Patricia Seybold's

UNIX IN THE OFFICE

VOL. 3, NO. 5
ISSN: 0887-3054
MAY 1988

INSIDE

PRODUCTS • TRENDS • ISSUES • ANALYSIS

AI and Unix

Here and Now

By Mickey Williamson

THE INFERENCING AND pattern-matching techniques of artificial intelligence (AI) add significant value to office applications. Until recently, few office systems contained AI capabilities. Those that did most often demanded the rarefied atmosphere of a dedicated Lisp-processing machine.

Today, the push is on to tuck pieces of AI into all kinds of applications, from order entry to database and general-ledger systems. Decision analysis and support systems are helping managers to solve common business problems. Companies that use AI technology to enhance their *(continued on page 3)*

• E D I T O R I A L •

Goodbye, AT&T—Hello, Olivetti!

Can AT&T Survive the Loss of Cassoni?

By Judith S. Hurwitz

THE NEWS THAT Vittorio Cassoni was leaving his adopted home to return to his original homebase sent some tremors through the computer industry. It is especially troublesome for users and third-party developers who were just beginning to believe that AT&T was on a path that they could enthusiastically follow. There is no doubt that Cassoni is a visionary and that he helped AT&T translate its technology into the first steps of a cohesive, long-term strategy.

While it was a good start, his work was not done. What he brought to AT&T was leadership—not just leadership within the company itself, but to the outside world. For the first time, a person was in front of the monolith. This persona will be sorely missed.

FILLING BIG SHOES. The inevitable question remains: What next? Robert Kavner, AT&T's senior vice president and chief financial officer, takes the reins from Cassoni. How well he will continue the course that Cassoni has set remains to be seen. No matter what course he takes, Cassoni will be a difficult act to follow. Visionary leadership is rarely learned. While financial management is obviously important for AT&T's future, Kavner will have to hold a broader vision. He will have his work cut out for him. AT&T is not a company that is accustomed to responding to change quickly. We venture to guess that its slow movement helped Cassoni make his decision to leave.

REVOLUTION. This is a particularly bad time for Cassoni to leave AT&T. It coincides with a revolt on the part of the newly-formed Open Systems Foundation. These vendors include IBM, Digital, Hewlett-Packard, Apollo, Siemens, Nixdorf, Honeywell Bull, and Philips—an impressive list. As these vendors begin to view Unix as a strategic part of their business, they feel the need to have more control over the operating system. Vendors did not seem to mind when versions of Unix were written first for AT&T's own 3B systems. However, the tide began to turn once AT&T joined forces with Sun Microsystems. The notion that Sun and its SPARC chip would have a six-month lead on implementing new versions of the Unix was too painful. Also, the notion that Sun, in conjunction with AT&T, could control the future of Unix without the direct input of other vendors has increasingly become an issue.

Indeed, this is a critical juncture for both the Unix operating system and AT&T. Should this group of vendors go its own way, it could have serious implications for the future of Unix. AT&T needs to find a way to make everyone happy—which will be a herculean task. Kavner will begin his tenure with an initiation by fire. We wish him luck.

KEEPING AN EYE OUT. We will never know how Cassoni would have handled this crisis. Yet, it would be a mistake to dismiss him at this new juncture. Cassoni's move back to Olivetti tells us that this European giant is worth watching. As it boldly moves ahead, Olivetti will need a visionary. And, as we all know, visionaries are hard to come by. ◐

Patricia Seybold's Office Computing Group

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

Managing Editor
RONNI T. MARSHAK
Senior Editor
MICHAEL D. MILLIKIN
Assistant Editor
DAVID S. MARSHAK
Staff Writer
LAURE BROWN
Sales Director
RICHARD ALLSBROOK JR.

Associate Editor
JUDITH R. DAVIS
33 Summit Road, Lexington, Massachusetts 02173
Telephone: (617) 861-3926
Contributing Editor
GARY J. NUTT
3450 22nd Street, Boulder, Colorado 80302
Telephone: (303) 444-0341
Circulation Manager
DEBORAH A. HAY

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

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

•ARTIFICIAL INTELLIGENCE•

(continued from page 1) business systems are realizing productivity benefits that can leave their competitors in the dust.

Meanwhile, as vendors respond to the growing demand for enterprise-wide information systems, Unix is taking on new importance in the creation of intelligent applications.

What AI Needs, Unix Offers

AI technology is a valuable adjunct to traditional data processing (DP) techniques. It can deal with problems that are poorly defined, and find solutions even where information is missing or of uncertain accuracy.

In the conventional DP paradigm, problems are well defined. You can draw a flow chart to define the control structure at every possible branch point. Logic finds data and decides what to do next.

Many business problems are less well structured. Information may be imprecise, contradictory, or nonexistent. Reasoning and knowledge born of experience come into play. If software is to help, control must be implicit: Data finds the rules that apply and decides what to do next.

Yet AI development projects differ from ordinary application-building only in degree. AI applications involve searching through stacks of symbolic information for patterns that match the object under consideration. Thus, they demand greater quantities of memory and processing power.

AI project teams tend to be larger and more dispersed throughout the organization; hence, communication takes on added importance. So does project management. AI applications often go through 10 times as many iterations as conventional systems before they are finally released to the user community. Version control under such circumstances can be a major concern.

Expert systems, the most common manifestation of AI technology in office applications, often require complex graphics and windowing capabilities.

Most important, few organizations will pay the price required to build a system that incorporates artificial intelligence but does not have it available on every computer, regardless of its logo.

Unix has much to offer the AI development effort in each of these respects. As a platform for AI software development, a Unix workstation takes second place to the dedicated Lisp-processing computer when the scoring is based on memory management, windowing, and graphics. But you can field an AI development team running Unix boxes at a fraction of the cost of a network of Lisp workstations.

Workstations such as those from Sun and Apollo offer excellent windowing and graphics facilities. Texas Instruments' Explorer LX represents a multiprocessor marriage of Unix and Lisp, the foremost language for AI development.

