Patricia Seybold's Office Computing Group

Editor-in-Chief
Judith S. Hurwitz

INSIDE

UNIX IN THE OFFICE

Guide to Open Systems

Vol. 6, No. 5 • ISSN: 0887-3054 • May 1991

Clarity's Rapport

The Design of an Integrating Application

By Laure B. Rowan

IN BRIEF: The Rapport compound document processor and communication package is one of a handful of new-generation applications that double as integrators. The system incorporates data from external applications into its structure via a conversion architecture. Rapport converters have two roles: converting data into a Rapport format for compound document integration and editing, and mailing documents that are automatically converted into the preferred application format of the recipient. Essentially, the system allows users to integrate data from various applications into a single format for compound document processing and communication.

Report begins on page 3.

The Commodity RISC Desktop

There's More Than One Road to Victory

COMMODITY MARKETS ARE funny things. Vendors love and fear them. They dream of mass markets where they can price their products low and still make lots of money, but, at the same time, they know that it takes a combination of brilliant marketing and a lot of luck to come out on top. Who could have predicted 10 years ago that a simple personal computer made by IBM, running an unsophisticated operating system (or, shall we say, file system) and made out of readily available parts, would have ruled such a large portion of the desktop market? There was no way to predict and no way to prepare.

Today, we are at a new juncture. Vendors are desperately looking for the next DOS/PC market—a return to the time when things were simple. A time when there was one binary standard and one mass platform that everyone could love (or hate). This is part of the motivation of the various consortia that are developing binary standards for key hardware platforms and of the group led by MIPS, Compaq, etc. There are indeed good reasons for wanting a single operating environment for hardware platforms. One of the difficulties of Unix, for example, is that each vendor adds peripherals and moves information among applications in a different way. These differences make it very difficult for software developers to find the commodity markets where the real money is. A developer can't simply write the application once and then sell it to millions of eager users.

The key question remains: Can Unix and, in the future, an open systems environment produce a mass market for technology? The answer is: Maybe. Some critical obstacles must be overcome before victory can be declared. First, vendors must achieve consensus on the underlying services (at the very least, at the API level). Developers need a sanctioned group of APIs that they can write to regardless of who owns the underlying platform. Second, developers also need a consistent distributed computing environment so that heterogeneous desktops can interoperate without mass customization.

Then, even if these two critical issues are addressed to everyone's satisfaction, there is the Application Binary Interface (ABI) issue. Binary interfaces solve the application portability problems for a single hardware platform, but they are based on the assumption that the world is about to converge on one hardware technology—Intel, MIPS, SPARC. It is becoming clear that this industry is simply too competitive for one hardware technology to prevail as *the* RISC desktop. (This doesn't mean that the various players will stop trying to win dominance.) One bright prospect for intra-platform interoperability is OSF's application-neutral distribution format (ANDF) program. We expect that OSF will go ahead and announce a winning technology for ANDF. The good news is that this would allow for interoperability among platforms. It does not negate the necessity for ABIs; it complements them. The bad news is that ANDF will require an interim step for third parties: They will have to run their applications through an intermediate compiler.

The bottom line is that the simplification of the mass desktop market is just that—a simplification of some complex market dimensions. We do not believe that the industry will end up with only one implementation or one standard. We can assume, however, that some innovative software developers will find ways to mask the differences among implementations of desktop technology. We may finally get to a mass market for RISC desktops, but the path will be circuitous.

UNIX IN THE OFFICE

Editor-in-Chief
Judith S. Hurwitz

MCI:
JHurwitz
Internet:
jhurwitz@mcimail.com

Publisher
PATRICIA B. SEYBOLD

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

News Editor
DAVID S. MARSHAK

Managing Editor
DOUGLAS A. FREEMAN

Art Director
LAURINDA PHAKOS

Sales Director
RICHARD ALLSBROOK JR.

Circulation Manager
DEBORAH A. HAY

Customer Service Manager
DONALD K. BAILLARGEON

Patricia Seybold's
Office Computing Group
148 State Street, 7th Floor,
Boston, Massachusetts 02109
Telephone: (617) 742-5200
Fax: (617) 742-1028
MCI: PSOCG
Internet: psocg@mcimail.com
TELEX: 6503122583

Unix in the Office (ISSN 0890-4685) is published monthly for $495 (US), $507 (Canada), and $519 (Foreign) per year by Patricia Seybold's Office Computing 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 *Unix in the Office*, 148 State Street, 7th Floor, Boston, MA 02109.

Clarity's Rapport

The Design of an Integrating Application

In Search of Integration

Integrated Office always seemed like such a good idea. You have a bunch of practical business applications—word processor, mail, and spreadsheet—all of which are seamlessly integrated. Some systems are so seamless that they sport compound document architectures. And all your co-workers have the same system, so you can E-mail spreadsheets and documents right from within your office environment, and the recipients get the files in the same format you sent them. No trying to make sense of bizarre characters—remnants of an unreadable file. No stripping out weird formatting codes.

Standalone Applications: No Way to Integrate

Unfortunately, Integrated Office never quite took off. The proliferation of the PC liberated users from the closed office environment. With a PC, they could use whatever application they liked, and, over the years, they became attached to specific, standalone applications. Thus, the popularity of office systems—along with their promise of integration—faded. Oh, standalone applications might (or might not) have a clipboard mechanism for static data integration from other applications. But integration hasn't been a priority for most software vendors. Cutting and pasting ASCII text between applications is okay, but it's hardly sophisticated. Ideally, you'd like to be able to exchange live, editable data among applications.

Networks Make Integration More Crucial

Users have reached an impasse. They want their own applications, but they want integration, too. The need for integration seems more acute now than ever. This stems largely from the ever-emerging roles of networks and heterogeneous computing. You may still enjoy the independence of working with your own applications on your own machine, but now you can't afford to work in a vacuum. Not when you're trying to incorporate the marketing department's financial figures into your spreadsheet. Not when you need to E-mail a graphic in readable form. Not when you use a Unix workstation and your product manager wants to review your work on his Macintosh.

Some Systems Suppliers Offer Integration Platforms

You're probably looking to a systems vendor to help your organization over the integration hurdle. Indeed, application platforms such as Hewlett-Packard's NewWave and IBM's OfficeVision are attempts at integration. As luck would have it, however, neither looks terrifically viable at the moment. HP hasn't had the independent software vendor (ISV) support it expected for NewWave, and the functionality of OfficeVision/2 LAN is quite shallow and unsophisticated.

If you have a serious problem with hardware and software discrepancies, you may have even resorted to a systems integrator.

Integrating Application: Enter Rapport

One Solution: An Integrating Application

But for integration on a less ambitious scale, you need not look so far (nor quite so expensively). Sometimes the right application will solve many of your integration problems—and solve them now. This is the case with the recently-released Rapport application suite from Clarity Software.

Integrating Application: Enter Rapport

Rapport is what we call an integrating application. At first blush, it resembles a high-end office system, with a snazzy compound document architecture and modules for document processing, presentation/slide processing, spreadsheets, graphics, voice, and mail. (The presentation/slide module is an interesting twist; you don't hear much about slide packages in the Unix market.) But Rapport is not an office system. It's a compound document editor that accepts and integrates data from popular standalone applications. It was specifically designed to be used in conjunction with—rather than as a replacement for—your existing standalone products.

Rapport Integrates via Conversion

The key to Rapport's integration lies in its converter mechanism, which transparently translates Rapport data into external file formats (and vice versa)—including those of PC and Macintosh programs. Thus, the converter also gives the system an avenue for communication with PC and Mac users and applications. Once application data are converted into the Rapport format, the user can take advantage of the system's compound document functionality.

Bridging the Integration Gap

In essence, Rapport bridges the gap between the integration of traditional office systems and PC-style application independence. It's certainly a timely product, directly addressing the problem of integration while providing a compelling and useful set of PC-style applications. Rapport is one of a handful of integrating application products available in the Unix marketplace, and—as this report will illustrate—it's a very elegant example of the kind of productivity and flexibility that integrating applications provide.

Company Origins

Rapport was the darling of the UniForum show last January. Clarity Software seemed to come from nowhere with a product that immediately intrigued both the press and users alike (and the competition too, no doubt). The company is only a little over a year old (founded January 1990), which made us wonder just who these guys at Clarity were and how they conceived and developed the product so fast.

Assessing the Needs of Commercial Users

According to Howard Smith, Clarity's president and founder, the fundamentals of the product were conceptualized before the company was even born. For some time before forming Clarity, Smith identified some severe voids in the Unix marketplace: a lack of adequate horizontal business applications (the kind that characterize the PC market); no easy way to integrate third-party applications; few options for clean communication within a heterogenous (both hardware and software) computing environment. Rapport is an attempt to address all of these needs.

A Workgroup Perspective

PRODUCT PHILOSOPHY. Smith had been vice president of Engineering and, most recently, general manager of the Workgroup Products Division—both at Silicon Graphics. The latter position most certainly helped crystallize the nature of Rapport into a workgroup-oriented product. What's great about Rapport as a group tool is its application independency; its converters allow you and your co-workers to use whatever applications you like. (Well, sort of. The application must be supported by Rapport's converters, and, as we went to press, there were 29 such applications.) You can pull in data from different applications, and Rapport will convert it all into a single format for integration, editing, and manipulation. You can also E-mail data from different applications, and your recipient will receive the data in the application format he or she prefers. Furthermore, although Rapport is a Unix application, its converters extend to PC and Macintosh applications, so you can receive E-mail data from these platforms and use it in Rapport as well. All in all, the product offers a flexibility that we rarely see in an application.

Company Origins

Clarity Has a Narrow Opportunity

QUICK TO MARKET. We noted that Rapport was a timely product, because the industry—particularly the Unix industry—seems ripe for practical business software tools that also address application integration. Indeed, the timing of the product was crucial for Clarity. Rapport doesn't have a wide-open window of opportunity: Popular PC applications are being ported to Unix all the time, and some applications are being extended to communicate with other programs. Therefore, Clarity had to get its product out and in the running before some other vendor ran away with its potential market.

Small Programming Group Speeds Development

In terms of getting the product out the door quickly, Clarity has come up with a development model worth noting. The company started with a small group of what Smith considers very high-caliber software designers (all have a minimum of 10 years of experience), and the smallness of the group gave Clarity a responsiveness that larger vendors can't emulate, encumbered as they are with multiple product lines and backwards compatibility—not to mention dead weight and corporate bureaucracy. (It's no accident that IBM is struggling with OfficeVision and that Apple has postponed the release of MacOS 7.0 so many times.) A few programmers can match—or surpass—the output of a hundred as long as they are focused on a single technology and have room for creativity. Clarity is proof. The company employs less than 30 people; about one-third of them are developers.

Architecture

Compound Document Processing and Communication

Rapport is an object-oriented and modular suite of applications, largely written in C++ (and C), used for constructing and distributing compound documents. The system contains modules for text, tables (spreadsheets), image, graphics, voice, and electronic mail (including fax), and all of these modules can participate to form compound documents and multimedia presentation slides.

Built with Toolkit Independence

Rapport is both Motif- and OpenLook-compliant. These aren't separate product versions. Rather, Clarity has abstracted the interface objects of both toolkits into a single interface library layer that will map to OpenLook on OpenLook platforms and to Motif on Motif platforms. Reaching this toolkit independence is no small programmatic feat; only a few ISVs have cared to deal with both X-based user interface "standards." We're impressed with Clarity's efforts in this respect.

Unlimited Nesting Enhances Compound Document Editing

COMPOUND DOCUMENTS. As a compound document editor, Rapport is excellent. Its object-oriented architecture offers unlimited nesting of object components, so you can have a text object that contains a spreadsheet object that contains a graphic object that contains more text. No matter how many object levels deep you find yourself, you still have the full-featured editor for the object you're working in. It's very nicely implemented. The unlimited nesting makes it very sophisticated. Not all compound document architectures are this deep. For instance, Slate, a competitive compound document processor from BBN, is only one layer deep: You may embed a spreadsheet into a text object, but the spreadsheet cannot contain additional objects.

Rapport's Editor. Rapport's compound document editor supports all five of Rapport's media elements. Actually the text, graphics, image, spreadsheet, and voice modules are editors in their own right, and the compound document editor allows them to interact in a single document or presentation slide. Switching from one Rapport component to the next is seamless and simple. As soon as you select a component from the menu, the system gives you the appropriate menus and tools.

Rapport Objects Are Document Enclosures

OBJECTS. Document processing is Rapport's focal point, and thus Clarity considers Rapport's various component applications—as well as third-party applications—to be document enclosures. To Rapport, any data combined with their accompanying application is an object, and,

therefore, can be included (or enclosed) in a Rapport document. In order for Rapport to recognize an external application, the user must first register it—a simple dialog box procedure (see Illustration 1). At that point, the application may be invoked right from within a Rapport menu. You can then file, print, fax, or E-mail the resulting document.

