offers mirrored-disk software. An online disk suite provides a pool of hot spares that can replace any mirrored disk that goes down.

FUTURE DIRECTIONS. By the end of 1993, 16-CPU SPARCcenter 2000 will be available. A 20-CPU version will be available in the first quarter of 1994. These systems will extend the reach of SPARC-based systems for Sun.

Tandem Pushes Fault Tolerance to the Limit

Tandem's systems are in a notch below Stratus's systems in terms of availability. Like Stratus, Tandem computers are high-end machines with very short fail-over times. Instead of paired, duplicate hardware, Tandem uses multiple loosely clustered CPUs and more complicated operating system software that does extensive software and hardware error-checking. If a CPU develops a problem, it shuts down, and the remaining CPUs in the cluster take over the work. Tandem's loose clustering scheme can support up to 255 nodes of up to 16 CPUs each. Fail-overs occur in less than one minute.

STORAGE. Tandem provides mirrored disks.

SYSTEM SOFTWARE. Tandem's Pathway Transaction Monitor provides load-balancing for the cluster. Tandem's Remote Database Facility (RDF) software allows a user to keep a second database image off-site that is up-to-date within one minute. This provides for very quick disaster recovery.

Future Directions In Open High Availability

In the next year or two, high-availability options for open systems are likely to improve on several fronts:

- The reliability of parts will continue to improve, making MTBF ratings rise and giving users more standard uptime.

Future Directions In Open High Availability

- Systems and storage subsystems will become more modular, allowing users to more easily replace parts and disk controller and data communication I/O cards even under power, improving uptime.

- A trend toward more graphical representation of error reports will allow users to better interpret them; more automatic reporting of faults will also cut failures resulting from operator error.

- More vendors will follow NCR's lead in allowing bidirectional switchover. This allows users to protect more processes and eliminates any switch-back downtime after a fail-over has occurred. This will reduce planned downtime.

- As the high-availability trend meets the trend toward distributed computing, an increasing number of high-availability features will be engineered especially for distributed open systems, including sophisticated fail-over software for LANs and high-speed optical links, which will also speed fail-over times. Distributed clusters will improve disaster tolerance.

- More sophisticated system software will allow for widespread true clustering (the ability to support large numbers of systems on a single cluster) and faster fail-over times.

- Database software will likely continue to lead the way in producing faster and more seamless application fail-overs.

- Support for larger amounts of storage per system and more ports per storage subsystems will increase the amount of highly available storage a system can support and increase the number of different I/O channels a user can go through in order to access that information. This will also help pave the way for larger clusters.

- Vendors will offer more sophisticated installation and support services, including turnkey loose clustering and turnkey true clustering solutions.

- Vendors will increase and fine-tune online hardware and software maintenance and upgrades to significantly reduce planned downtime.

- In addition, several years down the road, as high-availability features in Unix systems mature, truly open high-availability clusters that support systems from different vendors will most likely emerge.

Summary and Conclusions

High-availability solutions for open systems are improving significantly because several major players have recognized the importance of the market and are actively working to better each other's storage and system high-availability offerings by decreasing the potential for both planned and unplanned downtime.

Commercial users who want to take advantage of the price/performance of RISC processing and the flexibility of Unix systems will benefit from this substantial industry push to bring high availability to open systems at a fairly reasonable cost.

However, for today's offerings, the more sophisticated the high-availability solution, the longer the user should be willing to work to plan fail-overs and write scripts to direct fail-overs. Today's most sophisticated high-availability solutions still require that fail-over scenarios be planned out and scripts be written by a technician familiar with Unix and C.

Summary and Conclusions

In order to effectively compute what type of high-availability products are worth buying for any given situation, a user should estimate the costs of both planned and unplanned downtime that would potentially be avoided by using the products, and compare the potential savings to the premium that must be paid for the various types of high-availability systems.

The costs of high availability include the premium paid when the user initially buys a system, system overhead, and, especially with clustering solutions, planning and installation costs.

In deciding what highly available systems to buy, it is also important to take into account the vendor's credibility and ability to support these systems. ©

Afic Computer Incorporated 160 Broadway Suite 600 New York, NY 10038 212-406-2503	ClaM Associates Incorporated 25 First Street Cambridge, MA 02141 617-621-2542
Fusion Systems Group 225 Broadway New York, NY 10007 212-285-8001	Cluster Computing Corporation Sunnyvale, CA 408-370-2600

Next month's *Unix in the Office* will address
Sybase System 10.

For reprint information on articles appearing in this issue,
please contact Donald Baillargeon at (617) 742-5200, extension 117.

X/Open's XPG4

Light in the Open Systems Tunnel

XPG Is X/Open's Crown Jewel

While X/Open performs many valuable services and functions on behalf of a diverse set of stakeholders, its most influential products are the XPG and related verification and branding programs (See *Open Information Systems,* Vol. 8, No. 1, January 1993, for more background on X/Open). XPG3 has provided guidance to the industry for over three years, and vendors as well as users have eagerly awaited the publication of XPG4. Though it is too early to gauge its market impact (it was published in October 1992), the substance of XPG4 does address issues such as interoperability, heterogeneous systems management, integration of legacy data, and the packaging of technology for general purchase, which are of deep concern to users trying to implement open systems. With XPG4, X/Open might be positioned to bring the open systems splinters together and effectively slow the Microsoft juggernaut.

XPG4: The Next Generation

XPG4 Builds on, yet Departs from, XPG3

XPG4 incorporates the substance of XPG but transcends the XPG3 form by being more extensible. Although XPG5 may be needed someday, XPG4 has been designed to accommodate years of additional input from the X/Open technical working groups, with particular focus on the emerging distributed computing technology areas of the 1992 Common Applications Environment (CAE), including system management, object management, and online transaction processing (OLTP).

The XPG4 Shift: From Portability to Interoperability

XPG4 includes almost four times as many components as were specified in 1985's XPG1 and nearly twice the number found in XPG3. (See Table 1.) Far more important than the increased quantity of components is the fact that XPG4 signals a major shift in focus from application portability to the more complex topic of interoperability. It addresses issues of preserving investment in legacy systems and establishing the building blocks for heterogeneous enterprise computing.