Expert Systems: Finding a Suitable Subject

ESPECIALLY FOR THE first expert system project, choosing a suitable subject is of prime importance. The process begins with a list of current tasks and problems.

- **Repetitive work.** List jobs currently performed by several people at varying levels of skill.
- **Scarce expertise.** List problems currently solved by one person whose knowledge is much in demand.
- **Failure of traditional methods.** List problems that traditional, algorithmic programming methods have failed to solve.
- **Repeated judgment.** List decisions requiring human judgment, where the inputs change but the policies or rules applied do not.

Now cross off this list all problems that don't meet the following criteria:

- **Telephone consultation.** Given enough information, a person should be able to solve the problem in a single telephone call. Finding a solution should not require direct sensory observation or physical manipulation by a human being.
- **Time to solve the problem.** A person should be able to solve the problem in more than a few minutes and less than a few hours. If it is now being done in a minute or two, the task is too trivial for an expert system. If it takes more than a few hours, it's probably too complex for the current state of computer and expert system technology, and it is certainly too complex for a first attempt.
- **Available knowledge.** A body of knowledge must exist and be available. It can be either the know-how of a single individual enthusiastic about the project or the contents of a manual or training course. Algorithms don't work. If the problem is one that can be solved by a traditional algorithmic program, it should be solved that way. AI techniques are best reserved for situations where more familiar methods don't work. On average, an AI project will cost more, take longer, and involve more risk.

Finally, once a suitable topic is found, it should be refined to make it as specific as possible. Its scope can expand over time, but it is far easier to find a sponsor for an expert system that does one thing very well than for a system that does several things only passably.

An Expert System Vocabulary

Abduction. Also called "generate and test," a reasoning process that involves generating potential solutions to a problem and testing each solution's conformance to facts already known. (*See* Deduction, Induction)

Antecedent. Left-hand or IF side of an IF/THEN rule. (*See* Consequent)

AV pair. Expression linking an attribute with a specific value, e.g., age-45. (*See* OAV triplet)

Backward chaining. Reasoning method in which an expert system tries to satisfy a stated goal or prove a hypothesis by examining information about a specific problem. Most efficient where the number of possible outcomes is limited, as in a diagnostic problem.

Certainty factor (also confidence factor). Numerical value assigned to a statement or conclusion to indicate the degree of confidence that it is correct.

Consequent. Right-hand or THEN side of an IF/THEN rule. (*See* Antecedent)

Control mechanism. Method used by inference engine to determine how an expert system will reason to a conclusion, e.g., forward and backward chaining.

Deduction. Reasoning process that concludes something must be true because it is a specific instance of a general case known to be true. (*See* Abduction, Induction)

Domain. Area of expertise, subject of expert system.

Expert system. Software that applies problem-solving techniques to a specific domain. Correctly used to refer to system that embodies scarce human expertise. (*See* Knowledge-based system)

Expert system shell. Software tool that aids in developing expert systems, particularly one that can communicate with the rest of the computer environment. Sometimes termed incorrectly "expert system."

Forward chaining. Reasoning method in which an expert system collects information as it moves toward an answer. Most efficient where the number of possible solutions is too large to list, such as one that gives investment advice. (*See* Backward chaining)

Frame. Structure for representing knowledge in a knowledge base. Each frame contains knowledge about a single object. Frames can be related to each other in a hierarchy so that frames lower in the hierarchy inherit characteristics of those that are higher.

Heuristics. Educated guesses and rules of thumb an expert uses to solve a problem.

Hybrid system. Originally, an expert system that combined two or more ways of representing knowledge, such as rules and frames. Now also used to describe an expert system that runs external programs at a call from a rule proved true.

Induction. Reasoning process that derives rules from example problems and their solutions. (*See* Abduction, Deduction)

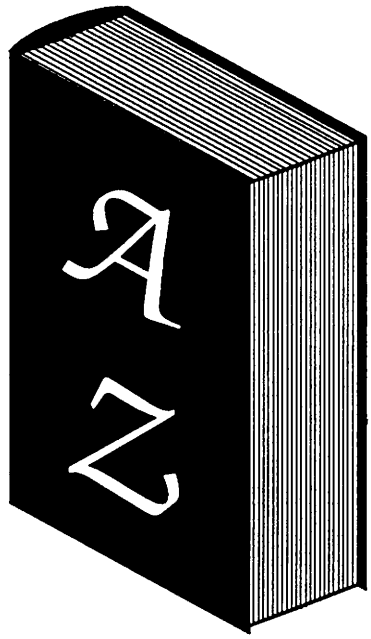
Inference engine. Part of an expert system that contains procedures for reaching a conclusion.

Knowledge base. Body of facts, rules, and heuristics that forms the basis of an expert system.

Knowledge-based system (also knowledge system). Expert system based on a body of written material, rather than the knowledge of a human expert, e.g., a policies and procedures manual.

Knowledge engineer. Person who designs and builds knowledge-based and expert systems. Title implies, but does not require, training in cognitive psychology and computer science. Some experts are their own best knowledge engineers.

OAV triplet. Expression linking an object with one of its attributes, and with that attribute's value in a specific instance, e.g., employee-age-45. (*See* AV pair)



But it is the 80386 processor that has brought down the entry barriers in the AI development field. For under $10,000, Unix on the 386 can do what MS-DOS cannot: ignore the 640K memory barrier and use all the real memory the machine can hold, plus enough virtual memory to make even jaded AI programmers speak in hushed tones.

Unix on the 386 meets AI development's multiuser and multitasking requirements with aplomb. Running DOS as a Unix process, Microsoft Windows supplies the graphics component, with standard Unix pipe mechanisms providing the necessary bridge between Unix and DOS. This method is the basis for Intellicorp's Knowledge Engineering Environment (KEE) expert system development tool for the 386.

Computers of the 80286 class are often sufficient for delivering AI-embedded applications. And, where existing 286s don't deliver enough power or memory, 386 add-in boards costing less than $1,500 offer an easy and economical upgrade path.