Application Registration

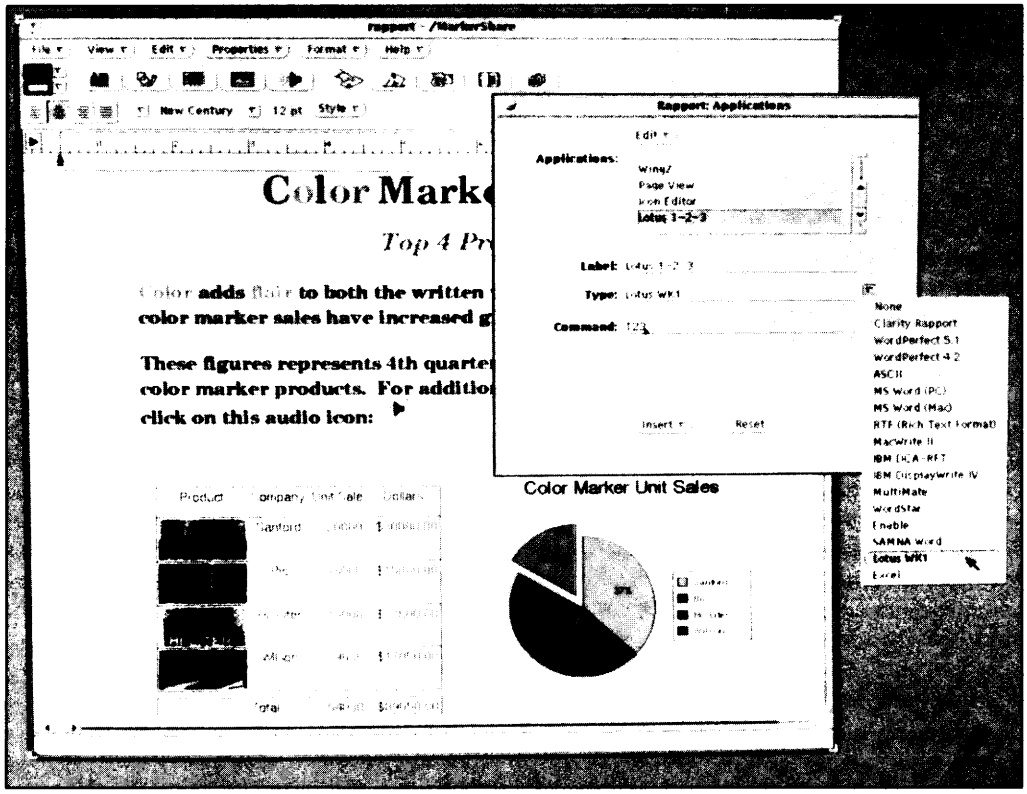

Illustration 1. Rapport uses a straightforward dialog box interface for registering third-party applications into the Rapport suite.

Two Levels of Integration: References and Conversion

But there's a difference between *including* a third-party enclosure and *integrating* a third-party enclosure. In other words, Rapport has two levels of third-party integration: references and conversion. To elaborate: At any point within a Rapport document, you may invoke another Unix application by selecting the application name from a menu, which places an inset box (or reference icon) within the document at a specified position. To run the application, you double-click on the icon. After finishing work with the external application, you may either convert the data into a Rapport-formatted object (see "Integration via Converters" below) or encapsulate—or reference—the data in its native format within the Rapport document. Afterwards, when you invoke the inset icon, Rapport will bring up the datafile and run its application in an X window. If the data have been converted, however, it has become a native Rapport object and is thus embedded directly into the document.

Referenced document enclosures are large-grain objects: entire documents, not sections or paragraphs, and whole spreadsheets, not individual columns or rows. Again, though, if the application data has been converted into a Rapport format, it can participate as a finer-grained object—right down to a single punctuation mark or individual spreadsheet cell—and take advantage of the system's compound document architecture.

Object-Sharing Limited to E-Mail

Local Object Management. Rapport has its own object management mechanism for keeping track of document objects and storing them in the Unix file system. It's a local facility; Clarity doesn't yet have any distributed or network object management services. So document and object sharing are limited to electronic mail. (That is, two users can't have access to the same Rapport object on a network; it can only be mailed.) This is too bad, because compound documents are meant to be shared, and mailing entire compound documents around can drain a system's resources. The company does plan to include distributed object management, perhaps even in the next release of Rapport. Ideally, Clarity would like to implement a standard solution, and the company is following the developments of the Object Management Group (OMG) in this respect. But the OMG isn't close to a standard solution, so, in the meantime, Clarity plans to implement a short-term strategy and then migrate when standards are available.

Translating Data into Rapport

INTEGRATION VIA CONVERTERS. Rapport converters transparently translate external data into Rapport's format and Rapport data into external formats. They play a crucial part in its architecture. Without them, Rapport would have no relationship with external applications. Converters are used both to integrate third-party application data into Rapport documents and to send and receive via E-mail third-party application data in user-prescribed formats. (If the user hasn't specified an application of choice, the converter defaults to ASCII.)

Converters Reach Mac and PC Applications

Not only do converters let Rapport users interact with other applications, they also let users interact with people on other platforms. Converters are available for popular PC and Mac applications as well as Unix applications (see Illustration 2). Clarity has licensed these products: The PC converter pack comes from KeyWord; the Macintosh converters from MasterSoft; and the Unix converters from Blueberry Software. These are separately-sold products.(A combination pack that includes the converters of all three environments is also available.) Your purchase depends on your organization's hardware configurations. Additional programs can be added to the packs, too, and Clarity has published Rapport's format in the hopes that software developers will write a Rapport converter program for their software. However, we imagine that it will be difficult for Clarity to convince many ISVs to write a Rapport converter. Clarity is such a small vendor that other application developers might not think it worth the effort. Instead, we think Clarity should continue nurturing its partnerships with the converter companies we listed above. They have the resources and the desire to do the converter development, and Clarity has already created a working relationship with them.

Converters Are a Good Start

Are Converters Enough? In a word: no. But they're certainly a good start. Some of Clarity's critics argue that converters are inappropriate for large-scale integration—that it's too difficult to write enough converters to support everyone's favorite application. While this is true, we think Clarity has started with a good set of often-used business applications. No, it doesn't cover all the applications out there, and frankly, we think that it needs to develop more. But, at this point, no integration solution covers all the applications out there.

Other integrating applications often use an extension language for linking modules to and bringing in data from external applications. (See "Applications as Integrators" page 20.) Extension languages may well offer more breadth of functionality, but they're also a lot harder to use. As we explained, converting external data with Rapport is simple, and mailing and receiving converted documents is completely transparent. We give Clarity credit for providing users with such an easy way to access outside applications. Keep in mind also that third parties can extend the converter packs to include additional applications. Furthermore, Clarity is planning to release its APIs later this year, which will offer a greater level of customization and integration for third parties. (See "Futures" below.)

Rapport Converters

PC	Macintosh	Unix
ASCII	ASCII	ASCII
WordPerfect	Microsoft Word	Cliq Word
IBM DCA-RFT	MacWrite II	Q-One
Microsoft Word	Write Now	SCO Lyrix
Enable	Lotus (.wk1)	Uniplex
Display Write IV	Excel	WordMarc
RTF		Interleaf
MultiMate		Frame
WordStar		Island Write
Samna Word		Lotus (.wk1)
Lotus (.wk1)		WordPerfect
Excel		

Illustration 2. Clarity uses converters to translate data between Rapport and other file formats and other platforms. Above is a partial list.

Converter Architecture

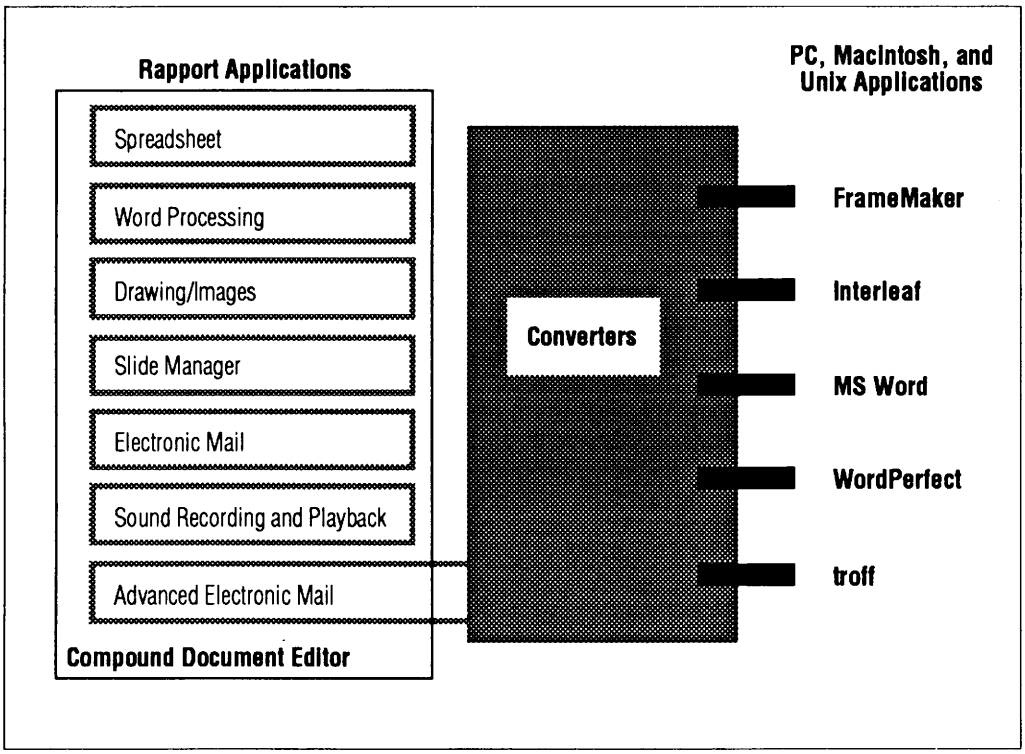

Illustration 3. An X.500 directory lets users create individual profiles indicating application preferences. The documents can then be sent via E-mail gateways and will end up on the recipient's desktop in his or her preferred application, retaining the text formatting and fonts.

Converter Interface

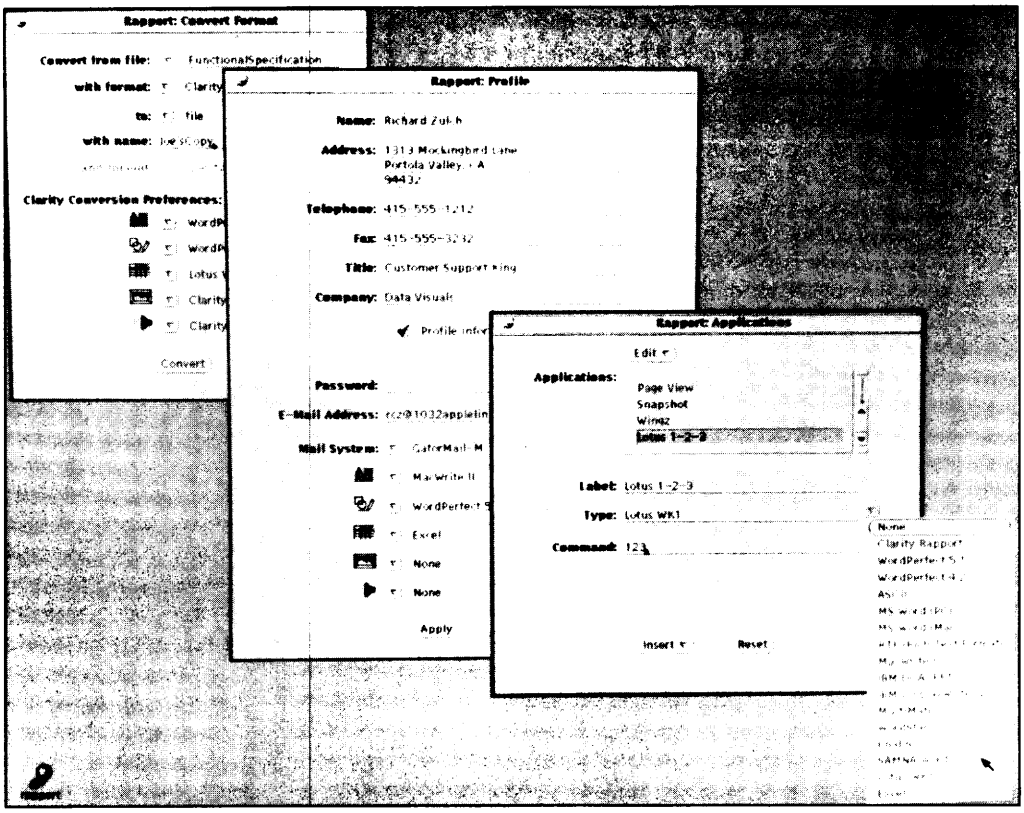

Illustration 4.