It is important to note that X/Open has assiduously maintained forward compatibility in growing the XPG. Table 1 shows the additive nature of the XPG component set through successive versions. In detailing the migration from one version of XPG to the next, X/Open carefully considers and specifies component migration issues. (See sidebar, "Sample Component Interface Description.")

XPG4 Component Verification and Branding

Of the 22 brandable components defined in XPG4, test suites exist for 9 of them: Internationalized System Calls and Libraries, all of the languages, ISAM, X Window System Display, and X Window System Application Interface. In addition, test tools are currently under development for NFS and the Commands and Utilities components. As the demand for test suites outstrips the supply, X/Open could face a crisis of confidence in the value of its brand. (See *Open Information Systems*, Vol. 8, No. 1, for more detail on the status of X/Open branding and verification.)

XPG4 Has Some New, Some Enhanced Components

X/Open not only develops new components for each new issue of the XPG, it also revisits existing components both to ensure alignment of the specifications with new developments from other standards groups and to enhance the components with features that users need.

Of the 22 XPG4 components, 9 are new and many of the existing components have been updated. Table 2 shows highlights of the changes.

Probably the most outstanding example in XPG4 of the ascendance of users in X/Open is the appearance of IBM's Common Programming Interface for Communications (CPI-C). CPI-C provides an applications programming interface (API) that external applications can use to access data on mainframes running IBM's System Network Architecture Logical Unit 6.2 (SNA LU6.2) protocol for peer-to-peer application networking. The focus on legacy data implied in that effort speaks volumes about the real-life needs of users.

Evolution of the X/Open XPG

Component Name	XPG1 1985	XPG2 1986	XPG3 1987	XPG4 1992
Internationalized System Calls & Libraries	X	X	X	X
C Language	X	X	X	X
COBOL Language	X	X	X	X
FORTRAN Language	X	X	X	X
ISAM	X	X	X	X
Magnetic Media	X	X	X	X
Commands and Utilities		X	X	X
Pascal Language		X	X	X
Terminal Interface		X	X	X
Relational Database		X	X	X
Ada Language			X	X
X Window System Application Interface			X	X
Transport Service (XTI)			X	X
X Window System Display				X
Byte Stream File Transfer (BSFT)				X
X.400 Gateway				X
X.400 Message Access				X
Directory Access				X
Network File System				X
Common Programming Interface-Communications (CPI-C)				X
PC-NFS Server				X
LMX Server				X

Table 1. In its first seven years, the XPG has evolved steadily in breadth and fundamental structure. XPG4 has been packaged as an extensible body of interface, format, and protocol specifications that have been organized into functional areas and, beyond that, into Profiles. Profiles represent the first attempt to package the specifications into entities that resemble systems and subsystems that users might purchase.

Sample Component Interface Description

XPG4 X.400 Gateway

X/Open is committed to ensuring that each component be defined rigorously so it can be integrated into the CAE framework. The four generic interfaces to each component (see "X/Open in the 1990s," *Open Information Systems Vol. 8, No. 1,* Illustration 3, page 11) cover all the requirements for backwards compatibility with the CAE, though not all interfaces apply to each component. The X.400 Gateway Component definition below, excerpted from the XPG4 "Component Definitions" chapter, General Interworking section, illustrates a typical component specification summary.

Name:	**XPG4 X.400 Gateway**
Label for Logo:	XPG4 Component
Description:	The X.400 Gateway component enables the construction of portable gateway applications that allow messages from one messaging (e.g. electronic mail) system to be converted and forwarded according to the formats and conventions of another messaging system. Usage is not restricted to electronic mail since the interfaces and protocols may equally be used to convey messages containing arbitrary binary data, e.g. electronic data interchange (EDI). Usage is not restricted to X.400 OSI networks as the programming interfaces are equally applicable to other protocols.
Migration:	New for XPG4
Overriding Standards:	None.

HUMAN/COMPUTER INTERFACE: Not defined.

PORTABILITY INTERFACE

Specification Interface:	Conforms to the requirements of the Message Transfer (MT) interface as stated in the conformance clauses in X/Open CAE Specification, API to Electronic Mail (X.400). The conformance clauses also reference X/Open CAE Specification, OSI-Abstract-Data Manipulation API (XOM).
Portability Environment:	XPG3 or XPG4 branded Internationalized System Calls and Libraries
Indicator of Compliance:	None planned. The test environment must consist of two systems communicating using protocol stacks declared in the Conformance Statement Questionnaire.

INTERCHANGE FORMATS: Not Defined

COMMUNICATIONS INTERFACE

There are International Standardized Profiles for X.400, but these refer only to X.400 (1988) for which there are few conforming products available. Products are widely available for the European Functional Standards for X.400 (1984). Products conforming to the XPG4 X.400 Gateway component definition shall be available in a configuration that supports at least one of the following X.400 profiles:

1. MHS (1984): Interpersonal Messaging UA +MTA: PRMD/ADMD to ADMD (P2 and P1): M-IT-02 Profile: A/311; CEN/CENELEC Functional Standard: ENV41 202.
2. MHS (1984): Interpersonal Messaging UA +MTA: PRMD to PRMD (P2 and P1): M-IT-02 Profile: A/3211; CEN/CENELEC Functional Standard: ENV41 201.

1988 ISP references will be added when approved. The Conformance Statement Questionnaire provides an opportunity to declare support for other protocols.