When the fully-implemented OS/2 appears, it will contain most aspects of Unix that make it supportive of AI development projects, plus the graphics and windowing capabilities that now have to be patched in on the 386. But the competition in the office between Unix and OS/2 won't be decided for some time to come, even if OS/2 Extended Edition keeps to its November 1988 release date.

We see no reason to wait for OS/2 before embarking on an AI project. Unix is here now, and can provide the capacity for developing any imaginable AI-included office application at reasonable cost. Also, Unix will remain for some time the best hope for enabling applications to talk to each other in a multivendor environment across all levels of hardware. Monolithic IBM shops may have to wait for Systems Application Architecture (SAA) to become real, but no one else has to.

The move toward a Unix standard, as evidenced both in the Posix effort and in the AT&T-Sun alliance, can only enhance Unix's future in supporting AI development. The more certain an organization can be that its applications will remain viable even if the hardware they were built for is replaced, the safer it is to embark on an AI project.

Artificial Intelligence Defined

You can find almost as many definitions of artificial intelligence as there are people working in the field. We begin by defining intelligence as the ability to—

- Take in information,
- Recall it when it is needed,
- Make connections between various items of information,
- Use these connected pieces to reach a conclusion, and
- Explain how the conclusion was reached.

Now, if *artificial* means "something created by human effort rather than by nature," then we can approach a useful meaning of artificial intelligence: the branch of computer science engaged

Variety of AI Applications

IN ADDITION TO EXPERT SYSTEMS, several other technologies developed in university and industrial artificial intelligence laboratories are making their way into applications for business and industry.

VOICE SYSTEMS. Speaker-independent word recognition and understanding continuous speech are the twin challenges of voice systems.

Today, users of voice systems must teach the computer to recognize each word they utter by associating the spoken word with characters entered at the keyboard. Because these systems recognize words by the audio frequency patterns they produce, a system taught by one person may not recognize speech of another person, and may even stumble when the original user has a cold.

In addition, just as a person hearing an unfamiliar language cannot tell where one word ends and the next begins, words that run together are generally unrecognizable by the computer system.

NATURAL-LANGUAGE QUERY PROCESSING. Getting information out of a database is a lot harder than getting it in. The goal of natural-language work is to build computer systems that understand a user's instructions no matter how they are worded—a goal much easier said than done.

Human beings can communicate without confusion when they share a body of knowledge about the subject augmented by an understanding of the world in general. Precisely what that understanding consists of and how to get it into a computer are subjects for research in natural-language processing.

Meanwhile, commercially available natural-language query systems (Intellect, Clout, Q&A, for example) can learn a finite vocabulary of words and phrases to be used in asking questions of a relational database, but none yet can deal with the ambiguity and imprecision of the spoken word.

ROBOTICS. Robots are everywhere on the factory floor. Equipped with digitizing vision systems and pattern-matching algorithms, they are used for inspection systems. Simpler robots perform assembly tasks. Mobile robots stand guard duty in warehouses. Voice-activated robots that understand continuous speech and contain embedded expert systems should begin to appear in the 1990s.

The Unix Workstation vs. the Lisp Machine

PUTTING ASIDE THE COST of acquisition, training, and a mountainous learning curve, there's nothing like a dedicated Lisp workstation for building AI applications. For one thing, Lisp is the mother tongue of AI development.

The level of functional integration on a Lisp machine is unparalleled anywhere else; a programmer can write and edit code, compile, execute, and debug with gratifying ease. Execution of Lisp code may be ponderous on multipurpose machines; on a Lisp machine, it's satisfactorily crisp, largely a product of superior memory-management techniques.

That's the good news. The bad news is mostly dollar signs. A ticket to ride the dedicated Lisp machine costs around a quarter of a million dollars, counting hardware, software, and services—acceptable, perhaps, in the price-is-no-object R&D market populated by the likes of DARPA, General Electric, Boeing, and the best-endowed universities, but hardly anywhere else.

In addition, the Lisp machine lacks network communication and project management tools that are as good as those found in the Unix environment.

Still, if you ignore the question of cost, Lisp machines are the environment of choice for development projects. Hardly anyone talks seriously anymore about deploying expert systems on such costly equipment.

Says Gary Fine, director of product development at Intellicorp, whose expert system shell, KEE, runs both on dedicated Lisp machines and on Unix workstations, "Once our programmers are on the Lisp machine, they are certainly much more effective than they are on the Unix machines, because it's a symbolic programming environment. The Lisp machines were built for programmers, and the Unix machines were built to run the programs."

in making computers behave in ways that we would consider intelligent in a human being.

In the artificial intelligence community, there is a corollary: If you know how it works, it isn't AI. As is often the case, there's an important truth embedded in the humor. AI is a laboratory science. What we're talking about here is the application of AI technology—useful ways of doing things that have emerged from the AI labs and are available for use in computer programs.

A good deal of AI technology is now so deeply embedded in conventional software that its origins are all but forgotten. Windows were developed to meet a need in AI research; so was object-oriented programming. Pattern-recognition algorithms, voice and vision systems, robotics, rule-based programming, and natural-language processing systems are all outgrowths of AI work that continues to engage hundreds of computer scientists and thousands of graduate students today.

Indeed, the AI technology that is applied today constitutes sort of an interim report from AI researchers—some of whom wish their work had not yet escaped their control.

The basic distinction between AI applications and conventional computer programs is this: AI technology lets the user specify the desired result without saying how to achieve it. The two major productivity-enhancing applications of AI technology to date are natural-language query systems and knowledge-based, or expert, systems.

Natural-Language Systems

Natural-language (NL) systems ease the process of getting information out of a database by making it possible for users to address the computer more or less in their own terms. There are still severe limitations; the languages that people use to communicate with each other are too complex and ambiguous for a present-day computer to understand.

Try, for a moment, to think like a computer that has been asked this question: Not counting the managers, who has the highest salary in the MIS department?