E-Mailing Editable Data

CROSS-PLATFORM COMMUNICATION. Rapport's mail system is much more than simple, text-based messaging systems. You can send live compound documents and editable document enclosures across various platforms and let your recipient receive information in the application format that suits him or her best. Let's say I compose a spreadsheet using Rapport and I want to send you the calculations, but you don't use Rapport. You use Lotus on the PC. Well, Rapport will send you that data in a Lotus format. Users or administrators can set up predefined user profiles in an X.500 global database that specifies users' preferred programs. Rapport simply converts the data to that format type.

Enclosures as Attachments

If you send a Rapport compound document to another Rapport user, he or she will receive it with all its compound document qualities intact. And he or she will have as much functionality to edit the document as you had when creating it. However, if you send a document that includes third-party enclosures, they will automatically be received as attachments—editable and in the preferred format of the recipient, but attachments nonetheless. We don't view this as a weakness, however. On the contrary, Rapport's automatic conversion makes the whole process as invisible and seamless as possible (especially given the various desktop platforms it supports). The only step missing is converting a Rapport compound document into the format of another Unix compound document editor.

Is Compound Document Translation a Reality?

However, compound document exchange and translation are a tricky business. In fact, we have yet to see a solution. The Office Document Architecture (ODA) compound document translation standard is close, but it's still immature—a subset of the functionality most compound documents contain. It doesn't include spreadsheets or equations, and the inflexibility of its style sheet mechanism has been criticized. Systems vendors like Digital Equipment and IBM are also pushing their respective compound document architectures as potential stan-

dards. Both IBM's MO:DCA and Digital's CDA are clearly more robust than ODA, but ODA has more support—especially in the European community, where it evolved. Therefore, neither vendor is ignoring it. Digital has an ODA gateway, and IBM is incorporating ODA into MO:DCA (the toolkit is not yet available). Clarity is placing its bets on Digital CDA, and plans CDA conformance by 1992. At that point, ODA support will be available via gateway.

Separating the Wheat from the Chaff

MAIL FILTERING. The other interesting part of Rapport's mail system is its filtering system, which lets you organize and prioritize the messages you receive. Rapport's is a rules-based filtering mechanism, continuously separating the valuable information from the garbage as electronic mail comes into your mailbox. Users only need to set up and fine-tune the rules in a personal profile. You can set up a filter that searches your incoming messages by sender's name or subject and then performs automatic mail procedures—e.g., forwarding a particular file or a message from a particular user, deleting specific messages, or mailing a reply to the sender. (See Illustration 5.)

Rapport's Message Filter Rule Editor

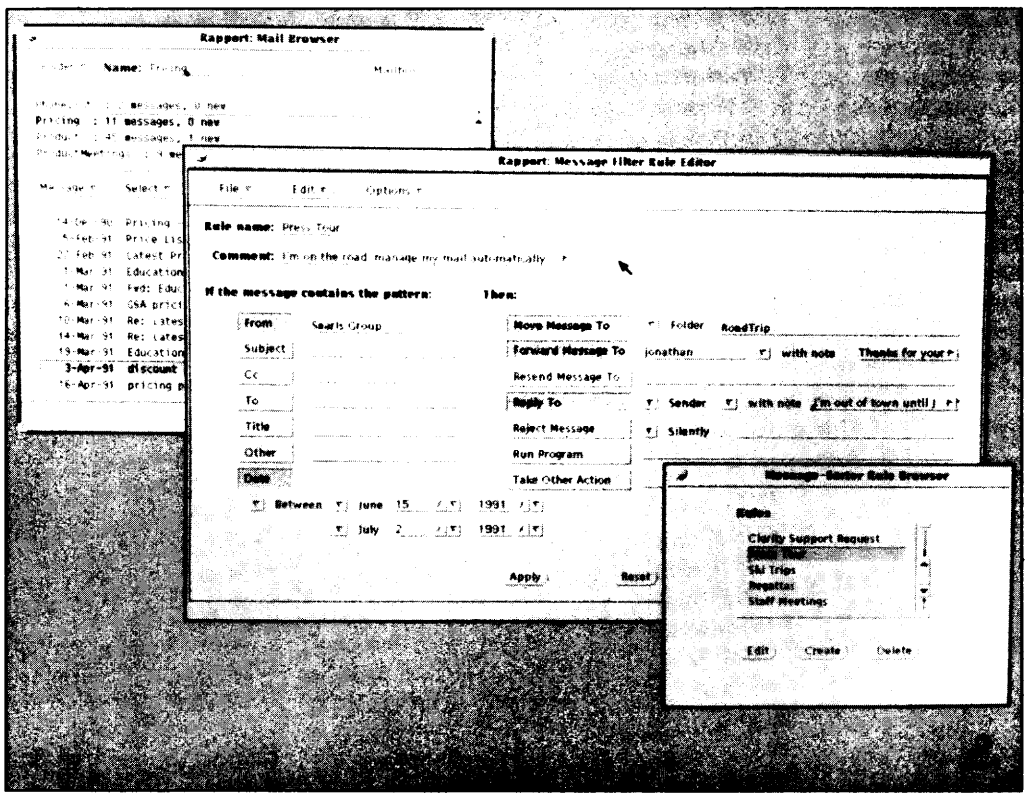

Illustration 5.

Mail filtering is becoming increasingly valuable as people become increasingly deluged with electronic data. Think about all your sources of electronic information: E-mail, computer conferences, news feeds, online databases, bulletin boards. Most of us just wade through it all, and what a waste of time that can be. You can read mail all day long and save perhaps five important messages. Rapport's filtering provides an important tool for turning the massive amounts of data into usable information.

Dabbling in Mail-Enabled Applications

Furthermore, Rapport's mail-filtering offers a degree of mail-based task automation. Clarity is on the brink of something exciting here. Currently, Rapport's mail filters are used to create automatic mailbox procedures on incoming messages. The point-and-click interface makes it

simple to automate routine mailbox options, such as moving messages to a named file or folder, forwarding messages, replying to messages, and deleting messages.

But automating mailbox functions is just half the story. Ultimately, Clarity envisions filters that spawn applications, retreive online data, and even provide communication between applications. What we're talking about is mail-enabled applications, where electronic mail transforms ordinary tasks and applications into network-aware applications. Consider, for a simple example, a calendaring application that sends meeting notifications directly to an online calendar instead of to a user. Or a report distribution application that reroutes urgent reports to another group member if the original recipient is absent. Or even database application that accepts data from an E-mailed spreadsheet. These are very simple, unsophisticated examples, but they give you an idea of how mail can be combined with other technologies to better leverage an application.

Clarity admits that it has only scratched the surface with filtering, but we think it's done a pretty good job so far. Rapport already lets you attach a named program to a mail message or even execute a Unix function, so it does allow you to go beyond automating the standard mail functions we listed above. And we expect Clarity to further advance its filtering development to allow for even more functionality.

Market Position

Conversion Architecture Distinguishes Rapport

We had a hard time type-casting Rapport. As we mentioned earlier, it seems like an office system at first. Not until you look at its integration capabilities do you see it for the compound document editor and cross-application communication package that it really is. The focus on file conversion sets Rapport apart from other compound document products like BBN's Slate system and Applix's Alis office system—although these are probably Rapport's closest competitors. Because of the similarities of its modules, Clarity may have a hard time differentiating Rapport from these products and even from more traditional office system products, like Uniplex. We recommend that Clarity be more vocal about Rapport's differences—plug it specifically as a compound document editor for document processing and presentation/slide management that works with existing software. Integration is the key, especially since its components are sometimes weak (see "Using Rapport Modules" below). Rapport is a complementary product: It uses and integrates your applications; it doesn't replace them.

Short Development Cycle: Help or Hindrance?

A QUESTION OF VIABILITY. We have already alluded to the productivity of Clarity's small group of developers. From one perspective, Rapport's short development life cycle is indeed remarkable. From another, it's cautionary, because Clarity may not seem as safe an investment for potential customers as other, more established software vendors. Ideally, we'd like to think that Clarity's innovative technology would overcome any such doubts.

Historically, though, innovation has not always been rewarded with lots of sales. Quite the contrary. Users are usually timid around unfamiliar technology, so new, leading-edge software often suffers in the marketplace. Up to now, the market hasn't been ready to buy compound document editors en masse. A full compound document product is probably too much for a typical user who works with maybe one or two applications all day. But perhaps compound document pioneers like BBN and Applix have paved the way for Clarity and other new compound document vendors by educating the market.

In addition, since Rapport is meant to be used with other applications, it has a different role than Slate or Alis, which are more solo-oriented applications. Alis even takes over your whole screen, making it a very closed environment. Furthermore, neither Slate nor Alis support Motif or OpenLook, which may have hurt them in the marketplace.

Marketing Strategy

Positioning Rapport as a PC-Style Application

RAPPORT AS HORIZONTAL SOFTWARE. Clarity is promoting Rapport as a horizontal software package—the kind of software tools that people usually turn to PCs for. Howard Smith contends that few Unix applications reach both traditional technical users and business users. He has a point. Too often, Unix workstations are used solely for specialized technical and engineering applications, while PCs and Macintoshes are used for business applications. Or worse yet, Unix users use overblown and often inappropriate Unix tools for business tasks. Some Unix techies tell us that they actually use high-end document publishing systems like Interleaf and FrameMaker for simple word processing. What an incredible waste of money! These packages can cost more than $15,000.

Boosting Workstation Sales?

Clarity thinks its product may even drive workstation sales. Rather ambitious, no? We do agree that the development of more horizontal Unix software tools should stimulate the workstation market. Unix versions of popular PC software like Lotus and WordPerfect should indeed help change the perception of engineering workstations to one of everyman's workstation. And Clarity can certainly capitalize on that growing market.

However, the current Unix market is still overwhelmingly technical—IDC estimated it as 92 percent in 1990 and 68 percent in 1994—and Clarity maintains that its applications are as appropriate for the technical user as they are for the commercial user. Especially the E-mail (for the internet and uucp junkies) and document processing (for all that technical documentation) components.

Distribution

Clarity is looking at four distribution channels for Rapport:

- Direct sales for key accounts. Clarity is counting on its relationships with hardware vendors (specifically Sun, HP, Digital, IBM, and Silicon Graphics) to generate direct sale prospects.
- Value-added reseller channels. Clarity is targeting the top 30 to 40 VARs of the hardware vendors listed above. A strong VAR focus could be very good for Clarity, since it's such a small company with a small marketing department.
- OEMs.
- Trial product campaign. As a sales promotion, Clarity is giving away Rapport on a free, 30-day trial basis. Trial customers receive a tape, loading instructions, and a tutorial. The company is encouraging its trial customers to lend the product out. Good idea, considering the Unix community's enthusiasm for shared ideas and technology.

Pricing

Rapport's pricing strategy is questionable. On one hand, it's comparable to other Unix-based compound document products, and, at $895 per seat, it's not at all excessive for a Unix application. But, on the other, Clarity is billing Rapport as a PC-like horizontal application, and a PC-level pricing strategy would bode favorably for the potential success of the product. Typical PC applications usually sell for about half the price of Rapport.

Of course, Clarity would counter that last remark by pointing out that Rapport is made up of several component applications, and if you were to purchase the standalone counterparts of all these components, you'd be shelling out twice the money Clarity charges for Rapport. But, none of Rapport's individual components would compete well with successful commercial applications. Rather, Rapport's main purpose is compound document processing and integration. Clarity claims that pricing has not hindered sales whatsoever, and we don't doubt that claim. But there's a difference between a price that doesn't hinder sales and one that actually

motivates sales. A truly aggressive, solo-product, PC-level price (around $500) could do wonders for Rapport's success.

Be Aggressive

MARKETING ADVICE. Our advice to Clarity is to be aggressive. Clarity will have to be a better marketer than BBN or Applix. We also recommend that Clarity be very vocal about Rapport's ability to integrate external applications and to communicate with Mac and PC users and applications. The industry may not bang the Clarity doors down to get compound documents, but they might to get distributed, heterogeneous computing, and Rapport has a very, very good story to tell there.

Using Rapport Modules

Primary Functions: Compound Document Processing and Distribution

As a compound document editor, Rapport has two basic functions: document processing and presentation/slide management. Both tasks can use all the Rapport object modules: text, spreadsheet, graphics (Clarity calls them Art objects), sound, and image (Picture objects). Mail is the third function, and it supports the compound document architecture of the Rapport application suite.