Overview of New or Enhanced Components in XPG

Operating Systems and Languages	
Component Name	**Description**
XPG4 Internationalized System Calls and Libraries *(Enhanced)*	Operating system kernel and header files specifications are updated to match the NIST FIPS 151-2 draft. The XPG3 Inter-process Communications component incorporated, as has *widechar* internationalization.
XPG4 Commands and Utilities *(Enhanced)*	Commands and utilities are not actually updated. Instead, parallel specifications provide either "historic compatibility" or "POSIX compatibility" (1003.2).
XPG4 COBOL *(Enhanced)*	Extended with ISO COBOL features for internationalization, interactive screen handling, file sharing, and file locking.
Data Management	
XPG4 Relational Database *(Enhanced)*	XPG4 SQL aligns with the ANSI/ISO SQL2 standard-in-process. It supports embedded SQL in C and COBOL as well as dynamic SQL. This component includes work co-developed with the SQL Access Group to support distributed data management and the XA transaction monitor /database interface. interface.
User Interface	
XPG4 Windowing System Display *(New)*	Enables remote applications to access local display services. Can include X servers running on a workstation, PC, or X terminal.
XPG4 Window System Application Interface *(Enhanced)*	This update includes the X11, release 4 Xlib update and new specifications for X Protocol, ICCCM, and Xt Intrinsics.
General Interworking	
XPG4 Byte Stream File Transfer (BSFT) (*New*)	Supports the initiation of file transfer using Internet ftp over OSI FTAM, and optionally supports the responder role to a BSFT requestor. Facilitates migration from TCP/IP to OSI.
XPG4 X.400 Gateway (*New*)	Enables building applications which can convert and forward electronic mail or other messages between systems with different messaging formats and conventions. Designed for text or electronic data interchange (EDI).
X.400 Message Access (*New*)	Enables building X.400-based electronic messaging systems by providing message transfer services to applications using a user agent or message store. As with the X.400 Gateway, this component can be used for EDI applications.
XPG X.400 Directory Access (*New*)	Provides directory services through an API which can access an OSI X.500 or proprietary naming directory.
XPG4 Transport Service (XTI) *(Enhanced)*	Provides a protocol-stack independent API . NetBIOS has been added to existing support for OSI, TCP/IP, UDP/IP.
XPG4 Network File System (*New*)	New as a component in XPG4, previously part of the CAE file system. Offers the standard remote file services of NFS. Updated for network error messages.
Mainframe Interworking	
XPG4 CPI-C (*New*)	Provides an API for communication with systems running SNA LU6.2.
PC Interworking	
XPG4 LMX Server (*New*)	Defines the LMX protocols to offer file and print services for DOS PCs.
XPG4 (PC) NFS Server (*New*)	Defines the NFS protocols to offer file and print services for DOS PCs.

Table 2. Proof that the XPG is a living document is that as many components from previous XPGs have been enhanced as new components have been added.

The Bold Proposal of Profiles in XPG4

Specifications Produce Components—Components Produce Profiles

XPG4 goes beyond the XPG3 to group individual components into *profiles.* Profiles incorporate components into logical sets that are designed to resemble what users want to buy in the marketplace. Just as components group specifications into brandable entities, profiles group branded components into higher-order brandable entities. The limited usefulness of components outside of the technical community has been discussed for over three years. The appearance of profiles in the XPG4 clearly illustrates the increased user perspective in X/Open. XPG4 defines six profiles, five of which consist of fully defined brandable components. (See Table 3.) The sixth profile, Distributed Computing, will be completed in 1993 and will include pieces of DCE and object management. The XPG4 Base

Profile consists of the same components as did the XPG3 Base System, and it serves as the basis for all the other Profiles.

Branded Components Do Not Automatically Aggregate into Profiles

Profile definitions and rules for verification and branding follow the stringent model of components. In order for a profile to be branded, all of its constituent components must be branded. It is important to note that component branding occurs on a specific system platform. Under the profile branding rules, a profile-compliant set of components, even if branded on a common system platform, does not automatically qualify as a branded profile. In theory, integration-testing is required. But, at this point in the program, the only testable profile is the Base Profile, because it was testable in the XPG3 as the Base System.

Profiles Could Become a Big Win for Users

The profile-branding proposal is accompanied by a stringent set of agreements that obligate the vendor to provide all the components described in the profile-branding application and to fix, support, and sell the branded profile as long as the brand is in effect. The significance of profile-branding to users is very narrow in one sense because the branded profile describes a specific set of branded components running in a specific configuration on a specific branded platform. The buyer cannot substitute equivalent branded components into a profile and still enjoy the power of the warranty that ensures that the profile will function as a system. In another sense, the significance to users could be very broad, signaling the beginning of a new era in open systems where the formula shifts away from "open systems = piece parts + integration services + risk" to a more reasonable environment for the nontechnical buyer.

Profiles Could Speed the Adoption of New Technologies

Profiles could become the "reverse Trojan horse" of the open systems movement. By being forced to incur the expense of integrating components, vendors may become aware of the benefits of making future components easier to integrate by design. The penetration of object orientation, which offers more abstract interfaces than procedural interfaces, could be accelerated because components with object-oriented interfaces will integrate more easily with one another.

XPG4 Profiles in Perspective

Profiles have evoked skepticism from vendors who sympathize with the sentiment expressed by users that the time has arrived for them to be able to "buy the car—not the car parts." The vendors question the usefulness of the first generation of profiles, raising issues such as who sells the components vs. who supports the profile, and why integration tests are required for components for which interfaces have been rigorously certified.

We consider the 1992-93 proposal of profiles to be a trial balloon to draw response and comment from the industry, which will, in time, help define requirements more fully. By backing profiles through the X/Open process, users have signaled to vendors that they are serious about wanting to be able to build open systems without the expensive hand-holding that vendors plan to offer to bolster their bottom lines. Profiles could also represent a threat to the well-earned reputation associated with the X/Open brand. Today, users see X/Open as a limited but safe haven from vendor hype and promises of open systems nirvana. The availability of profile integration tests is likely to trail the availability of profiles, which will probably force X/Open into reliance on the vendor conformance claims that are submitted as part of the Conformance Statement Questionnaires (CSQs) used in the branding process. The dependence on the vendor honor system could compromise the reputation of X/Open and end up disenchanting users, instead of helping them travel further down the road to open systems utopia.

XPG4: Where to from Here?