As a person who speaks English, that question gives you no problems. If you had to answer it, you could ask the personnel department for a list of employees showing their names, job titles, and salaries. You'd draw a line through every entry where the title included the word manager, and find the largest number in the salary listing for those remaining. You'd know that the desired answer consisted of a person's name. Anticipating the logical followup question, you might volunteer that person's title and salary.

For you, easy; for the computer, impossible. What does "not counting" mean to a machine, even if it can count items in a set? What in the world is the "highest" salary?

An NL query system can store definitions for terms such as these, equating them with commands to be run against specific tables and fields in a database. That's about as far as NL technology can go to date.

Laboratory work continues on ways to build a natural-language system that includes built-in knowledge about the world in which it operates and can interpret statements by the context in which they appear. Such a system would, for example, know that a theft was not being reported in the statement "Jane was in the hospital; Dick took her flowers."

Expert system technology is considerably more advanced in terms of applicability to business problems than natural-language technology is. Today, plenty of examples exist of working

expert systems yielding significant returns on the investment in developing them. Not all of them embody human expertise, the hallmark of a real expert system. No less useful, some contain and reason about large bodies of knowledge contained in books and manuals.

What Expert Systems Are Doing

Expert systems have been put to work in the following problem domains:

- Project management
- Instruction and training
- Configuration of insurance policy computer systems
- Analysis and screening in hiring, credit approval, and expense account authorization
- Diagnosis in medicine, electronics, telecommunications, and mineral exploration
- Planning and scheduling in finance and manufacturing
- Simulation in manufacturing, design, and decision support
- Process monitoring and control in manufacturing and power plant operation
- Design in manufacturing, VLSI, and software development

Expert System Concepts

An expert system has three major components: a knowledge base, an inference engine, and an input/output (I/O) facility.

Expertise in a subject area—*domain* in AI terminology—is usually comprised of a variety of elements: vocabulary definitions, objects and the relationships between them, constraints, hypotheses, rules, heuristics, descriptions of processes, and experience gained from situations previously encountered.

Heuristics—educated guesses and rules of thumb—are often key elements in expert systems. They are, after all, what make an expert more valuable to the organization than a merely experienced practitioner. They guide the expert to select correct solutions quickly and efficiently, based on experience often beyond conscious recall.

The method of expressing domain expertise is termed the expert system's knowledge representation paradigm.

RULE-BASED PARADIGMS. Most familiar is the rule-based paradigm. Say, for example, that an applicant must show combined costs for housing and installment debt to be less than 32 percent of total income to qualify for a mortgage loan. That could be translated into the rule: IF the sum of an applicant's housing and installment debt costs is less than 32 percent of total income, THEN the applicant is eligible for a mortgage loan.

If that were the only criterion, then the expert system would need only an applicant's total income and housing and installment debt costs to determine eligibility. Of course, nothing is ever that simple. A typical rule-based expert system contains somewhere between 100 and 2,000 rules.

Automating Credit Reviews at GMAC

WORKING FROM FINANCIAL statements and credit reports, analysts at General Motors Acceptance Corporation (GMAC) review the financial condition of car dealers applying for credit. GMAC has a host of guidelines and procedures for performing these reviews. It's easy for a human credit analyst to overlook something.

GMAC built Analyst, an expert system, to automate the process. Its goal is to improve the quality, consistency, and completeness of the review process. The knowledge it contains comes from the GMAC manual and from a human analyst recognized throughout the company as an expert.

Analyst is written in Copernicus, from Teknowledge. A hybrid expert system development tool written in C, Copernicus supports rules and frames, and contains its own procedural language.

The system runs on Sun workstations. Eventually, it will be networked to an IBM mainframe. While developers figure out how to implement that interface, Analyst downloads mainframe data to a database resident on the Sun, using off-the-shelf tools and the Structured Query Language (SQL).

GMAC anticipates several benefits from this expert system. Cost reduction will result from the fact that MBAs will not be needed to perform the analysis. The company expects to decrease its risk by conducting a more thorough analysis that should spot marginal businesses sooner. As losses decrease, GMAC expects to reduce its margins, improving the company's competitive position. The system can also provide a service to GMAC's customers by pointing out ways to improve their credit worthiness. Finally, it can be used as a training tool for new employees.

Analyst is a large system, containing about 2,000 rules in addition to data in frames. It took four people a year to build an extensive prototype, and another six months to complete integration with the database and an interface with which its human users feel comfortable.

Engenious, GE Aircraft Design Expert System

IN THE PROCESS of designing aircraft engines, design engineers at General Electric's (GE's) Aircraft Engines Division use highly complex algorithms, mostly written in Fortran, that reside on large IBM mainframes and Cray XMP supercomputers. Developing the systems programmer-level skills required to establish the interface between these mainframes and the Unix workstations on which they do their design work is considered a poor use of engineers' time.

Engenious is a graphically oriented, intelligent interface between the engineer-user at a workstation and the mainframe that holds the required algorithms. After the engineer provides design parameters to the expert system, Engenious selects the proper program codes to invoke the algorithms on the mainframe system, and then returns the resulting values to the workstation. It acts as the engineer's intermediary.

The system is written in KEE, which, in turn, is written in Common Lisp. Currently, it runs on Sun, Apollo, and HP Unix workstations, and under VMS on the Digital VAX.

Frame-based knowledge representation and object-oriented programming are the primary paradigms used in Engenious. An on-going project at GE, the system reflects about four work-years of effort to date. Currently, individual design groups are expanding it by adding rules specific to their own project assignments.

Benefits are twofold. Engenious provides a common interface across a range of engineering workstations. Engineers can move from one group and one vendor's hardware to another with no time lost in learning a new machine. Additionally, it allows engineers to complete their jobs without spending time learning to be system programmers in order to access the computer systems that hold the design algorithms.

The system is currently licensed on 100 GE workstations, with field-testing just completed. Ultimately, GE expects to make Engenious the standard interface on all engineering workstations.

INDUCTION FROM EXAMPLES. Where examples of successful decisions are plentiful but the rules underlying them are unclear even to the experts, an example-based paradigm is more appropriate.