Components Can Be Weak...

As a general comment regarding functionality, Rapport's compound document editor gives the product some advanced features, but individual components are weak. Usually, we would slam a product on this point, and we do certainly advise Clarity to develop better functionality as its application set matures.

...But Can Work Well for Compound Document Editing

However, with the exception of the very casual user, few people will use a Rapport component as their standard application-of-choice. The product isn't being positioned that way. Clarity was not out to design full-blown applications. Rather, Rapport is an attempt to let you incorporate various forms of data into a compound document—something more than just an uneditable snapshot. Therefore, Clarity has assumed that users will continue with whatever applications they've always used, and turn to Rapport for tying those applications together in a compound document or for sending the information to users of different applications. In that way, Rapport components don't have to be as functional as their standalone counterparts; they just have to offer generic functionality for minor adjustments. In other words, a serious spreadsheeter will continue to rely on Lotus or Excel for power and depth of functionality, but she will convert the data into Rapport when she needs to couple that information with a graphic from another application and then do some number-tweaking. Or when she needs to send that data to a user of a different application.

We definitely agree with Clarity's converter strategy, and with the company's reasoning that people will remain faithful to their old applications. But we are concerned that the market won't buy into a mediocre application suite, especially if users have trouble differentiating Rapport from an office system (which is easy to do). Clarity is facing a real danger there, and must be very clear about its product positioning. Products like Alis and Slate have had a hard time finding an audience largely because of their shallow components. We don't want to see Rapport similarly suffer from an incorrect market perception.

Intuitive GUI

INTERFACE. Rapport has an intuitive and consistent windowing graphical user interface made up of icons, pop-up menus, and dialog boxes. In fact, the GUI is probably the most compelling feature of this set of applications. When we launched Rapport, we were greeted by a window with header panels of buttons for creating, opening, and formatting documents (see Illustration 6). It's all very simple and straightforward, and we rarely needed to turn to the manual. To open a file, we looked look in the Filing menu; to edit a file, we looked in the Editing menu; to change the line spacing, we clicked on the appropriate line-spacing button; to get Help, we clicked on the question mark.

A Rapport Window Header

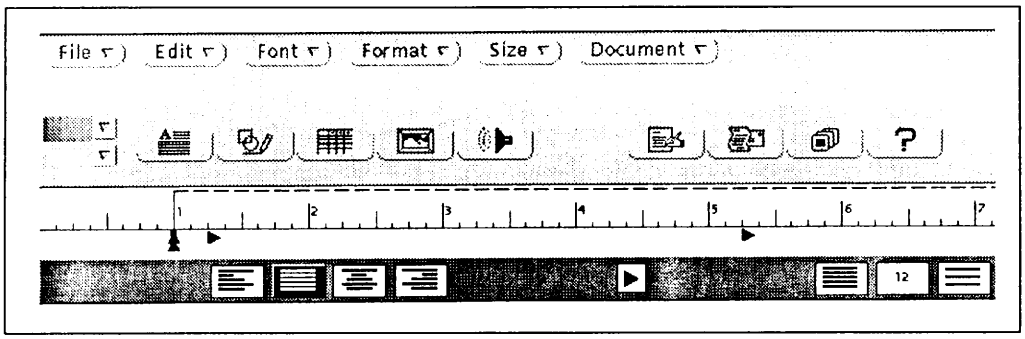

Illustration 6.

Room for Improvement

Interface Weaknesses. Although Rapport's point-and-click interface makes it easy to navigate around the system, Clarity would do well to add a few features. First would be some sort of macro facility for automating navigational procedures. Macros are a godsend, especially when you're dealing with a formulaic application like a spreadsheet. Many commercial applications are incorporating them. Alis, for example, has an exceptional, fully functional keystroke capture mode that creates macros that you can then edit. And Slate at least has a scripting language for automating document-related procedures.

Another valuable feature would be viewable keyboard bindings for those of us who don't like using mice. Rapport's documentation has a few pages of keyboard bindings, but the system itself doesn't show them—at least not the version we looked at. Apparently, this is a fault of OpenLook; its style guide doesn't allow for viewable keyboard bindings. They will be available for Motif.

We also took occasional offense at the system's unfriendly responses. For example, although Rapport has an easy-to-use dialog box for running a search, when we tried searching for a word that didn't exist in the document, the system told us that the "search failed." A message like this is not only curt; it's inaccurate. We'd rather that the system tell us what really happened—that the word we sought did not occur. We got that same "search failed" as a notification that a search was complete.

Providing Information and Flexibility

Help. Rapport's online Help facility is very good and very flexible. Invoking help is intuitive and context sensitive. We simply positioned the mouse pointer over the control we needed information about and hit the Help key. The system responds to your mouse pointer position and gives you the appropriate Help screens. The system also offers fine granularity of Help information. For instance, when we were in the Search and Replace panel, we could receive Help on each of the control buttons (i.e., Search, Search and Replace, Replace, Replace All) if we needed to better understand our options. The system also offers related Help items at the bottom of the screen to direct you further, which is always a nice touch, making it easier to reach the precise information you need in the shortest amount of time. You can also get Global Help by invoking the Help button (labeled by a question mark) on the Rapport control panel.

Powerful Editor; Generic Modules

COMPOUND DOCUMENT PROCESSING. A Rapport document can contain text, graphics, image, spreadsheet (and charts), and sound. The best part of Rapport's document processor is definitely its compound document editor, which lets the modules interact in a single document. In fact, sometimes the compound document capabilities make up for missing component functionality. For example, the text object (or word processor) doesn't allow for snaking columns (or columns of any sort, for that matter). The only way we could make columns was to embed the text object into columns of a spreadsheet object. Essentially, we let

the text object take advantage of the columnar functionality of the spreadsheet object. Clearly, this isn't the best way to create text columns. It works, but it's a hob-job approach.

Performs Most Basic Word Processing Functions

Text. Rapport's text editor lacks state-of-the-art word processing features. We're beginning to see word processors with powerful macros, compound documents, electronic publishing features, outlining, and customizable interfaces. Although Rapport is at the top of its game with compound documents and electronic publishing features, its functionality is generic. We were able to perform most basic word processing functions: multiple ruler, justification, multiple fonts, search and replace (non-case sensitive), on-screen attributes, headers and footers, mark and go to, undo, multiline spacing, spell-checking, cut and paste (live or ASCII if cutting and pasting different objects), and copy and move.

Pros: Rich Character Attributes

On the plus side, the X Window system gives the text editor some flexibility in terms of character attributes. By clicking on the Fonts menu, we were able create text in Times Roman, Courier, Helvetica, Symbol, AvantGuard, Bookman, (among others), and in point sizes ranging from 8 to 72. And the Format menu contained text properties such as bold, italic, underline, and strike through. Rapport doesn't have a redlining editing feature for Rapport text, but the strike-through mode is probably useful for editing purposes. Another editing option might be to have your reviewers identify their comments by inserting different-colored text. Rapport also has an outlining feature that helps users structure their documents with collapsible sections.

Cons: Missing Common Features

We've already mentioned the awkward columnar functionality. Also lacking in Rapport text are footnoting, indexing, thesaurus, and table-of-contents functions. These are fairly standard on most word processors, and Clarity mentioned that it plans to at least add footnotes and indexing in the next release. We'd also like Clarity to add a real redlining editing feature to Rapport text that identifies whose comments are whose and when they were made.

Graphics Are Well-Suited for Business Purposes

Rapport's drawing module—the art editor—is adequate for business purposes. It may not have the advanced functionality that high-end, production-quality graphics programs do, but for business documents, it's just fine.

Art provides a palette of basic drawing tools for creating and modifying objects that contain lines, boxes (and rounded-edge boxes), arcs, circles, ellipses, text, free-hand lines in color and patterns. We drew outlines in a range of widths, colors, and textures. Rapport also offers some useful shading options (line point-size options, grey fill, and drop shadow). Editing aids, such as rulers and grids, are also available.

Getting into the art editor was easy enough; we just clicked on the Art icon to create a drawing at the position of the cursor, and the art palette popped up. Once we were on the display surface—the canvas, so to speak—we were able to move, scale, rotate, enlarge, shrink, and include the drawing into other Rapport objects. We even popped an art object into a spreadsheet cell simply by cutting and pasting.

Image Module Capabilities

Rapport's image editor supports monochrome and color images and screen shots. With a scanner or online fax, you can include hard-copy black-and-white and color photographs, maps, or drawings into a Rapport document. Of course, any online drawing (like a CAD drawing) or bit map can also be included.

To include a raster image, we first clicked on the Image button to create a space in the document. From there, we moved to the File menu and selected the Import Raster command to specify the raster filename, and the system read the file into the document. We could have also

done a screen capture if the image was already displayed on screen. Once imported, the picture could be rotated, cropped, flipped (or reflected), and sized.

Spreadsheet Excels in On-Screen Presentation

Rapport's spreadsheet leans more towards flexibility of on-screen presentation than towards functionality, although it performs most basic functions (statistical, mathematical, boolean, financial, date, and what-if). It reaches to 702 columns and 8,192 rows—not as much space as many full-featured spreadsheets, but adequate for less intensive number-crunching.

The strength of Rapport's spreadsheet, like that of its other modules, is the consistency of its interface and its integration with other Rapport elements. You can embed other objects in the cells of a spreadsheet just by positioning the cursor and selecting the appropriate object from the menu. The converter mechanism also makes it ideal for retrieving and incorporating other spreadsheet data into your document.

Navigational Asset: Familiar Manipulation Options

One spreadsheet feature worth pointing out is the alternative navigation options. We were able to use cursor and scrollbar, Tab and Enter keys, or arrow keys to move around the spreadsheet. In this way, Rapport can be consistent with your standard spreadsheet in terms of navigation. In other words, if you use Lotus and it's become second-nature to hit the Tab and Enter keys to move from column to column and row to row, you won't get bogged down in a different navigational interface when you occasionally use Rapport. We really like this feature. It says a lot about the product's adaptability.

Common Charting Functionality

Creating a chart was not much different than creating charts in other spreadsheet programs. We first selected the cells we wanted to include, and then chose the chart-type option from the menu. Chart types include pie charts, column and bar charts (and stacked bar charts), line and step-line charts, surface line and surface step-line charts. We used the art object tools for further manipulation. (See "Graphics" above.)

Like Using a Tape Recorder

Audio. Users can work with the Rapport Sound object, which is a digitized audio passage (usually voice), to make comments in a document. For presentation purposes, each Sound object is represented graphically by an icon of a loudspeaker.

Clarity kept the interface simple with standard recording controls such as record, playback, and volume. We felt as if we were using a tape recorder. To create a Sound element, we merely invoked it from the menu and spoke into the microphone. But we could also have imported prerecorded audio files, like music, lectures, or foreign language instruction, into the document.

All Object Modules Available for Slide Creation

SLIDE MANAGEMENT. All the Rapport object modules are available to its slide manager—even Sound. (An online slide presentation might really benefit from audio elements.) You can create both overhead transparencies and 35mm slides.

Primary Tools: Templates and Master Slides

Templates and master slides are the main tools for creating and modifying transparencies and slides. Templates give guidelines that specify the form, style, and position of elements such as titles and body text. Templates are modifiable: Once you apply a template to a slide, you have complete control over its elements, so your slides can vary as much as you choose. Master slides are more permanent. Like templates, these provide a guideline to design slides, but, unlike templates, master slides are unchangeable. They're used for consistency. Here at the Office Computing Group, for example, we like to preserve a specific look and feel for our analysts' slide presentations. Thus, every 35mm slide set in our library sports the same format, layout, logo, and colors. Master slides guarantee that kind of uniformity.

Converters Offer Wide Range of Input Sources

A slide sorter with thumbnail representations of your slide set is available for rearranging and repositioning slides. Once the collection of slides is created and arranged in the proper sequence, the slide set is saved as a single file.

The slide manager gives you plenty of drawing and text capabilities—all of the Rapport graphics and text editor functionality. And with the converters, you can incorporate drawings and text from lots of different word processing and graphics packages—even CAD drawings. This is an important feature for a slide manager. You usually can't gather all the visual and textual information you need for a slide presentation into a single source.

Enhanced by Compound Documents

Rapport's graphics flexibility is enhanced by its compound document architecture. It's easy to update data because it's all live. Say one of your slides features a business chart generated by a Rapport spreadsheet. Every time data in that spreadsheet source file changes, the business chart will change, too. Thus, your slide information is always current.

Output Devices

Rapport supports PostScript both for black-and-white and color printing and for 35mm film-recording. You can also output to Rapport's fax server (at this point, Rapport only supports its own fax server) or even E-mail your slide set as document enclosures to your print shop for professional production. You can also present your slides directly on your workstation or X terminal.