Filling in the CAE Blocks

The 1992 CAE sets up three major areas of future work for X/Open: system management, object management, and OLTP. XPG4 has been designed to incorporate other ongoing work as well as these areas over the next two years or more. In each of these areas, the skills of X/Open as standards integrator will be tested in melding the work of OMG, OSF, OSI and other groups where standards exist but have not yet been commercialized. In some cases, such as system management, X/Open appears to be following two paths, one based on OSI and the other based on OMG objects, with a convergence and migration path that will be determined in the future.

In 1993, the X/Open collaboration with OSF should bring a DCE-based profile to completion. Despite the membership flap, X/Open has maintained a working relationship with OSF at the direction of its board.

Overview of XPG4 Profiles

Profile Name	Components	
Base	System Calls and Libraries Commands and Utilities C Language	
OSI Communications Gateway	System Calls and Libraries Commands and Utilities C Language X.400 Message Access	Transport Service X.400 Gateway Directory Access BSFT
Base Server	System Calls and Libraries Commands and Utilities C Language LMX Server X Window System Application Interface	Transport Service Terminal Interface Directory Access BSFT NFS PC NFS Server
Workstation	System Calls and Libraries Commands and Utilities C Language X Window System Application Interface	Transport Service Terminal Interface Directory Access NFS
Database Platform	System Calls and Libraries Commands and Utilities C Language	Transport Service Relational Database COBOL

Table 3. XPG4 profiles use common components to built application-specific profiles. The key will be developing test suites for the integrated products.

X/Open and OSF's DCE: A Case Study in Process

Early in 1991, months in advance of early source code deliveries to system vendors, the OSF DCE began to populate RFPs in the private and public sector and to appear on user-driven architectures and frameworks from activists such as the Group of Ten and POSC. Users in X/Open militated strongly to get DCE technology out of the hands of special interests and into the public process of X/Open. This groundswell convinced the X/Open Board of Directors to charter an effort to "adopt in principle" the DCE as the basis for an X/Open distributed computing profile. X/Open began working with OSF on a "fast track" basis, while DCE was still in an immature stage, to quickly identify issues and comments related to CAE integration and to bring discrepancies to a head quickly. OSF retains the option to withdraw from the process if it appears that shareholders, such as Sun, begin to interfere with adoption. So far, the process has unfolded smoothly.

Two DCE Pieces Down, Four to Go

Of the six technology pieces in DCE, the X.500 directory services already existed in the CAE because OSF had adopted its use from X/Open, which left five pieces on the table for approval: RPC, time services, distributed file system, network security, and naming.

X/Open chose to tackle the RPC first, which represents about half of the work, and has already brought it through the approval process with only a few minor comments. The schedule calls for each of the other elements to be completed at two-month intervals. At this time, federated naming is not part of the specification under consideration for the distributed computing profile.

But Even X/Open Can't Guarantee DCE Interoperability

Despite the involvement of X/Open with DCE, the results of the effort will not guarantee cross-vendor interoperability of branded DCE implementations due to the diversity of transport protocols that will be used to enable the DCE RPC. The X/Open method of dealing with this issue is to accept the fact that the RPC transports could be TCP/IP, NetBEUI, SNA, or OSI, and to rely on the vendor CSQs to arm buyers with the information needed to address cross-vendor interoperability. The CSQs are filed as part of the branding application. For the distributed computing profile, the CSQ will require vendors to specify in detail the communications protocol stack that will be implemented below DCE, thereby enabling the buyer to match up the CSQs for compatible transports that support interoperability across implementations. While X/Open cannot assure universal interoperability of branded DCE implementations, users, vendors, and OSF all benefit greatly from the transition of DCE from source code product to a managed set of specifications.

Conclusion: The Future Is Now

XPG4 is about the future. With open systems, the duality of supply and demand has an additional element: the "what to demand." XPG tells open systems users "what to demand." The market is now in a fertile state for XPG to have a major impact on RFPs and purchasing patterns, reshaping the industry over the coming years. The X/Open branding program under XPG4 will be stressed to cover new components and profiles, but this critically important concept must remain credible in order to support the continued penetration of open systems in the mainstream commercial market. X/Open has driven the XPG4 stake in the ground well beyond today's market. Now it must lead the way there. ◐

Information about how to order the XPG4 documents can be obtained by contacting one of the following X/Open offices:

United States: 703-876-0044 (Virginia), 415-323-7992 (California)
Europe: 44 73 450-8311 (United Kingdom)
Asia Pacific: 81 33 251-8321 (Tokyo, Japan)

Open Systems: Analysis, Issues, & Opinions

FOCUS: APPLICATION DEVELOPMENT

SmartStar Vision: Focusing on the Business Rules

SmartStar Vision is a cross-platform, rapid application development tool from SmartStar Corporation (Goleta, California) which is built on the assumption that a corporate developer should have to write only the code that implements the business rules for an application. The company's goal is to eliminate the need to have a developer write code for designing and managing the user interface or for making the connections among the business rules, the user interface, and the database.

SmartStar Vision is also aimed squarely at the demand for tools that support development of client/server applications. It is an object-oriented environment that supports the development of applications that will be deployed on heterogeneous desktops and back-end servers. It accomplishes its objective of minimizing coding by providing the developer with commonly used application objects whose attributes can be set without programming and with a 4GL for coding the business rules.

Company Background

SmartStar Corporation was founded as Signal Technology in 1977. It changed its name in 1989 to SmartStar, the name of its leading product, which is a 4GL for multiple SQL databases on the Digital Equipment Corporation VAX/VMS platform. SmartStar has had a number of product firsts over the years in areas as varied as digital signal processing (ILS), VAX resource accounting (PACS Plus), and a 4GL interface to Britton Lee's database machine (Omnibase). SmartStar VMS was first introduced in 1984.

The company became SmartStar when it decided to focus on tools for application development, seeing as its mission the need to provide high-productivity tools for accessing multiple databases in distributed environments.