Here, the knowledge base consists of a decision matrix—a symbolic spreadsheet in which factors that enter into the decision (total income, installment debt payments, and housing costs, to continue the previous example) are arrayed across the top, permissible decisions (yes and no, in this case) are listed as conclusions, with data from previously handled applications forming the example set.

Given a sufficient number of examples, such a system will induce its own rules. To the user, during a consultation, its appearance is indistinguishable from a conventional rule-based system.

An interesting feature of this inductive scheme is that it can find causes that might otherwise go unnoticed—as, for example, in analyzing the results of test market data or an advertising campaign. If the results are anything other than random, then an induction system can point out the factors that were critical in the campaign's success or failure.

CASE DIFFERENTIATION. A variant of the rule-by-example paradigm is the case differentiation system. Here, the built-in knowledge starts with a conclusion and describes the attributes of a particular case that yielded that conclusion. Then the system is presented with a different conclusion and with a single attribute differentiating a case leading to that conclusion from the case previously introduced.

FRAMES AND SEMANTIC NETS. Many areas of knowledge, however, don't lend themselves to any of the foregoing knowledge representation paradigms. Frames and semantic nets deal with more complex relationships than either rules or examples can.

A frame is a data structure containing a slot for each attribute of a particular object. To continue the mortgage application example used above, a frame-based expert system might contain a frame named "Applicant," for example, with slots for identifying information such as name, address, telephone number, occupation, and employer, and additional slots for income and relevant expenses.

Another frame, called "Eligibility," would contain slots for minimum required income, required maximum ratio of housing costs plus debt payments to income, and so on.

Frames can have a parent-child relationship between them, in which case the child frame inherits all the characteristics of the parent. Values that are true for an entire class of objects—for example, the fact than an automobile has four wheels—need not be repeated for a specific instance of the class—e.g., in the Buick Regal's frame. This is an economy measure, both in terms of memory storage and of keystrokes during development and use.

Semantic networks, found in a few knowledge bases but more commonly implemented in experimental natural-language systems, link factual statements in a relational network. At present, the semantic net is the least developed of all knowledge representation schemes for expert systems, but the paradigm holds great promise, particularly for diagnosis and configuration systems.

Most often, frame-based and semantic net expert systems will contain rules as well, but the number of rules required is greatly reduced because of the economy of these more complex paradigms.

Inference Engines and Control Mechanisms

Reasoning about the contents of the knowledge base is the job of the inference engine.

FORWARD CHAINING. Forward-chaining inference engines look at the left-hand, or IF, side of a rule to determine whether the fact it asserts is true. If it is not, then the system moves on to the next rule and tests the truth of its premise.

When a rule is located whose IF clause is true, the forward-chaining system declares the right-hand, or THEN, side true as well. The rule is said to have "fired," and the inference engine proceeds to look for a rule whose premise matches the conclusion just found true.

Eventually, when no more rules can be made to fire, the system either reports its conclusion or announces that no conclusions can be drawn from available information.

BACKWARD CHAINING. By contrast, the backward-chaining inference engine begins with a goal or hypothesis and works backward through the rules in the knowledge base, considering those that are relevant to the goal.

If the goal is Z, then the backward-chaining system looks for a rule whose THEN side contains Z. Having found one, it then looks for another rule whose right-hand side contains a match for the IF side of the rule just fired, continuing until it fires all the rules necessary to prove Z true. If it cannot satisfy the requirements for Z, then it picks another hypothesis and tries again, or, if no other goal is available, it reports to the user that no conclusion can be reached.

Generally, forward-chaining systems perform more efficiently where a body of data must first be assembled to perform a task, as in configuring a computer system or selecting the best candidate to fill a hiring requisition. Backward-chaining systems lend themselves to problems where the user can choose from a list of possible outcomes, such as in diagnosing a mechanical fault or illness, given both a set of symptoms and an educated guess as to where the fault lies.

I/O and the Outside World

The third component of the knowledge-based system is information pertaining to the problem at hand. Some or all of this information may be entered at the keyboard by the person running the consultation. Additional input may be obtained by plucking data from a spreadsheet or database, or by reading the gauges and meters of a bank of process-monitoring instruments.

In addition to normal user interface considerations, then, an expert system needs a competent and efficient means of reaching

Turbomac, Diagnosing Causes of Vibration Problems in Large Rotating Machines

THE HARTFORD STEAM BOILER Inspection and Insurance Company (HSB) is in the business of insuring machinery and equipment and of providing related inspection and engineering services. Among the machines they insure are large steam and gas turbines, motors, generators, and compressors. All of these machines depend on a large, rapidly rotating shaft to convert energy into work.

Most malfunctions in such machines manifest themselves as abnormal vibrations. Some abnormal vibrations are due to nonserious causes and can be tolerated until the next scheduled maintenance overhaul. Others are caused by serious conditions that could destroy the machine if not corrected immediately. Obviously, it is important to have a means of identifying the causes of abnormal vibration quickly and accurately.

Sources of Turbomac's expertise are a set of diagnostic charts developed by a well-known expert in diagnosing problems in turbomachinery, the expert himself, and members of the staff of Radian Corporation, an HSB subsidiary and the organization that developed RuleMaster, the system's underlying language.

RuleMaster is an inductive language in which knowledge is expressed primarily as a series of examples showing the conclusions that might be drawn from the presence of various combinations of attribute-value pairs. In addition to some 60,000 lines of RuleMaster code, Turbomac is augmented by about 3,000 lines of C-coded routines.

At present, Turbomac deals with 42 possible causes of abnormal vibration. Its knowledge base consists of approximately 9,700 rules relating the presence or absence of 144 symptoms of the existence of these problem sources. It was developed and currently runs on a Sun 3/160. Time to run a diagnosis ranges from 5 to 15 seconds.

Ultimately, HSB plans to make Turbomac available online to its customers by way of a telecommunications hookup. The company anticipates that use of Turbomac will lead to a decrease in casualty losses due to machine malfunction.