Compound Document Structure Maintained through Mail

MAIL. You can send even the most complex Rapport documents electronically. Rapport's compound document architecture is maintained through transmitted documents as long as you're sending them to another Rapport user. Otherwise, the system converts the document data into predefined application formats specified by your recipient (see "Address Books" below). Rapport's mail is not yet X.400 compliant, but the next release will include X.400 services in the mail client.

Transparent Conversion

What's nice about using Rapport mail is its seamless integration with the Rapport modules. When sending a message, you don't need to go through the hassle of closing your document, exiting the document processor, opening the mail application, and then importing the document as an attachment to a mail message. Instead, the document becomes your mail message. To send it, you invoke the Sendmail button (it looks like an envelope being put in a mail slot), fill in the header information, and away the document goes. To compose a new message, you open a new document, fill it in, and hit Sendmail in the same fashion.

Profiles Determine Conversion Preferences

Address Books. Rapport Mail is built on top of Unix Sendmail, but a number of third-party gateways are available to reach users of other systems. An X.500 directory service lets users create individual profiles indicating their gateway and application preferences, so that popular software packages may be substituted for individual Rapport applications. The mail system matches the username against the directory and then automatically converts documents to the specified formats. This is done behind the screens; users don't have to know which applications their recipients use.

Clarity refers to this directory as an Address Book, and you can keep your own private Address Book or use the public version. Rapport provides a browser for scanning through entries, searching for a specific entry, selecting entries for more profile detail, adding entries to distribution lists, and adding and deleting entries. User profiles include more than just E-mail and application preferences; the Address Book doubles as an online rolodex, containing the following information:

- Name
- Street address
- Telephone number
- Fax number (mail can be sent directly to the fax number)
- Title
- Company name
- Password
- E-mail address
- Text editor
- Table editor
- Raster editor
- Graphics editor
- Mail gateway
- Sound editor

Employs a Dialog Box Interface

Creating an Address Book entry is as easy as filling out these fields from a dialog box, and any field may be omitted or left blank for privacy. For creating a public entry, you must be in Administrative mode, but the dialog box interface is the same. Once the profile is set up, you only need to provide a username to send a piece of mail; Rapport takes care of the addressing and file conversion. (See Illustration 7.)

Rapport Address Book Browser and Profiles

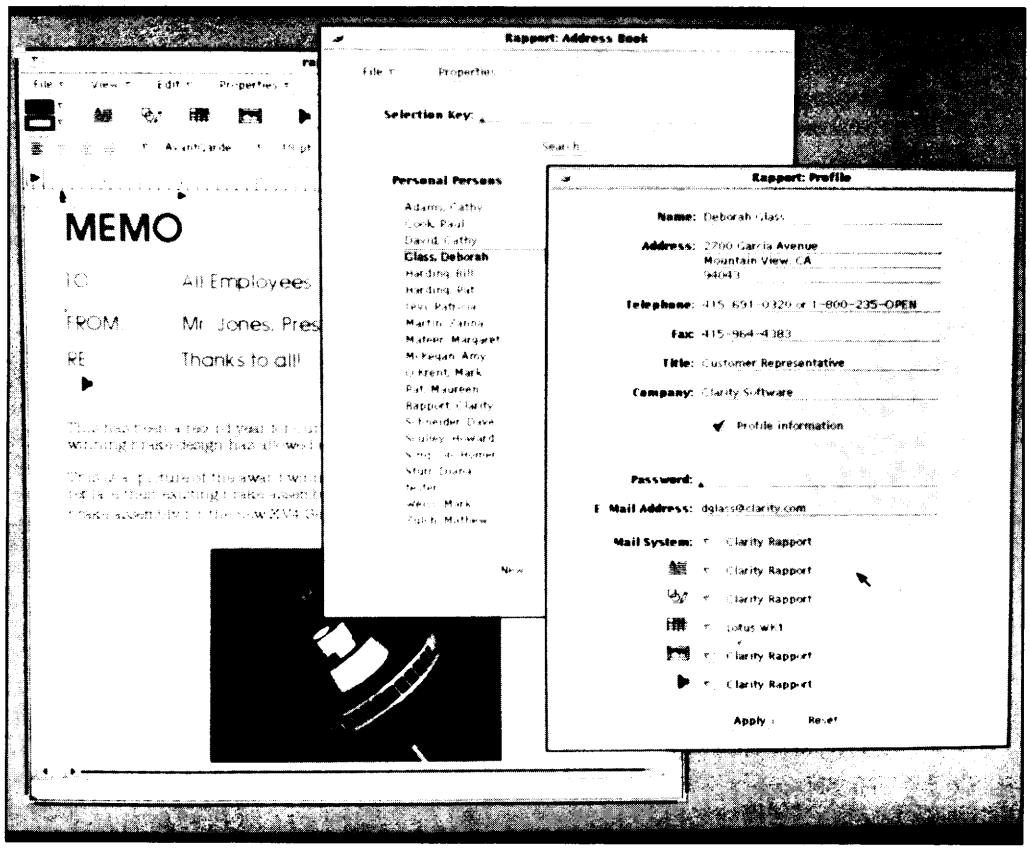

Illustration 7.

The Address Book is always accessible; you click on the button on the Rapport mail control panel. But it would be nice if the system also provided some sort of automatic look-up facility for misspelled names.

Conversion Strips Compound Document Functionality

Converter Limitations. Rapport documents sent electronically lose none of their characteristics, not even voice—as long as you're a Rapport user, that is. A converted compound document won't arrive as a compound document, but as a series of file attachments. If your recipient isn't in the Address Book or if he uses an application that the Clarity converters don't support, the document gets sent as plain ASCII text, and none of the other media elements is converted. Instead, Rapport leaves a text message letting your recipient know what he missed (e.g., "There was a spreadsheet here.") And once the document is stripped to text, Rapport will only recognize it as a text document. For instance, if you're sending a document round-trip to a text-only user, it won't come back to you with its compound document elements.

Futures

Publishing APIs

Clarity has planned a number of noteworthy directions for Rapport. The most important is that the company will be publishing its APIs later this year so that third-party developers can adapt their programs for tighter integration in the Rapport environment, enabling them to participate as compound document modules. Clarity was hesitant about giving us details about exactly what its APIs will provide and the specific tools they will offer. Generally, Clarity anticipates providing class libraries and browsers, object management tools, and compilers. The company also hinted at recruiting a partner with more expertise in object development environments to help it build these tools, which is a logical decision, considering the small size of Clarity. (ParcPlace, the group that brought about the SmallTalk language and development tools, was mentioned a few times. But whether or not ParcPlace will indeed partner with Clarity and to what extent it would be involved, we have no idea.)

Additional Plans

Releasing Rapport APIs will be a big undertaking for Clarity. Its other product plans appear to be less ambitious. Clarity plans to integrate Rapport with some of the more popular databases. This is wise, since users always need a front end to corporate data, and we were disappointed with its lack of database support. Again, however, Clarity gave no indication of precisely which databases it plans to support (although, from our standpoint, Oracle and Informix seem likely, given their dominance in the Unix marketplace). Clarity also intends to port Rapport to more platforms—and not just more Unix platforms. The company suggested that it is looking into DOS and OS/2 as potential Rapport platforms. The product already runs on the more popular Unix workstation lines: X terminals, Sun SPARCstations, DECstations 3000 and 5000, HP 9000 series workstations, Silicon Graphics Iris workstations, and IBM RS/6000 series workstations. Lastly, Clarity plans to build additional object modules.

Conclusion

Rapport: Application and Integrator

Rapport is as much pipes and plumbing for integration and communication as it is an application. Its converters provide a means of incorporating editable data from various standalone applications into a single document for compound document processing and distribution. You really don't even have to use Rapport modules to take advantage of its automatic mail conversion. The system will automatically translate a mailed document in your preferred application format regardless of the application with which it was composed. We don't know when we've bumped into an application that bends over backwards to interplay with other applications the way this one does. And the transparency of the interplay is especially attractive. Applications that use extension languages for integration don't have the same luxury. Although extension languages let users design scripts to bring in specific application data, they are also faced with the often daunting task of dealing with a programming language. With Rapport, however, all

Conclusion

the file conversion takes place behind the scenes. The user only has to set up a user profile—no programming involved.

Needs Clear Market Position

But Rapport also raises some concerns. The first is market perception. Initially, Rapport looks too much like an office system product, and potential users might not see past its less-than-optimal components. Clarity must be very clear in positioning the product for what it is: a mixed-application compound document processor and communication package. Rapport is very good at what it does, but even we had trouble figuring out its role. And if we had trouble, potential customers probably would, too.

Increasing the Converter Set

Another concern is the converters themselves. Clarity converters currently support 29 third-party applications. This is a good start, and Clarity has zeroed in on many important commercial applications, such as WordPerfect, Lotus, Excel, WordStar, MacWrite, MS Word, Interleaf, and Frame.

But the company can hardly rest on its laurels with 29. Nor does it intend to. The company has published the Rapport file format, so software developers can write Rapport converters to their applications. But we doubt that Clarity will be particularly successful convincing many ISVs to do that. Meanwhile, the company intends to add to its portfolio of converters.

The Jury Is Still Out

Assuming that it will continue offering additional converters and that its marketers will succeed in getting across Rapport's integration capabilities, the product should do well. Again, the timing of the product could not be better. Users who are fed up waiting for their systems vendor to provide an integration solution may well turn to an application like Rapport. Converters may be an imperfect integration vehicle, but with so few options available, it's looking mighty attractive.

But it's still too early to tell how the industry will react to Rapport. Its compound document architecture, cross-application compound document processing and transmission, and mail filtering make Rapport an exciting technology. In our view, the burden of success now lies largely on marketing. Clarity has a useful, adaptable product. We just hope it doesn't get lost in a muddled market perception.

Applications as Integrators

Integration via Extension Languages

Rapport is just one of a few Unix applications that make room in their systems for third-party applications, and the products described below are among the few. Implementations vary. What these products have in common that Rapport doesn't share is a programmatic extension language. (Rapport relies strictly on a converter mechanism for application integration.)

We should point out that these applications share a common goal of integration, but not necessarily a market (although Slate and the Applix products surely do). Also, reviewing the details of these products as applications is beyond our scope here. We're most interested in their extension capabilities, so we'll focus specifically on those aspects.

Slate

New Language for Extending the Product

BBN (Cambridge, Massachusetts) recently announced an extension language for Slate, its compound document processor and E-mail system. The Slate Extension Language (SEL) lets administrators customize the system, streamline applications, and automate functions.

Slate

An administrator's or technical user's tool, it's made up of a library of scrollable functions with point-and-click access.

Works across Slate Modules and Host Applications

SEL is quite powerful. It works across applications—both Slate modules and applications running on the host. Using SEL, users can chain together applications, and can pull data from other applications and use it within Slate. For example, an SEL script can be written so that a user can open a Unix window, launch another application—a database, perhaps—perform a query, and pull the queried data into the Slate environment, ultimately linking it into a spreadsheet. BBN is planning to include a direct SQL query front end in SEL for Slate's next release later this year.

Not for End Users

Essentially, SEL is a programming language. End users would probably be more comfortable using Slate's enclosure facility, which sports a friendlier dialog box interface. Slate's concept of enclosures is not much different from Rapport's: They provide a means of including non-Slate application data into the environment for mailing purposes (i.e., for mailing documents to users who prefer different applications than Slate's). However, Slate doesn't automatically convert the data into its own format as does Rapport (although a programmer can attach a conversion routine with SEL to do that). By using an enclosure, you are, in effect, adding another module to the system. A Slate enclosure is really only a source file; it appears in your document as a little envelope icon. (You must invoke the icon to actually view the data.) But you can attach commands to the datafile—to print it, for instance, or to copy it to another directory. The common practice, though, is to attach the file's native editor to manipulate its data.

Complements Compound Documents and Conferencing

Both Slate's enclosures and extension language are important elements, especially combined with Slate's powerful communication facilities. Slate's niche thus far has been in the electronic transmission of compound documents for editing and conferencing, and, as such, it's a valuable workgroup computing tool. Thus, SEL-based automated tasks can be routed around to other Slate users. Likewise, you can send editable non-Slate enclosures to other users for editing and conferencing.

WingZ

Offers Hypertext-Like Language

The graphical spreadsheet from Informix includes a hypertext-like programming language called HyperScript to interact with other programs. HyperScript is a portable, fairly intuitive command- and event-driven programming language that has a DataLink option for accessing and manipulating database information. HyperScript provides a strong application development environment. With it, you can design and customize graphical interfaces, create applications using regular programming constructs (e.g., global and local variables, CASE statements, and loops), or build a form (or screen, or menu). WingZ also has a Learn mode, which records your actions as you navigate your way through point-and-click and dialog box options. Thus, you don't need to know HyperScript to use it. Very nice.