Key Features of SmartStar Vision

The internal architecture of SmartStar Vision is object oriented, but it does not impose an object-oriented paradigm on programmers. This is an explicit attempt to appeal to corporate developers, who are likely to be more familiar with traditional 4GL and SQL development environments than with object-oriented concepts.

A SmartStar application is composed of objects contained in an Application Repository and a logical database dictionary (LDB). (See Illustration A.) These are the focal points of the Vision architecture. All window objects and SmartGL reusable functions are stored in the Application Repository, while all metadata regarding the structure of the database are stored in the LDB. The LDB has multiple interfaces that can be used for design and maintenance. It can be maintained either with SmartStar Vision's SQL Windows administration facility, by importing ANSI-SQL files, or with any CASE tool that can generate ANSI SQL-92 entity/relationship diagrams.

HETEROGENEOUS DATABASE SUPPORT. Since the LDB represents a logical view of the data, it can support heterogeneous databases at the same time. One unique feature is its ability to support application-level referential integrity among multiple heterogeneous databases without requiring a transaction monitor.

Reusable functions are written in the SmartGL 4GL language stored in the Application Repository. SmartGL can be used to write SQL, to call 3GL functions, to set values for objects, or to produce reports. Business rules are coded into the application in the SmartGL language.

DYNAMIC BINDING. SmartStar takes a very aggressive approach to dynamic binding in the Vision product. The company believes that its dynamic approach to objects gives it many advantages, including easier support for both heterogeneous platforms and mixed GUIs. In a networked environment, the SmartStar Vision application resides on a server. When the developer modifies the application, SmartStar's dynamic binding capability enables any user on any platform to immediately run the modified application, eliminating compiling and linking efforts for each platform. This runtime binding is referred to as *late binding*.

Although the early binding school emphasizes the performance penalty paid by late binding, the fact is that more and more performance is available for less and less money. This makes the performance penalty argument much less tenable.

SmartStar Vision Components

SmartStar Vision extends many of the back-end capabilities of the SmartStar VMS product to graphical front ends. (SmartStar VMS Version 6.2 is scheduled for release in March 1993.) In fact, SmartStar VMS applications can be accessed from Vision applications. The product is composed of the following components:

- **Developer Facility.** The application developer's main environment is the developer facility. It is a graphical, windowed environment, currently with a Motif interface.
- **Window Editor**. The window editor appears as a separate window for editing interface objects, such as fonts, colors, and other objects. The editor includes an object hierarchy browser for displaying and editing both visible and invisible objects.
- **Interactive SQL Facility.** The interactive SQL facility is provided for use by users and developers to construct complex, ANSI SQL-compliant operations. It has been designed with broad functionality, with extensions to ANSI SQL for functions like creating reports and macros, and for obtaining statistics. It can also be used to access database-specific capabilities such as stored procedures.
- **SmartGL Language.** SmartGL is a 4GL that is based on SQL and is used for manipulating objects, sending messages to objects, executing general purpose logical operations, and executing both standard SQL and extended SQL commands. It can also make calls to 3GL functions.
- **SQL Windows Administration.** SQL Windows administration is a database administration facility that can be used to create a logical database and SQL database definitions without an external SQL database.
- **Object Debugger.** The object debugger is integrated with the development environment and supports inspecting, testing, debugging, modifying,

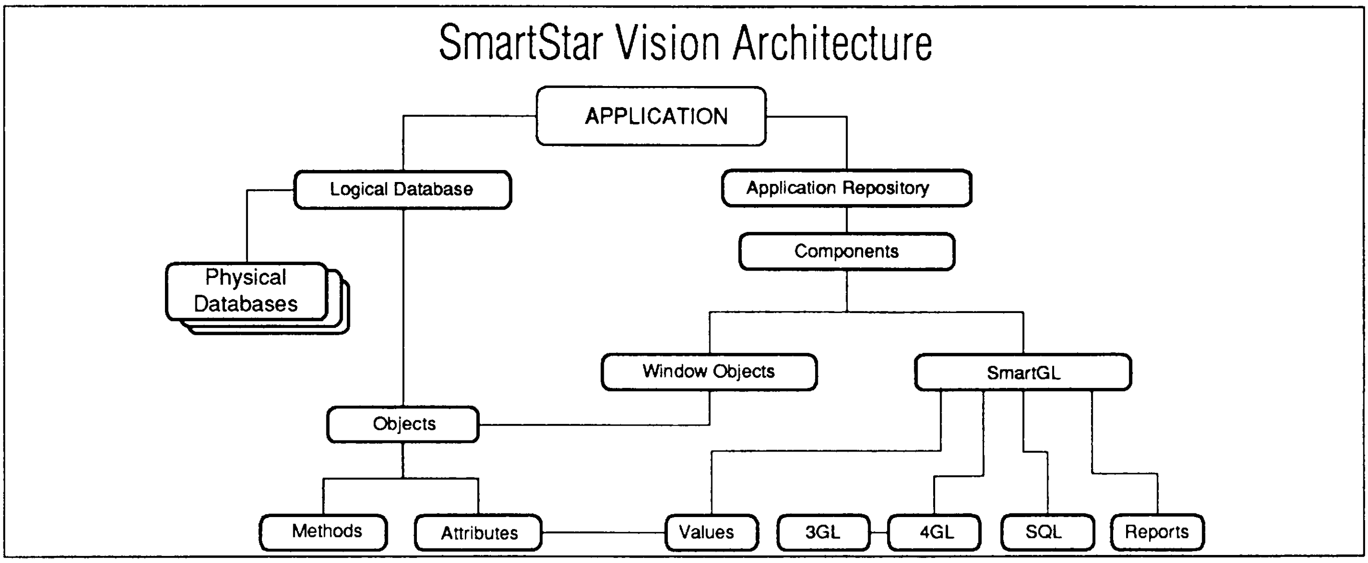

Illustration A. An application is made up of a logical database and an Application Repository. Objects can be defined from the logical database along with their methods and attributes. Attributes have values which determine the actual functionality of the application. The Application Repository consists of components that are defined either by window objects or by SmartGL. Window objects contain other objects, including database objects. Components can also be defined by SmartGL. SmartGL can contain values for objects and SQL, and can support 3GL as well as 4GL syntax. SmartGL also can be used to produce reports.

and saving changes in the runtime application.