LISP-ITS, A Computer-Aided Instruction on Programming in Lisp

EQUIPPED WITH SOME 400 rules describing the way an exemplary beginner would write Lisp code, the Ideal Student Model knowledge base assumes only that the student knows material already covered in an accompanying Lisp programming course. It tries to deal with Lisp generically; where various dialects of Lisp differ significantly, Lisp Intelligent Tutoring System (LISP-ITS) follows the conventions of Common Lisp.

LISP-ITS contains a series of exercises that require the student to write Lisp code. It monitors the student's keystrokes during a programming session, rather than asking the student direct questions and evaluating the answers.

A second knowledge base, the Bug Catalog, contains rules defining more than 1,000 mistakes and misconceptions common among beginning Lisp students. When the Ideal Student Model determines that the student is offering suboptimal responses, it calls the Bug Catalog to diagnose probable causes of error.

A third knowledge base consists of tutorial rules that determine when to interrupt the lesson and offer help to the student, what type of help to provide, and when the student has mastered the current topic well enough to be allowed to take on new material.

LISP-ITS was developed at Carnegie-Mellon University, Pittsburgh, by researchers in psychology and cognitive science. It has been in use there since 1984 and is available for purchase from Advanced Computer Tutoring, Incorporated, of Pittsburgh.

It has been shown to reduce the amount of time spent on homework and to improve the test scores of students using it, when compared with those of students doing the same work without aid. Its performance was inferior when compared with a human tutor, measured in terms of the length of time students spent on homework assignments and their subsequent test scores, but, given the scarcity of Lisp tutors, LISP-ITS can be considered successful in shortening the learning curve for beginning students.

into the outside world. It may also need the capacity to simulate graphically the meters and dials, or wiring diagrams and the like, whose data enters into the system's conclusions. The more the computer's display looks like the environment where the user is accustomed to working, the greater will be the expert system's acceptance and the resulting productivity gain.

Reports and Explanations

An expert system's output is its conclusions. In rare instances, a screen display of conclusions is sufficient. More often, some form of report is required, which may mean feeding the system's conclusions into a word processor and/or graphics package.

One of the characteristics of a real expert system is its ability to explain itself. Some can explain both why they want a specific item of information and how they reached a particular conclusion; others can do just the latter.

This feature is especially significant in training systems, when the user requires reassurance that the system knows what it's talking about, and during the development and debugging process, when being able to trace the way the system reasons is key to spotting errors in logic and missing rules.

A Unix User Testifies

Ted Kowalski built the Design Automation Assistant (DAA) for his doctoral thesis at Carnegie-Mellon University and finished it at AT&T Bell Laboratories, where he is a member of the technical staff. DAA is an expert at designing VLSI chips using a high-level program description and constraints such as maximum chip size and required speed. One of Kowalski's thesis reviewers, a 25-year veteran in chip design at IBM, told the thesis committee that Kowalski's expert system was as skilled as one built by any of the better designers at IBM.

"This particular application couldn't have been done on anything other than the Unix boxes I was using for three reasons," Kowalski says.

First was memory usage. Much of the work an expert system does involves searching memory for a pattern in the knowledge base that matches the current input. Kowalski's current Unix machine has 256 megabytes of real memory. "I have grown more and more greedy in my searches," he says. "I look at more and more designs, and I need more and more real memory. Virtual memory is cute, but you need real memory to back it up."

Reason two is the ability under Unix to grow the system from a strictly rule-based prototype written in OPS-5 into a hybrid system containing both rules and C routines, with a resulting improvement in execution by a factor of 10.

The third reason is that the world of Computer-Aided Design (CAD) is the world of Unix and C. Kowalski says, "I have, quite literally, megabytes upon megabytes of simulators, compilers, editing tools, layout tools, graphics tools, all written in C. By being able to tie into those systems, I've gained a tremendous amount of leverage in being able to quickly create my system."

Software Tools for Expert System Building

While expert systems have been written in conventional programming languages such as Basic, Pascal, and C, Lisp and Prolog are more commonly associated with expert system development projects because of their symbolic processing capabilities. Lisp is generally favored in the United States, Prolog in Europe and Japan.

Expert system development software, commonly referred to as expert system shells, offers a convenient way to approach a project. The tools are roughly analogous to database application generators—if you can accept the structure they impose, they save a great deal of work.

An expert system shell has an inference engine built in. It may chain forward or backward, or both. It may allow the developer to specify whether the system should find all possible answers to a problem, or stop after the first good solution is identified.

It may search for answers first in breadth, looking along the top level of branches on a decision tree, or first in depth, traversing the length of one branch looking for a match and starting on the second branch only if the first search fails.

Also built into the shell is the framework of a user interface, with windows for asking questions and accepting answers, and a format for reporting conclusions. Most shells allow access to the rest of the computing environment; sooner or later, developers want to do something that the shell isn't built to do, and they must resort to routines written in C or assembly language to supplement the tool's capabilities.

The best expert system tools include more than one knowledge representation scheme; rules and frames are the most common. Example-based systems are more limited in the kinds of knowledge they can express, but, where they are appropriate, they more than make up for this limitation in the ease with which they can be programmed.

The Development Life Cycle.

Expert systems haven't been around long enough yet to have associated with them the kind of formal life cycle descriptions that accompany conventional database application projects. What is known, however, is that the differences aren't great. What follows is a composite life cycle schedule, based on a dozen or so projects of widely varying scope and complexity. Its purpose is to describe the flow of a project, not its timing.

IDENTIFY THE PROBLEM. A candidate problem is proposed and evaluated according to a checklist such as the one in the box "Finding a Suitable Subject" above. Requirements analysis consists of breaking the problem down into its smallest possible components, describing what a satisfactory solution would look like, and then picking the easiest, most pressing, or most inter-

Design Automation Assistant, VLSI Chip Design at AT&T Bell Labs

WHAT MAKES VERY large scale integration (VLSI) design so complex is the number of interactions among decisions about architecture, logic design, circuit design and layout, and the difficulty of predicting or tracing the consequences of a specific design choice.