Groomed as Informix Front End

Informix is positioning WingZ as a front end to its database, and this gives it a different purpose than something like Rapport or Slate. The DataLink option lets you reach a remote Informix database from within the spreadsheet. In other words, HyperScript can be used to create a customized front end for entering and updating data across a network. You can either type statements into an interactive SQL editor or use menus and dialog boxes.

Applix

Sharing Data with External Applications

Applix (Westboro, Massachusetts) is counting on its Extension Language Facility (ELF) for third-party application integration with Alis, its original compound document office system product, and Asterix, its new offering of X-based "productivity tools" (i.e., word processor, graphics package, macros, and optional spreadsheet package). ELF has matured a lot since it was first introduced a few years ago as Alis's macro language. It was developed to let users customize the Alis environment, and, initially, it included some 20-odd commands. Now it has over 800 commands as well as a huge reference manual. Applix developers assert that it's a complete programming language, not just a macro language. Actually, keystroke-captured macros are just part of ELF's strength.

Aside from its customization and macro capabilities, ELF lets users share and manipulate data with external applications. These can be commercial applications like Lotus or WordPerfect, or even Unix files and procedures. For some time now, Alis users have been able to create ELF macros to open a Unix window, run an application, and pull resulting data into the Alis environment (as you can with Slate's SEL). But now ELF also lets users include files from foreign formats into Alis or Asterix documents—as long as the file can be converted (i.e., via a conversion routine) to a format that Alis and Asterix understand. ELF also allows you to embed keystroke-captured macros directly into a document. Your organization might, for instance, distribute a weekly document that contains a financial report. Here, a macro could be embedded into that document that automatically runs and updates financial figures. Thus, the document becomes intelligent.

Used Strictly within Applix Environments

Like Clarity, Applix is cautious about promoting ELF outside the context of its applications. Customers easily recognize the value of Applix's applications, and applications, not development environments, make markets. However, customers may well want to extend the system, and ELF is there when they need it.

Interleaf

Developer's Toolkit Is Robust but Complex

Interleaf's Developers Toolkit is a very thorough environment for customization and third-party integration. Some programming languages don't even have the breadth of this product. Of course, it's also complex to use; Interleaf doesn't provide an end-user interface to the environment. This is an object-oriented, Lisp programming interface—an interpreted language that lets developers program on the fly without compiling first. The toolkit comes with a set of objects modeled after the Interleaf application (e.g., text, graphics, documents, containers, etc.), which can be used as they are, modified, or redefined. Again, this is a full-fledged language, complete with a source code editor, compiler, interprocess communications, and debugging tools. Programmers can even create completely new applications that have nothing to do with the Interleaf application—other than the fact that they are created with the Interleaf toolkit.

Toolkit Uses

Within Interleaf, the toolkit has several uses:

- Modifying or creating a new user interface.
- Creating "active" Interleaf documents. In other words, the toolkit attaches scripts to objects within a document to give it intelligence (à la the document-embedded macros found in ELF). A document itself may include menus, perform statistical analyses, retrieve database query results, give online help, provide hyperlinks to additional information—you name it.

- Integrating third-party software. There are two ways to do this: Invoking another application as a subprogram and then pulling data into Interleaf; or using Interleaf's Interprocess Communication (IPC) facility, which allows bi-directional application communication over TCP/IP. In fact, you can use Interleaf simply for its IPC capability to connect other client processes.

Comments

Extension Languages versus Conversion Architecture

The advantage that products like these have over Rapport is that they are development environments. Not only do they allow you to completely customize the application at hand, they also let you create completely new procedures and use them in the application. Rapport doesn't have the kind of extensibility, task automation, and smart document capabilities we've mentioned here.

Instead, Clarity has focused particularly on integrating existing applications, not writing new ones. And third parties can add their applications to the Clarity converter packs, thus enabling the application to interact with Rapport in a tightly-integrated manner. Rapport will also be releasing its APIs later this year, and we anticipate that they will take on the same kinds of features you find in these extension languages. Perhaps more.

Clarity Software
2700 Garcia Avenue
Mountain View, California 94043
Phone: (415) 691-0320
Fax: (415) 964-3483

Next month's *Unix in the Office* will address
Uniface: Developing Database-Independent Applications.

For reprint information on articles appearing in this issue,
please contact Richard Allsbrook at (617) 742-5200, extension 116.

Open Systems: Analysis, Issues, & Opinions

•RISC•

ACE: A Race for the RISC Desktop It's amazing how much excitement can be generated over a group of vendors getting together. The latest group rallying around a promised standard is 21 vendors intent on making the MIPS architecture into the desktop RISC standard. The group is not a consortium like the ones the industry has come to know and love, on-going organizations with management and a budget. This group will look more like the EISA bus consortium. EISA came into being when key vendors banded together to produce a specification for a PC bus that would compete with IBM's MCA. Similarly, ACE (Advanced Computing Environment) has come together to establish two binary standards for the MIPS R/4000 architecture. Given the number and stature of some of the players involved, this consortium could have a significant implication for the future of RISC, Unix, and Microsoft's portable OS/2 on the desktop.

Desktop Winners When Sun first announced that it planned to push its SPARC architecture as a licensable desktop standard, the rest of the RISC vendors understood that this could be a threat. If Sun was successful, it would be able to become the RISC desktop standard and gain significant market share. Sun has been able to build up some momentum in attracting OEMs for its architecture but, thus far, has not succeeded in becoming the preeminent Unix commodity desktop. In fact, no vendor has yet achieved this distinction. However, now that the market is fractured between Microsoft with its MS Windows (16-bit, 32-bit, and portable OS/2 or New Technology—NT—kernel) thrust and IBM with its continued Presentation Manager and OS/2, it appears that there is some room for Unix to have an impact on the desktop.

Power Play The other component that has led to the new coalition is the shift in the power structure within the Unix industry. When Sun was the key competitor, the power structure seemed to hold. However, when it became apparent that IBM was gaining strength in the Unix market with sales of close to $1 billion in less than a year, the power in the industry suddenly began to shift. This coincided with a more aggressive push by Hewlett-Packard in the workstation market. The present power shift comes at the same time as a shift in the PC industry itself. Vendors like Compaq, Dell, and others can no longer assume that selling commodity PCs will keep them on a growth curve—especially with the dramatic price/performance curves demonstrated by RISC platforms. Digital Equipment, although already adopting RISC technology, found itself losing strength in Unix against Sun, IBM, and Hewlett-Packard. Against this backdrop, it isn't surprising that the vendors that have market potential but are outside of the inner circle would look for a way to win power and market share. Each vendor that is part of the ACE group has a different perspective and a different reason for joining.

MIPS. While MIPS Computer has come on strong over the past few years, it hasn't achieved the kind of widespread acceptance of an Intel 386 or SPARC. But with Digital Equipment Corporation as a key partner, MIPS had a lot of potential. It was therefore in the best interest of both MIPS and Digital to discover a way to make the MIPS chip a more strategic player. The timing seems right. MIPS has begun to release samples of its long-awaited R/4000 processor, which has shown impressive performance and sophistication in design. However, one problem haunting MIPS has been the lack of compatibility among its OEMs. Since each OEM implements its own operating system, there is no implicit benefit in being one of MIPS's OEMs. To propel itself into a more strategic and powerful role vis-à-vis Sun and IBM, it was logical for MIPS to try to get all of its hardware partners to run the same operating system and the same user interface and to support the same binary. After all, the dream of a shrink-wrapped marketplace where users can go to the local distributor and buy a Unix application off the shelf is very alluring.

DIGITAL EQUIPMENT. Of all of the players, Digital, in our opinion, has the most to gain from the emerging MIPS group. Digital has thus far been perceived as the outsider, doing things just a little bit differently from others. While other Unix vendors selected Big Endian byte ordering, Digital chose Little Endian, the PC standard. Now Digital gains credibility because the new group has standardized on the Little Endian byte ordering. Digital will also be responsible for implementing the Unix operating system, which will be based on the OSF/1 kernel and will conform to System V APIs. The system will also be binary compatible with Ultrix so that the group starts out with Digital's 2,200 Ultrix applications. This operating system then will become Digital's standard operating system across the range of its DECsys-

tems, ranging from the desktop to multiprocessor implementation. Digital will also be charged with helping to implement other OSF technology, such as DCE and, eventually, DME on top of the OSF/1-based product. Digital had hoped the group would converge on its TurboChannel bus architecture. However, instead, the group has adopted two: TurboChannel plus the EISA bus.

Switching at this time from a Berkeley-based operating system to OSF/1 will help Digital become more mainstream. It will have the same MIPS binary as some key industry players and, therefore, will have the opportunity to build a larger portfolio of applications. Most importantly, if this consortium is successful, Digital will have the opportunity to compete with Sun's SPARC. At the same time, we expect that Digital will also take advantage of Microsoft's NT kernel. That becomes straightforward, since NT will run on the same hardware components and with the same byte ordering as the OSF/1-based operating system.

SANTA CRUZ OPERATIONS (SCO). SCO has emerged as a key player in this new group. Like Microsoft, SCO's fortunes have been linked to the Intel platform because its version of System V.3 is tied to the Intel 386 architecture. However, SCO has pushed further into an enterprise distributed environment via Open Desktop, a desktop environment based on components including Motif, Unix System V.3, a data manager (primarily Ingres), and communications via TCP/IP and Lan Manager/Unix (LM/X). SCO was brought into Open Desktop via Digital Equipment, which wanted to have something on the market that would compete with OS/2EE. Digital took a low profile with Open Desktop at its introduction. Lately, it has increased its involvement with SCO, primarily on the distribution side, because SCO has the distribution channels on the low end that Digital lacks. SCO has a much better understanding of the pragmatic Unix user environment and is a good partner for Digital. SCO brings some key components to the arrangement. It will be a key supplier of the OSF/1-based operating system, it will contribute the knowledge of System V, and it will be the default desktop for the MIPS consortium.

MICROSOFT. Microsoft's fortunes have been tied to the Intel platform since the company's inception. While Intel has served the company well, Microsoft's lust for power and dominance demands that it not be relegated to a single-platform strategy. Even Microsoft's acquisition of 15 percent of SCO still left the company dependent on Intel, since SCO Unix is an Intel 386-based operating system. For more than a year, Microsoft has been talking about a portable version of OS/2. The company has expanded this concept with development of the New Technology (NT) kernel, a hardware-independent, object-oriented kernel that could host a number of other operating systems and processes. This technology will be aimed at the desktop and will become part of a 32-bit version of MS Windows. Microsoft has begun to prototype its NT kernel on the MIPS platform. For Microsoft, a binary standard for a second platform other than Intel is part of the grand strategy to have its operating system and its suite of applications become a de facto standard. While the conceptual design of the NT kernel is very alluring, Microsoft must solve some key issues, such as its lack of standard networking components. For example, there is no promise of integrating DCE into its future product offering.

COMPAQ SYSTEMS. Like Digital, Compaq is a company in transition. While Compaq has been successful in becoming an important PC player, to grow, the company will need a more robust platform. Compaq's base is being rapidly eroded by PC-compatible clone-makers, who have improved their products enough to make it clear that Compaq needs new future directions. Compaq has made some inroads outside of this base via its arrangements with SCO. It is OEMing SCO/Corollary multiprocessing implementation of Intel 386/486 platforms. But, like SCO and Microsoft, the MIPS agreement allows Compaq to reach beyond the restrictions of the Intel platform. Also, if the consortium succeeds in providing a price/performance platform with a large portfolio of applications, Compaq has the opportunity to become a major RISC desktop supplier—especially if it is able to reinvent the PC shrink-wrapped software model through the MIPS arrangement. Compaq has also announced a strategic relationship with another participant in the consortium—Silicon Graphics. Silicon Graphics has the expertise in the workstation world that Compaq will need to become a serious player.

Conclusion

There is a lot of work to be done before all the pieces of the complex arrangement come to fruition. First, the operating systems (the NT and the Unix versions) must be completed. Networking components must also be completed, and pricing and bundling options must be decided upon. Both the Microsoft and the Unix contingents are convinced that they can get developer's kits to the market before the end of 1992. Even so, it will be at least a year or two more before these kits become working products. In addition, MIPS is only just beginning to produce sample batches of the R/4000 processor. The product will not be widely available for some time. While the group is working to make these products viable, don't expect Sun, HP, or IBM to stand still. It will not be an easy road for this latest industry power play.