- **Runtime SQL/RSQL Facility.** The runtime SQL/RSQL facility allows 3GL programmers to execute SmartStar Vision dynamic SQL calls.

- **Smart Report Painter.** The Smart report painter is an optional module that currently develops report procedures in character mode. These procedures can be stored as SmartGL functions in the Application Repository and executed by window objects.

- **Deployment Module.** The deployment module is the executable and runtime environment that accesses the application repositories on the server and the LDB dictionary. It allows access to the object debugger and may access the RSQL processor on any supported client or server system without modifications to the application.

These are all standard components of the SmartStar Vision environment. Clearly, this product is a complete, MIS-oriented product with a rich set of capabilities for corporate developers. It is aimed at mainstream application development.

SmartStar's View of Development

One of the primary goals of the SmartStar Vision development environment is to limit the programming required to build an application to the code necessary for implementing the business rules.

ATTACKING THE BACKLOG. The application backlog that virtually all companies face can be attributed to one four-letter word: code. Developing applications means writing code. To the extent that the amount of code that has to be written can be reduced, more applications can be written and be written faster. Reuse is one strategy for reducing coding. Eliminating coding is another. Developing graphical applications is a particular nightmare because of the amount of code that has to be written just to make the interface work, let alone to implement any of the data access procedures or business rules of the application. Although interface builders from a number of companies have helped out, interface builders still require a lot of code to be written before the GUI actually does anything. (See *Open Information Systems,* Vol. 7 No. 7 for more information on GUI builders.)

ATTACKING MAINTENANCE. Another problem that arises when an application requires extensive code to be written is the amount of maintenance that has to be done to that code. Every time the application is modified, changes may ripple throughout thousands of lines of code. Object-oriented development tools can help, but, when an object that has a lot of dependencies is changed, the maintenance effort can be substantial.

SmartStar has identified five areas where coding is normally required:

- The GUI, including the controls, menus, dialogues, and other interface objects.

- The rules to the GUI, which determine the behavior or action that the interface engages in when an event like a mouse-click or keystroke occurs.

- SQL statements that are executed to insert, retrieve, delete, or update data.

- Rules for SQL, which determine what is done with that data once they are retrieved, including where they are displayed, in what form, who has access to them, etc.

- The business rules, which determine just what the application actually does. To the extent that business rules are common within an industry, a vertical package can be developed. Business rules that are specific to a company must be expressed in some form of code or script. An example of a business rule might be:

  "If an order is over $50,000 and the customer has an outstanding balance over $10,000 and the customer placed less than $500,000 in orders last year and the user is not authorized to approve non-conforming orders, display a message to the user that the order cannot be confirmed until authorization has been approved. If the user is authorized to approve non-conforming orders, display a message to the user that the order is non-conforming and that the user's approval is required."

SmartStar Vision is designed to eliminate programmer coding from as many of the above five areas as possible. The ideal is that only the business rules are left for the professional corporate developer to code.

Development with Vision

Development with SmartStar Vision first requires the definition of the underlying database structure. This can be done with either the native DBMS tools or Vision's own tools. The next step is to design the windows for the application. A number of tools are provided that assist the developer in window design, including the window editor, tools editor, font editor, geometry editor, and object browser. (See Illustration B.)

An object debugger is provided for runtime debugging. A dynamic object engine is provided as an integral part of SmartStar Vision, and it allows any attribute of any object to be changed at runtime. This allows the object debugger to be used to test and modify any number of open application windows concurrently. Changes can be ignored or saved in the Application Repository.

APPLICATION WINDOW STYLES. One of the arduous programming tasks facing business developers using relational databases is the construction of forms that combine master views and detail views, including multiple master records, multiple detail records, master/detail, and master/multiple details. Building these views requires managing table joins, keys, data types, scaling, referential integrity, and primary adjunct table components, including full SQL syntax generation and processing. SmartStar Vision manages all of these functions, eliminating code-writing by the user. SmartStar Vision provides these views as window objects the developer uses in designing the application's windows.

SETTING OBJECT ATTRIBUTES. In the Vision environment, the behavior of objects is defined by their attributes and by the values of their attributes. Each object has an attribute window into which attribute values can be entered. Arbitrary behavior for an object can be included by using the SmartGL to code specific business rules and entering them as object attributes. SQL statements, conditional operators, validity checking, error testing, and access to object methods can all be represented this way using SmartGL.

SMARTGL LANGUAGE. SmartStar's SmartGL is a 4GL built on top of ANSI SQL which has been given a broad range of extensions that support functions including conditional operators, reading from and writing to field objects using SQL, and report writing using SQL. The syntax of SQL is further extended to include messaging capabilities allowing various SmartStar objects to request another SmartStar object to invoke a method.

SmartGL is the foundation for developing the business rules for the application. Routines can be created, edited and checked internally within the SmartGL facility, or used as reusable functions of window objects. The developer also has the option of calling 3GL routines from SmartGL, if that is necessary.