AT&T Bell Laboratories has an expert system, Design Automation Assistant (DAA), that produces specifications for VLSI chips, given programmed constraints such as size and performance desired. The system's goal is the automatic design of large (100,000 transistor) VLSI systems whose quality is competitive with those produced by human beings when measured in terms of performance and cost.

Although still in testing, DAA recently became the first expert system to go from a top level description all the way to producing the chip in silicon. Critiqued by a 25-year veteran IBM chip designer, DAA was rated as functioning on a level with one of IBM's better designers, although somewhat below expert level.

DAA's knowledge base comes from about 600 hours of interviews with chip designers at AT&T and Intel. Knowledge acquisition began with a series of conversations with the designers, three experts and one novice. All of the conversations were tape-recorded and consisted of questions such as: Where do you start in VLSI design? What do you do next? Is there a next-best decision?

Then came a prototype of very limited capabilities that could produce designs for the human designers to critique. Effort expended to this point totaled about three work-months.

Then the system was required to design a Motorola 6502 microprocessor, the heart of the Atari video games at the time (1983). The experts were asked to comment on the design, then the system was taught the experts' modifications, and the process continued in an iterative loop that consumed about 300 hours of interview time before the system produced a satisfactory 6502 chip. Now, some 300 additional hours later, DAA has produced designs for other, far more complex chips.

DAA's underlying language is OPS5. Portions of the system determined unlikely to change are recoded in C.

Vendor List: Expert System Development Tools

ART (Automated Reasoning Tool), Version 3.1
Inference Corporation
5300 West Century Boulevard
Los Angeles, CA 90045
(213) 417-7997
For Sun workstations 3620, 3640
Single copy $60,000.

Copernicus
Tecknowledge, Inc.
1850 Embarcadero Road
P.O. Box 10119
Palo Alto, CA 94303
(415) 424-0500
For Sun, Apollo
Site license $12,500 per year, or a $30,000 one-time fee.

EXSYS
Exsys, Inc.
P.O. Box 112477
Albuquerque, NM 87192-0247
(505) 256-8356
Up to 16 users, $5,000
EXSYS Professional to be released third-quarter 1988

KEE
Intellicorp, Inc.
1975 El Camino Real West
Mountain View, CA 94040-2216
(415) 865-5500
For Apollo, Digital, Symbolics, Sun, Texas Instruments, and Xerox Unix workstations, $30,000

KES
Software Architecture & Engineering, Inc.
1500 Wilson Boulevard
Arlington, VA 22209
(703) 276-7910
$7,000

Knowledgecraft
Carnegie Group, Inc.
5 PPG Place
Pittsburgh, PA 15222
(412) 642-6900
Single CPU license $27,500 for Sun

OPS-5 Version 3.03
Computer Thought
840 Avenue F, Suite 104
Plano, TX 75074
(214) 424-3511
Single copy of Sun version, $2,995

Rulemaster II
Radian Corporation
8501 Mo-Pac Boulevard
P.O. Box 201088
Austin, TX 78720-1088
(512) 454-4797
For Sun, Hewlett-Packard $7,500
Xenix version $2,495

esting (depending on management's wishes and/or the leanings of the development team) to attack first.

FIND AN INFORMATION SOURCE. If the source of expertise is not immediately apparent, one must be found. Sometimes the source of expertise appears before the problem is identified; for example, the Campbell Soup Company decided to build a diagnostic system for its soup-cooker because the company's expert in cooker repair was about to retire. The resulting system saved 44 years of experience that would otherwise have been lost.

Human experts must be recruited with care. They are already busy, and aren't likely to relish giving time to an expert system project—particularly if they haven't already announced their intention to retire. A good book is better than an uncooperative expert.

REFINE THE PROBLEM DEFINITION. The result should be a clear, brief statement of the problem the system is to solve. Required output, such as reports of conclusions, should be tentatively identified now.

SELECT DEVELOPMENT SOFTWARE. Different kinds of problems lend themselves to different software tools. Survey product literature, seek demonstrations, consult with vendor representatives and, perhaps, with a consultant in expert system software selection.

COLLECT EXPERTISE. Have a member of the development team observe the expert at work on an instance of the problem in question, and/or interview the expert about the problem's characteristics and methods of solving it. The interviewer—knowledge engineer, in AI terminology—should have learned as much as possible about the domain before approaching the expert.

BUILD A WORKING PROTOTYPE. Code a small working prototype that does one thing as well as possible, given the present level of understanding of the expertise involved. Turn it over to the expert for comment and correction. Be prepared to start over at least once. "Build one to throw away" is a common bit of advice in expert systems circles. Repeat the last two steps—again, and again, and again. Iteration is the watchword of

expert system development. The goal is to have a computer system that performs as accurately as the person whose knowledge it contains. Two translation processes are going on here: from expert to knowledge engineer, and from knowledge engineer to computer, usually with a number of programmers involved.

DEPLOY A PILOT. Let the system work in tandem with the expert for a while, as a pilot. Make corrections when the system is wrong. Gradually, the user community will come to trust it.

MAINTAIN IT. Once the expert system is in regular use, maintain it. Maintenance of a well-designed expert system is not difficult. It should be possible to add new rules as information becomes available without contradicting old rules. Data that is subject to change (e.g., part numbers in a configuration system, ratios in a credit rating system) should be kept separate from rules (in frames, for example) to facilitate modification.

Conclusions

Unix offers a highly supportive environment for development of projects that embody artificial intelligence techniques, such as expert systems. Key Unix advantages are multiuser and multitasking capabilities, the ability to field a developed system across the entire range of computing architectures, and the richness of available software tools, both general and AI specific.

Unix workstations and 80386-class computers are ideal for group development efforts. The 80286 is often adequate for deployment; where a 386 is needed, upgrading already-owned 286 machines is an economical option.

Nothing that OS/2 offers is sufficiently compelling to lead us to recommend waiting for it. By the time OS/2 matures into a product worthy of the investment implied by an AI project, a Unix-based expert system begun today can have paid for itself several times over.