If the ACE group is able to achieve its goal of producing two binaries for the MIPS chip and can convince enough vendors to adopt this technology, it could make an important impact on the potential for RISC on the desktop. The notion of a single desktop operating system and binary has been attractive ever since the DOS PC became the desktop standard in the '80s. Since then, vendors have looked for a formula to replicate that success. Again, the proof will be in the number

of ISVs these vendors can lure to their platform. The combined strengths of these players could make significant strides toward making commercial Unix a reality. But there are major hurdles to overcome, and this is not a short-term answer. —*J. Hurwitz*

•USL•

Unix Systems Laboratories Cuts Loose

AT&T's long-promised decoupling of its Unix operating System Laboratories (USL) from the computer and telecommunications business took another critical step when AT&T officially announced USL's first round of investors. While several had long been rumored, others were a surprise. Thus far, USL has sold approxmiately 25 percent of its holding. AT&T retains 60 percent for now, but, long term, this share will eventually fall to 40 percent. USL employees hold 10 percent of the company. The new investors include Amdahl, Sun Microsystems, Motorola, Novell, NEC, Fujitsu, Toshiba, Oki, ICL, Olivetti, and the Taiwanese-based Institute for Information Industries. Two other investors are anticipated, although, because of complex negociations, USL isn't able to announce them yet. The two most interesting aspects of the agreement are the buy-in by Novell and the role of Fujitsu (which owns part of Amdahl and all of ICL). Novell, because of its success with Portable NetWare, seems to want to ensure its recognition as a major Unix player. Fujitsu, likewise, wants to gain a more powerful position in the market.

Now that USL is being decoupled from AT&T, the mother of all bureaucracies, it has the opportunity to become important in its own right. But that importance is not just tied to its work on System V.4 (although this will remain an important focus). The fact that USL also controls AT&T's object-oriented language, C++, means that the company could emerge as a player in the object-oriented world. One area that USL is eager to distinguish itself in is networked object-based desktop environments. No specific plans have yet been announced. Other technologies that USL will focus on include the Tuxedo OLTP Monitor, OSI networking products, and tools for these areas. In fact, USL is trying to position itself as a software infrastructure company focusing on system level software (i.e., operating systems), tools, and services—in other words, everything but applications software. USL will benefit from a continued relationship with AT&T's Bell Laboratories. USL will have the option through a contractual relationship to "productize" research coming out of the labs.

How will this change affect the software arm of AT&T? First, USL will have 500 individuals on its payroll and another 100 on contract. Over time, the role of AT&T will diminish from its current 60 percent ownership to about 40 percent ownership. We expect that AT&T will go out of its way not to appear to be too controlling an influence in the young company.

CONCLUSION. Ironically, had AT&T spun off USL several years earlier, this announcement might have been different. We probably would have seen Hewlett-Packard, IBM, and Digital picking up a piece of USL. But with the growing power of the Open Software Foundation, major systems vendors have less need for USL than they might have had a few years ago. What will be the impact of USL as an independent company? Initially, we do not expect any noticeable change. In the long run, if the company is able to change into a proactive and entrepreneurial company, it could have a positive impact on the Unix industry. However, there are many challenges. First, USL will have to find a way both to meet the requirements of its assorted investors and still focus on the long-term need to develop its own identity and focus. Second, it will have to find a new way to work with Unix International. We expect that, as the role of USL changes, so will the role of Unix International. We predict that Unix International will strive to become much more independent of USL than it has in the past. This is not necessarily a bad thing for USL. Unix International could become the type of "super user group" that helps keep USL on its toes. In the long run, however, it will be up to USL to prove that, as an independent company, it has the vision and the drive to become a profitable and meaningful participant in open systems. —*J. Hurwitz*

•ODAC•

Coalition to Drive ODA Use

IBM, ICL, Unisys, Siemens/Nixdorf, Bull, and Digital Equipment have created the Office Document Architecture Consortium (ODAC) with the goal of promoting the development and use of ODA software. Each of the founding members have proposed products that will be developed under the aegis of ODAC and then licensed through ODAC to the user community.

Current proposed contributions are as follows:

- Bull is supplying its existing ODA formatter, which will be based upon the forthcoming ODAC ODA toolkit.
- Siemens is providing diagnostics to analyze the ODA data streams.
- Unisys is providing a Page Description Language (PDL) to the ODA converter.

- ICL is providing a Document Application Profile (DAP)-level API to be built atop the ODA toolkit.
- IBM is providing project management and integration.
- Digital is providing the ODA toolkit.

The toolkit is a critical item. ICL also had an ODA toolkit as a result of its work with the ESPRIT project. However, Digital got the nod to provide the toolkit based on its experiences with Compound Document Architecture (CDA) and the development of its own ODA CDA gateway.

The ODA CDA gateway provides a high-fidelity, bidirectional interchange between ODA and CDA environments.

Having Digital provide the toolkit should boost user confidence in implementing CDA solutions now. In effect, buyers are getting a guarantee of compatibility between ODA and CDA.

The presence of the consortium can only help spur the acceptance of ODA. ODA, of course, needs to evolve technically a little more quickly than it has done in the past. From what we've seen, the major systems vendors are committed to making this happen. IBM and Digital, for example, are both quite active in prodding along the ODA process.

—M. Millikin

•OBJECT MANAGEMENT•

OMG Interoperability Standard Moves To Critical Phase

The attempt by the Object Management Group (OMG) to identify a standard framework for communicating objects has entered a make-or-break phase. The effort will succeed if the vendors who submitted technology proposals for the framework cooperate to create technology that meets all requirements. Individually, none of the seven Object Request Broker (ORB) technology submissions meets all requirements of all users.

The OMG's effort will fail if vendors turn the search for a standard framework into a fight for their own political gains. We're very disturbed by press reports that a fight is indeed brewing between the Sun and Hewlett-Packard partnership and a nascent Digital/Microsoft pairing. In this battle, there can be no winners.

What Is the ORB?

The OMG's framework is called the Object Request Broker (ORB). It performs three functions in a network of heterogeneous systems:

FUNCTION #1: FIND OBJECTS. The ORB must manage object names. When an object—an application or a service—issues a request to another object for a particular operation or service, the ORB must be able to find the object needed to provide the requested service.

FUNCTION #2: CARRY MESSAGES BETWEEN OBJECTS. The ORB must coordinate the packaging of parameters, arguments, and other objects within the messages that pass between objects.

FUNCTION #3: BIND METHODS TO OBJECTS. The ORB must provide a means to bind messages to objects. To support interactions in heterogeneous systems, the ORB must be able to mediate different binding methods. Individual RPCs, for example, bind to procedures in different, incompatible ways.

In addition, an object request may not specify which remote object supports the methods (operations) needed to satisfy the request. In this case, the ORB must be able to determine which object can perform the requested operation and, possibly, create new object classes on the fly.

There Are Two Major Uses of ORB Technology

Two types of users will put the ORB to work: systems integrators and object-oriented software developers.

SYSTEMS INTEGRATION. The systems-integration users will use ORB technology primarily to integrate old, standalone applications. The ORB allows one application to ask another to do work on its behalf. The ORB (indeed, object management technology in general) provides a basis for independently-developed applications to communicate and cooperate with one another across platforms and networks.

For example, an ORB could be used by an analysis application to contact a stock quote service and request stock price information sorted according to specific criteria. The analysis application does this by sending a request to the stock quote service. The request includes the criteria to be used in processing the request for data. In response, the stock quote service performs the requested operation and returns the results to the application.

Requirements of Systems-Integration ORB Users. The systems-integration users have two requirements of ORB technology.

First, the ORB must allow new services and applications to be added to a network without requiring a change of existing objects and applications. These users will not tolerate reworking, recompiling, or relinking existing software to accommodate new software.

Second, adding new applications and objects to an existing network must be straightforward and platform independent.

The submitters primarily interested in ORB as systems-integration technology have created separate interface definition languages and/or APIs to describe objects (applications and services) to the ORB.

In these submissions, object interactions take place in, essentially, a parallel framework. Objects are defined to the ORB; they are not self-defining. These solutions are typically not well-integrated with existing directory structures, security services, programming languages, and other facilities, but they are platform independent. Tighter integration is a future goal for the technology.

OBJECT-ORIENTED DEVELOPMENT TECHNOLOGY. The second use of the ORB is to locate and deploy locally created objects in a distributed environment. A C++ contingent within the OMG expects the ORB to extend its local programming environment across distributed systems.

Developers working with object-oriented programming languages work with the kinds of object interactions the systems integration community does, but at a much finer level of granularity. The classes these developers create pass messages to other classes, asking them to perform operations in their behalf.

For example, a customer service application might use a class called "payments" to register payments in a database. The class is a generic description of payments of all varieties, and it contains pointers to specific payment methods, or operations. The application might allow customers to be billed later, for example, or to pay by American Express credit card but not MasterCard. Depending on the type of request, the right payment operation must be invoked by the single generic payment class. The generic payment class does this by mapping the type of the request to the proper payment subclass. The subclass then executes the proper response to the request. Also, the developer can add new payment operations and be sure the system won't break.

Object-oriented languages support applications such as the above in a local environment. C++ features supporting inheritance and "overloading" allow developers to send messages to classes asking for an operation without knowing the details of how the right operation gets invoked. Object-oriented developers want the ORB to extend their reach to classes running on remote systems.

Requirements of Object-Oriented Developers

Developers of object-oriented software, as represented by the OMG's C++ contingent, have two requirements of an ORB.

First, they want to use the ORB from within their object-oriented language. A lot of the work in coding with an object-oriented programming language is defining the semantics of objects that perform particular operations. The members of the C++ contingent want an ORB that can use their C++ definitions to interact with remote objects.

Defining an object's semantics in an interface definition language—no matter how small and simple—is anathema to an object-oriented programmer. It is redundant work.

Second, the object-oriented programming contingent doesn't want to be constrained by the transport mechanism the ORB uses to carry messages. Most of the ORB submissions rely on standard RPCs, which require object-oriented semantics to be mapped to C structures or something similar. These structures define basic data types, parameters, and arguments, but not objects. The mapping process from objects to flat structures and back again exacts a performance toll that object-oriented developers consider unnecessary. In addition, object-oriented developers are concerned that, in the mapping, they'll lose important semantics, making it harder for them to interact with remote objects.

Seven Companies or Groups Submitted Technology in Response to OMG's RFT

HEWLETT-PACKARD/SUN. The HP/Sun submission to the OMG request for technology (RFT) is based on the common RPC interface definition language the two companies recently committed to create. The common language unifies the two companies' separate RPCs.

DIGITAL. Digital Equipment's submission is its Application Control Services (ACS) 2.0. ACS is a transport-independent location and binding service that can use multiple RPCs and other transport mechanisms. ACS is in use within Digital.

HYPERDESK. HyperDesk's submission is a transport-independent location and binding service from an unannounced product. HyperDesk's submission is very similar to Digital's.

NCR AND OBJECT DESIGN. NCR's submission is the location and binding services included in NCR's Cooperation office environment. NCR's ORB is transport independent. The service is implemented in a set of C++ classes, and it features the kind of programming interface the C++ contingent at the OMG wants. Object Design is supporting, but not adding to, NCR's submission.

GROUPE BULL. Groupe Bull's ORB is a Unix-based location and binding service with interfaces to a proprietary object-oriented language and the Eiffel programming language.

APM. APM submitted its distributed computing architecture, which has been implemented on a handful of major systems. The APM distributed environment is based on an extended RPC and a robust set of services to support very large networks.

DSET. DSET's submission is an extended RPC with location and binding services.

The Selection Process Must Resolve a Welter of Issues

The OMG's technology selection process is being run by OMG members. There is no technical staff sitting in the background evaluating technology and requirements. The OMG would like to select one of the seven submissions and allow the industry to get on with building solutions.

Unfortunately, OMG cannot choose any one submission. None of the seven ORB submissions meets all requirements, now or for tomorrow.

NCR AND BULL VS. THE SYSTEMS INTEGRATION CROWD. The pure object-oriented approaches of NCR and Bull give object-oriented developers what they want. But this approach doesn't give the companies interested in systems integration the simple systems integration service they want.

The object-oriented approach virtually requires applications to be registered with the ORB via C++. Digital, as an example, doesn't have a C++ compiler yet. So there's a product line problem. In addition, Digital is almost religiously opposed to imposing any one programming language on the user of the ORB. A simple interface definition language is as far as it wants to go.

DIGITAL, SUN/HP, ANSA, AND HYPERDESK VS. THE OOPS CROWD. On the other hand, the RPC-based ORBs, such as HP/Sun's submission, don't give the object-oriented programmers (OOPs) the transparent interface to distributed objects they want.

PLATFORM PROBLEMS. The HP/Sun ORB and the Bull ORB have been implemented only on Unix platforms.