SmartGL's use of objects, methods, and attributes gives developers rich capabilities with minimal coding. For example, a developer may want to get some data from the database and display both a single master record and multiple related records in a scrollable "Detail" region in a window. This is the typical Customer + Orders situation. The developer would write the SmartGL statement "db.DoFetch();" as the "OnClick" attribute of a button object. When the user clicks the button, this code causes the database object to perform its DoFetch method, which retrieves data from the SQL database

SmartStar Vision Development Environment

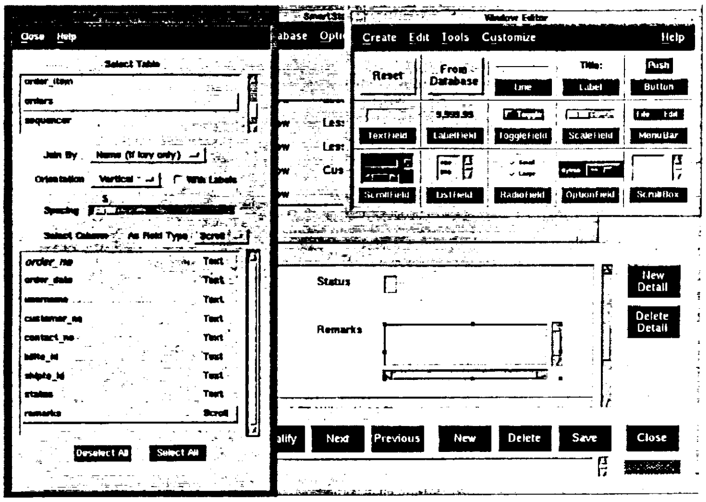

Illustration B. The SmartStar Vision development environment includes a window editor, tools editor, database object editor, font editor, geometry editor, and object browser.

into whatever views (in this case, a master view and a repeating detail view) the developer has created in the window. This method dynamically uses any record qualifiers the user has specified at runtime. The developer could further customize the DoFetch method by specifying an order in which records should be retrieved. This would be accomplished by setting the OrderBy attribute of either or both views using standard SQL syntax, ASC, or DESC.

In this way, the developer has managed the fairly complex behavior of bringing records into a view in a particular order using only a snippet of SQL syntax.

Release 1.0: A Good Start on Interoperability

The first release of SmartStar Vision came at the end of October 1992. Release 1.0 supports Sun SPARC, VAX VMS, and RISC Ultrix for both application deployment platforms and development servers. Clients supported are Motif for the SunOS, VMS, and RISC Ultrix environments, and X Window servers for MS Windows and Macintosh. The SQL databases that are supported, either in the base product or as options, are RdB, Oracle, Ingres, and Sybase SQL Server. SQL support is provided for CISAM on Unix and RMS on VMS. Access to DB2 can be achieved using Digital's VIDA gateway. Other approaches to getting at DB2 data are being explored, and support for other databases is being investigated.

Release 1.1, due in February 1993, is essentially a maintenance release, adding support for native window toolkit programming (OpenArea object), asynchronous event handling, enhancements to usability, and some new object attributes. It also includes enhancements to the LDB, such as table name synonyms and table and column labels.

A version for IBM's RS/6000 is imminent, once some final tweaking is complete. An HP-UX version should follow in a few months. Support for Digital's Alpha AXP machines will be on OpenVMS first. An Alpha OSF/1 port will depend on when Digital makes that operating system available.

Release 2.0 Will Expand Vision's Reach

SmartStar is committed to providing native MS Windows and Macintosh GUIs in the second release of Vision, due by the fourth quarter of 1993. The Windows development is being done in conjunction with the development work on a version for NT on Intel. The NT version will be both client and server, while Windows will only be a client. The two will be released simultaneously.

SmartStar Vision will support the NT SQL Server shortly after it is available. That support will be very similar to current support for Sybase SQL Server, since the NT version will be more like the Sybase version than the current Microsoft, 16-bit OS/2 SQL server is.

CROSS-PLATFORM DEPLOYMENT. Under Release 2.0, applications will be deployable from any development environment to any deployment environment, i.e., NT to Motif and vice versa, except where a developer has used SmartStar's OpenArea object to access native toolkits. That part of an application would have to be rewritten. Support for other NT platforms, including DEC AXP, MIPS, and Integraph, will depend on the level of demand that emerges.

MACINTOSH NOT IGNORED. Macintosh deployment will be limited to Digital's Pathworks for Macintosh environments. Connectivity is held forth as the issue here, although numerous reasonable TCP/IP implementations do exist for the Mac.

Positioning Vision Competitively

Conceptually, Vision's closest competitor is probably NextStep Version 3.0's DB Kit. (See *Open Information Systems,* Vol. 7, No. 10, October 1992.) Both are object oriented, use graphical design tools, minimize coding, and maximize productivity. But there the similarities end. Vision is already available on multiple platforms and will soon be available on many more. Vision also allows dynamic query qualification, which NextStep does not support. All qualifications (or filters) have to be coded in advance using DB Kit. Dynamic qualification means that developers don't have to code which qualifications are allowed in each application: Any qualifications are allowed unless they have been explicitly turned off.

PowerBuilder is also likely to be compared to Vision. However, this product from PowerSoft (Burlington, Massachusetts) supports only 16-bit APIs today for databases such as Microsoft OS/2 SQL server, and the Sybase NLM for Novell NetWare. Vision only supports full 32-bit interfaces, which should reduce conversion problems in the future. PowerBuilder is currently available only for Windows as a deployment platform, limiting its appeal for customers looking for maximum flexibility, and it does not allow dynamic qualification.

Perhaps the most direct competitor for SmartStar Vision is Uniface. (See *Unix in the Office,* Vol. 6, No. 6, June 1991.) Although Uniface is extremely popular and is gaining momentum, it lacks the object orientation and the dynamic nature of Vision. Application development in Uniface requires extensive 4GL programming.

Summary and Conclusions

The market is being flooded with client/server development tools today. Some are simply GUI builders, some are more complete environments. The ones which will succeed will be earmarked by two characteristics:

- They work as advertised.
- They solve real business problems economically.

SmartStar's long history as a supplier of application development tools bodes well for its ability to deliver products that work.

Its goal of restricting coding activities to the writing of business rules strikes at the heart of the challenge in object-oriented development today—identifying and defining business objects. The more Vision frees corporate developers from having to spend time with less relevant details of implementation, the more time they will be able to spend codifying business rules and developing business class libraries.

SmartStar is in a highly competitive segment of the market. Name and image are almost as important as technology. If there is a single challenge SmartStar must face, it is to be as effective a marketing company as it appears to be a technology company. —*M. Goulde*

PowerSoft Corporation 70 Blanchard Road Burlington, MA 01803	(617) 229-2200
Uniface Corporation 1420 Harbor Bay Parkway Suite 140 Alameda, CA 94501	(510) 748-6145
SmartStar Corporation 120 Cremona Drive P.O. Box 1950 Goleta, CA 93116-1950	(805) 685-8000

Patricia Seybold's

MARATHON WEEK

Actual User Case Studies

Come to the *Only* Conference for *Users* Migrating to Open Systems, Client/Server, and Distributed Object Computing

April 18 - 23, 1993 • Copley Marriott • Boston, MA

Distributed Object Computing Technology Forum

How Can I Use Distributed Object Computing Technology Today to Build Mission-Critical Applications?

Open Systems Forum

How Can Organizations Build Production Systems, Save Money, and Gain Interoperability with Open Systems?

Seybold Executive Forum℠

How Can Attendees Design a Real Business Solution That Implements Open Distributed Systems?

Who Should Attend Patricia Seybold's Marathon Week?

YOU should, if you fit any of these profiles:

- Are you implementing client/server solutions?
- Is your organization migrating from proprietary systems to open systems?
- Is your development organization interested in object-oriented approaches?
- Are you redesigning your business processes, re-engineering your organization, and streamlining workflows?
- Is your company tying its LANs together?
- Are you downsizing—moving applications from mainframes to networks?
- Are you designing an information architecture, selecting a database direction, or seeking document and image management solutions?
- Do you want to know how to leverage the next wave of technology—open distributed object computing?

3 Complete Technical Programs

Cross Participation Encouraged

To Register, Call Deb Hay at (617) 742-5200.

Patricia Seybold's Computer Industry Reports

ORDER FORM

Please start my subscription to:

		U.S.A.	Canada	Foreign
☐ *Patricia Seybold's Workgroup Computing Report*	12 issues per year	$385	$397	$409
☐ *Patricia Seybold's Open Information Systems*	12 issues per year	$495	$507	$519
☐ *Patricia Seybold's Distributed Computing Monitor*	12 issues per year	$495	$507	$519
☐ *Paradigm Shift—Patricia Seybold's Case Studies in Distributed Computing*	6 issues & tapes per year	$395	$407	$419
☐ *Paradigm Shift—Patricia Seybold's Case Studies in Distributed Computing*	6 issues per year	$295	$307	$319

Please send me a sample of: ☐ *Distributed Computing Monitor* ☐ *Open Information Systems* ☐ *Workgroup Computing Report* ☐ *Paradigm Shift—Case Studies in Distributed Computing*

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: ______________________________

Fax No.: ____________________ 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 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.

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

IOI-293

Topics covered in Patricia Seybold's Computer Industry Reports in 1991 & 1992:

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

Office Computing Report

1992—Volume 15

#	Date	Title
1	Jan.	The Groupware Phenomenon—Does It Focus on the Proper Issues?
2	Feb.	Digital's TeamLinks—A Renewed Focus on the Client Desktop
3	Mar.	Requirements for Workflow—What Should We Expect from the Vendors?
4	Apr.	Desktop Multimedia—Moving beyond the Chicken and the Egg
5	May	Borland International—A Database-Centric, Object-Oriented Approach to Desktop Integration
6	June	Apple's Macintosh—Can It Become "the Cadillac of Collaboration"?
7	July	Business Intelligence—A Framework for Data Analysis Applications
8	Aug.	The Quest for Common Mail APIs—Clearing up the Confusion
9	Sept.	BeyondMail for Windows—Epitomizing the Mail-Enabled Application
10	Oct.	Microsoft's Workgroup Strategy—Moving Group Functionality into Windows
11	Nov.	Visual Programming—Application Design for End Users and Professional Developers
12	Dec.	The Notes Phenomenon—The Industry Reacts to Lotus Notes

1993—Volume 16

#	Date	Title
1	Jan.	Microsoft Access—"Cirrus" Database Project Gets down to Earth

UNIX in the Office

1992—Volume 2

#	Date	Title
2	Feb.	System V.4 and OSF/1—Matching up in the Marketplace
3	Mar.	Europe's Harness Project—Integrated Technology for an Open, Object-Oriented, Distributed Applications Platform
4	Apr.	The X Window System—Where is Its Future?
5	May	HP's Master Plan—Winning Is Everything in Palo Alto
6	June	Digital's DECworld Gems—Alpha and Accessworks Shine

Open Information Systems

#	Date	Title
7	July	Integrating Applications in the Real World—Evolution, Not Revolution
8	Aug.	Windows NT 3.1—Microsoft's Bid for Desktop Dominance
9	Sept.	Oracle's Version 7—Can It Leapfrog the Competition?
10	Oct.	Galaxy from Visix—Application Portability Breakthrough?
11	Nov.	European Open Systems Architectures—Europe's Vendors Strike out for Open Distributed Systems
12	Dec.	The Unix Data Center—Fact or Fiction?

1993—Volume 8

#	Date	Title
1	Jan.	X/Open in the 1990s—Making Open Systems Safe for Users

Network Monitor

1992—Volume 7

#	Date	Title
6	June	Distributed Printing—Major New Approaches Begin to Relieve One of Distributed Computing's Most Frustrating Problems

Distributed Computing Monitor

#	Date	Title
7	July	The New E-Mail APIs—Finally, Real Progress toward Mail-Enabled Applications
8	Aug.	Distributed Object Computing—The Merger of Distributed Computing and Object-Orientation into a New Architecture
9	Sept.	Gradient DCE for PCs—PC-DCE Spurs OSF to Make PCs Peers to Unix and Other More Powerful Platforms
10	Oct.	Database Interoperability—A Comprehensive Approach to Database Access for the 1990s
11	Nov	Transarc Encina—Will Distributed OLTP Systems Overrun the Mainframe's Last Stronghold?
12	Dec.	UI-Atlas Distributed Management—Object-Oriented, Distributed Management for the Unix System V World

1993—Volume 8

#	Date	Title
1	Jan.	Component Software—A Market Perspective on the Coming Revolution on Solutions Development