INTEROPERABILITY PROBLEMS. The Sun/HP ORB fails to mask the different naming conventions of its underlying RPCs. Thus, the two implementations of the HP/Sun ORB technology do not interoperate without some kind of mapping bridge.

INTERFACE VS. IMPLEMENTATION. It's difficult to see how to separate the NCR ORB's interface from its implementation as a set of C++ classes. It's not clear how NCR's interface will support additional languages, even object-oriented languages.

NO SECURITY IN API. None of the submissions include authentication in their APIs. This limits their usefulness in widespread distributed networks.

Needed: A Combination of ORB Technologies and a "Win-Win"

What's needed to meet all requirements is cooperation among the vendors. This cooperation can begin if all submitters agree that none of them is going to reap huge profits from ORB technology. The solutions that ORBs allow will generate new business for vendors, not the core technology.

We're heartened by our strong sense that negotiations to create a "cooperative" ORB are under way. This cooperative solution could comprise the following components:

LOCATION AND METHOD BINDING CORE FROM DIGITAL OR HYPERDESK. HyperDesk may lose out if it can't persuade Digital to back a combination of their two technologies. The two submissions are very similar in approach (single API location/binding service), but implemented differently. Digital would rather see HyperDesk go away.

STUB COMPILER FROM HP/SUN. The Common Definition Language (CDL) is the heart of HP/Sun's submission. It embeds a rich set of semantics in RPC stubs, but fails to address cross-RPC interoperability. For the class of applications requiring compile-time control over references, HP/Sun's CDL is attractive, possibly state-of-the-art.

C++ LANGUAGE BINDING FROM NCR. The really whizzy aspect of NCR's ORB is its implementation of a transparent C++ interface as a set of class libraries (the Cooperation Frameworks). This is extremely appealing to the small but vocal C++ contingent in the OMG.

EIFFEL AND OTHER LANGUAGE BINDINGS FROM GROUPE BULL. Bull is the only submitter to tackle multilanguage binding.

LARGE-SCALE NETWORK NAMING SERVICES AND OTHER EXTENSIONS FROM HP/SUN AND APM. HP's Manager of Managers is the key here—it addresses the problem of locating objects across huge networks. APM's architecture assumes global networks, and has some interesting features (The Trader) that could be add-ons.

Cooperation as a Result of Pressure to Meet the Deadline

The ORB process is on a tight schedule. The Task Force selection is due in early June. OMG president Chris Stone and technical vice president Richard Soley are barred by antitrust laws from actively participating in negotiations. However, Chris Stone *did* help the submitters find the negotiating table by telling them that the OMG isn't forced to select only one submission. The notion that the ORB Task Force

had no choice but to choose one winner (leaving six losers) had hung like a pall over the selection process until then.

Thus, although the OMG process doesn't directly aid the negotiators, the pressure it exerts is important. Without the external pressure of a deadline, the ORB negotiators would have little stimulus to reach an agreement. There's no common foe driving them (as Microsoft appears to be driving Patriot Partners and as Sun SPARC drove the 88Open Consortium). Customers aren't really calling yet for an ORB (open systems still means a common operating system to many users).

Solving the Problem vs. Winning the Beauty Contest At an organizational meeting that followed the three-day ORB presentation conference, participants expressed worry that the ORB selection would be a "beauty contest" rather than a serious effort to identify a viable technical solution.

We believe selection of any one submission will be a big mistake. A cooperative arrangement of some sort is the only way for the OMG ORB to satisfy all constituencies *and* preserve the submitters' enthusiasm for implementing the eventual standard. —*J. Rymer*

And Then There Were Four

Since the above article was written, the OMG has narrowed its list of finalists.

Seven companies and/or partnerships submitted technology for consideration as the Object Management Group's Object Request Broker (ORB). In mid-April, the Object Management Group's ORB Task Force took its first step toward selecting a standard approach to object management by eliminating two of the submissions from consideration.

The task force cut Groupe Bull and DSET from its "short list" of finalists to supply the ORB. And, more recently, APM withdrew its submission. The list of finalists for consideration now includes:

- Hewlett-Packard/Sun
- HyperDesk
- Digital Equipment Corporation
- NCR/Object Design

The ORB Task Force must now sift through the four finalists, judging them against a list of 11 criteria. A final selection is due to be announced in early June. —*J. Rymer*

•OLTP•

Stratus Launches RISC, Promises Distributed Support Stratus Computer Incorporated has begun implementing its open, distributed online transaction processing (OLTP) strategy in earnest. Stratus announced the first of a new generation of RISC-based computers in March. At the same time, the company outlined the software announcements it will make during the next year to support distributed OLTP.

Unix-based OLTP has become a key battleground for Stratus in its competition with larger rival Tandem Computers. Both companies announced Unix product lines during 1990, but Stratus lagged behind Tandem in bringing RISC processors to market. Stratus now has closed the RISC gap in its competition with Tandem for new applications in open OLTP.

Stratus's description of its software strategy indicates that it will not compete only on the price and performance of its hardware and the openness of its operating systems software. Stratus plans an aggressive move into distributed OLTP during the next year. In addition, the company plans to offer applications development platforms to support customers as they create new applications.

NEW HARDWARE AND SOFTWARE PRODUCTS. The new XA/R Model 20, which is based on the Intel i860 RISC chip, was the highlight of Stratus's announcement, but important software was also announced.

The XA/R RISC Line. The XA/R Model 20 is the first of a full line of RISC-based computers that Stratus plans to develop and sell during the next two years. The first model is a midrange computer, rated at 40 MIPS, that Stratus is positioning as a departmental or branch-office solution. It will be available in the third quarter of 1991.

The XA/R conforms to Stratus's long-standing systems architecture—it just plugs into it a more cost-effective chip. The XA/R line uses the same hardware-based, fault-tolerant design that Stratus's older computers use. Two identical processor complexes run in parallel, each checking the other to ensure accuracy. If a failure or problem occurs, one of the two processing complexes can be serviced or even replaced, while the other continues processing.

Stratus chose the Intel i860 because the chip integrates integer core processing, floating point processing, memory management, instruction and data caches, and bus and cache control into a single chip with a 64-bit design. These characteristics are not unique to Intel among competing RISC architectures, but some of the major RISC architectures

(notably, Sun's SPARC) integrate less functionality on a single chip.

Stratus's choice of the i860 was a surprise. Few of the major systems vendors have selected Intel's RISC, choosing, instead, the RISC chips from Sun, MIPS Computer Systems, or Motorola. The big selling point for Stratus on i860 was Intel's future plans for the chip—which Stratus is not revealing. Stratus believes Intel's planned enhancements for i860 will give it a powerhouse for years to come.

Provided that the i860 performs reliably, Stratus's use of Intel's RISC chip is not an issue for users. Stratus hardware is very specialized. XA/R systems optimize the i860 for continuously operating applications by supplementing the i860's data cache and off-loading communications and disk I/O onto subsystems. Vendors of general purpose systems may need binary compatibility with major chip families to compete in open systems markets, but we don't believe Stratus does.

Stratus now has two hardware lines, the XA/R and the more expansive XA2000. The XA2000 is based on the Motorola 68030. Stratus also announced two new XA2000 systems, the Models 270 and 280, which are high-end extensions of the line.

XA/R: The Software Story. The XA/R series runs both a new release of Stratus's FTX Unix variant and its VOS proprietary operating system. This is unusual. We don't know of another RISC outside of Hewlett-Packard's HPPA that runs a proprietary operating system.

Stratus has implemented support on the XA/R only for Release 2 of FTX, as opposed to Release 1. Release 2, which is compliant with Unix System V Release 4, will be available in the third quarter of 1991. Thus, applications written to VOS or FTX Release 1 on Stratus's XA2000 line can be run after a recompile on the new XA/R system. The reverse is not true for applications written to XA/R under FTX. They will not run as easily on XA2000s under FTX Release 1—at least until Stratus provides FTX Release 2 on its other hardware line.

SOFTWARE STRATEGY FOR THE 1990s. Stratus's software strategy announcement clarified its migration toward support of open and distributed OLTP with its two operating system platforms. Stratus intends to provide the following software components for *both* of its operating systems platforms.

Open, Distributed Transaction Monitor. Stratus has previously announced its intention to offer transaction monitoring software based on the Distributed Computing Environment (DCE) from the Open Software Foundation (OSF) and extensions from Transarc Corporation.

Transarc's technology provides, among other things, a transaction monitor that manages the launching and completion of transactions across distributed networks. Without such software, distributed transaction processing is impossible. Because Transarc's technology is based on the DCE, and because DCE is open and available to many vendors, it can be considered a basis for open systems.

High-Performance File System. Many low-end and midrange OLTP applications are being built using relational DBMSs (RDBMSs). However, high-rate, high-volume applications still rely on files. RDBMSs aren't fast enough.

Stratus has always specialized in these applications. To apply its Unix and distributed technologies to this class of applications, the company requires a fast file system. We suspect that the high-performance file system that Stratus referred to is the Structured File System from Transarc Corporation, although Stratus didn't reveal details.

Communications Development Platforms. Stratus announced its intention to develop three separate platforms that will help users develop distributed applications. The first is "Streams," which the company described as a general platform for developing communications applications.

The second is Network Express, which is both an applications development environment for communications-intensive applications on LANs and WANs, and a run-time environment for the resulting applications.

The third is an OSI Server to support development of applications using OSI protocols.

These platforms signal to us that a major thrust for Stratus's future product line will be applications-development software and support. Stratus already markets the Stratus Intelligent Network Applications Platform (SINAP), an applications platform for the telecommunications industry. The platform runs only on FTX, complying with the telecommunications industry's requirement for Unix-based systems.

CONCLUSIONS. Stratus's March announcements indicate that its evolution as a supplier of open and distributed OLTP systems is progressing. We are watching Stratus reengineer itself as a company. Stratus has completed two key steps in this process by creating a Unix-based operating environment for its fault-tolerant systems and by introducing the more cost-effective processors offered by RISC technology to its hardware product line.

We're impressed by the continuity that Stratus has been able to maintain in its product lines in making these two moves. Stratus's dual operating system strategy makes the choice of hardware platform and operating system a small issue for its customers. By comparison, Tandem's Unix product line is

based on a different architecture from its proprietary Non-Stop computers.

Still, Stratus has a long way to go to complete its transformation. It will be hard enough for Stratus to create the infrastructure to support open and distributed OLTP applications. Having built the foundation, Stratus must then make it usable with applications-development and -management tools. —*J. Rymer*

STATEMENT OF OWNERSHIP, MANAGEMENT AND CIRCULATION. **1A. Title of Publication:** Patricia Seybold's UNIX in the Office. **1B. Publication no.:** 08873054. 2. **Date of filing:** October 1, 1990. 3. **Frequency of issue:** Monthly. **3A. No. of issues published annually:** 12. **3B. Annual subscription price:** $495.00. 4. **Complete mailing address of known office of publication:** 148 State St., 7th Floor, Boston, MA 02109. 5. **Complete mailing address of the headquarters of general business offices of the publisher:** 148 State St., 7th Floor, Boston, MA 02109. 6. **Publisher:** Patricia B. Seybold,148 State St., 7th Floor, Boston, MA 02109. **Editor:** Judith S. Hurwitz, 148 State St., 7th Floor, Boston MA 02109. **ManagingEditor:** Douglas Freeman, 148 State St., 7th Floor, Boston MA 02109. 7.**Owner:** Patricia B. Seybold, 148 State Street, 7th Floor, Boston, MA 02109. 8. **Known bondholder, mortgagees, and other security holders owning or holding 1 percent or more of total bonds, mortgages, or securities:** None.9. **For completion by nonprofit organizations authorized to mail at special rates:** Not applicable.**10. Extent and Nature of Circulation:**A. Total no. copies printed: Average 12 Mos. 1500; Single Issue 1700. B. Paid Circulation 1. Sales through dealers and carriers, street vendors, and counter sales: Average 12 Mos. None; Single Issue None 2. Mail Subscriptions: Average 12 Mos. 850; Single Issue 1476. C. Total Paid and/or Requested Circulation: Average 12 Mos. 850; Single Issue 1476. D. Free distribution by mail, carrier, or other means; samples, complimentary, and other free copies: Average 12 Mos. 50; Single Issue 34. E. Total Distribution: Average 12 Mos. 1900; Single Issue 1510. F. Copies not Distributed 1. Office use, left over, unaccounted, spoiled after printing: Average 12 Mos. 600; Single Issue 190 2. Returns from news agents: Average 12 Mos. None; Single Issue: None. G. Total: Average 12 Mos. 1500; Single Issue 1700. I certify that the statements made by me above are correct and complete. (signed) Patricia B. Seybold, Publisher

Printed on recycled paper.