A New Office Umbrella

Can It Save Good Old NBI?

By Ronni T. Marshak

POOR NBI IS NOT doing very well. One of the few survivors of the old generation, the dedicated word processing vendor, it has been losing money for the last nine quarters. In a bold move—at least for a conservative company like NBI—the company is betting everything on a new product strategy based on industry standards with Unix as the kick-off point. (For more information on NBI's financial situation, see the *Office Computing Report*, Vol. 11, No. 5.)

ADHERING TO STANDARDS. NBI's hopes rest on two new products: Legend and OfficeWorks. Both products are based on industry-standard operating systems and operate on industry-standard platforms and workstations.

Legend is an impressive document processor which combines standard word processing, graphics editing, and desktop publishing in a $695 package. It runs on a PC (286- or 386-based) platform running Windows 2.0. A Macintosh version is in the works. (For more information on Legend, see the *Office Computing Report*, Vol. 11, No. 5.)

OfficeWorks is a software environment that includes applications such as document retrieval, electronic filing, and a macro facility. It runs under Unix and supports DOS, Mac, Unix, and OASys workstations. At first release, the product will run on the NBI 520 (16 users) or 570 (64 users) servers with a minimum of 4MB RAM and 106MB online storage. But ports to other Unix hardware are planned. Both OfficeWorks and Legend will eventually move onto the OS/2 platform under Presentation Manager.

OfficeWorks

NBI is having a tough time verbalizing exactly what OfficeWorks is and how to position it. Is it

a. A desktop environment?
b. A document retrieval system?
c. A macro facility?
d. A filing system with version-tracking?
e. An office system with E-mail and calendaring?

And the answer is

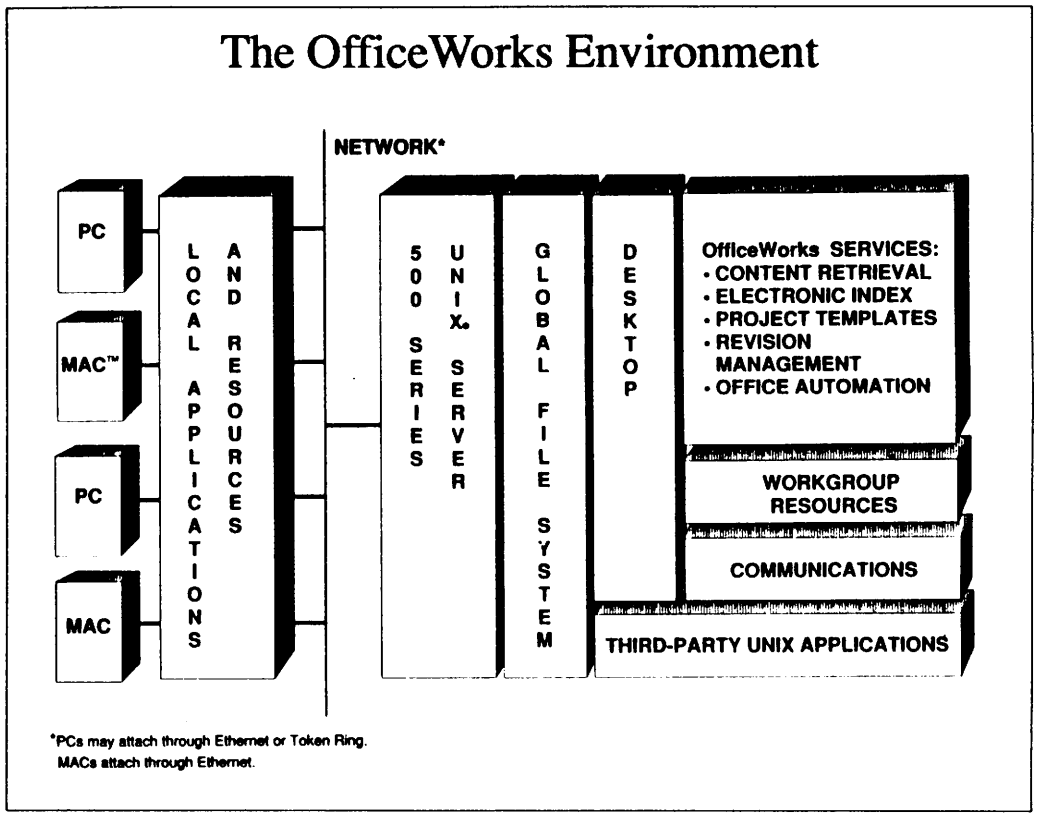
f. All of the above.

Now this is a problem. If NBI can't describe OfficeWorks in 20 words or less, how can anyone else? And customers don't stand still for long recitations. You have to tell them briefly and in a focused manner exactly what it is you expect them to buy. Therefore, NBI's challenge is to find some way of explaining the essence (dare we say Gestalt?) of OfficeWorks without resorting to giving a laundry list of features.

DESKTOP ENVIRONMENT. Most systems force you to choose among PCs, Macs, and Unix terminals. Oh sure, they offer terminal emulation on the micros or downloading of the word processor, but you still have to buy into one environment or another. OfficeWorks attempts to solve that problem by allowing users to work in their environment of choice, be it PC, Mac, OASys (NBI's proprietary system), or Unix. The company provides a Microsoft Windows Terminal emulator for Unix and OASys

users. Unfortunately, they also need to use a Windows emulator on the Mac version for certain functions, but this will be remedied in the next release. The OfficeWorks system itself runs under the Unix operating system and requires an NBI 520 or 570 Unix minicomputer. NBI is planning on porting the system to other platforms—Unix-based, OS/2-Presentation Manager-based, and AU/X—as soon as possible, though the company is reluctant to commit to a time frame at this early date.

The OfficeWorks Desktop is a graphical user interface based, as you may have guessed, on Microsoft Windows. PC users are always in the Desktop environment, except when they are using a PC application that is not Windows-based or when they go into DOS or Unix from the main menu. Even system administration tools, such as backup, are available from the Desktop.

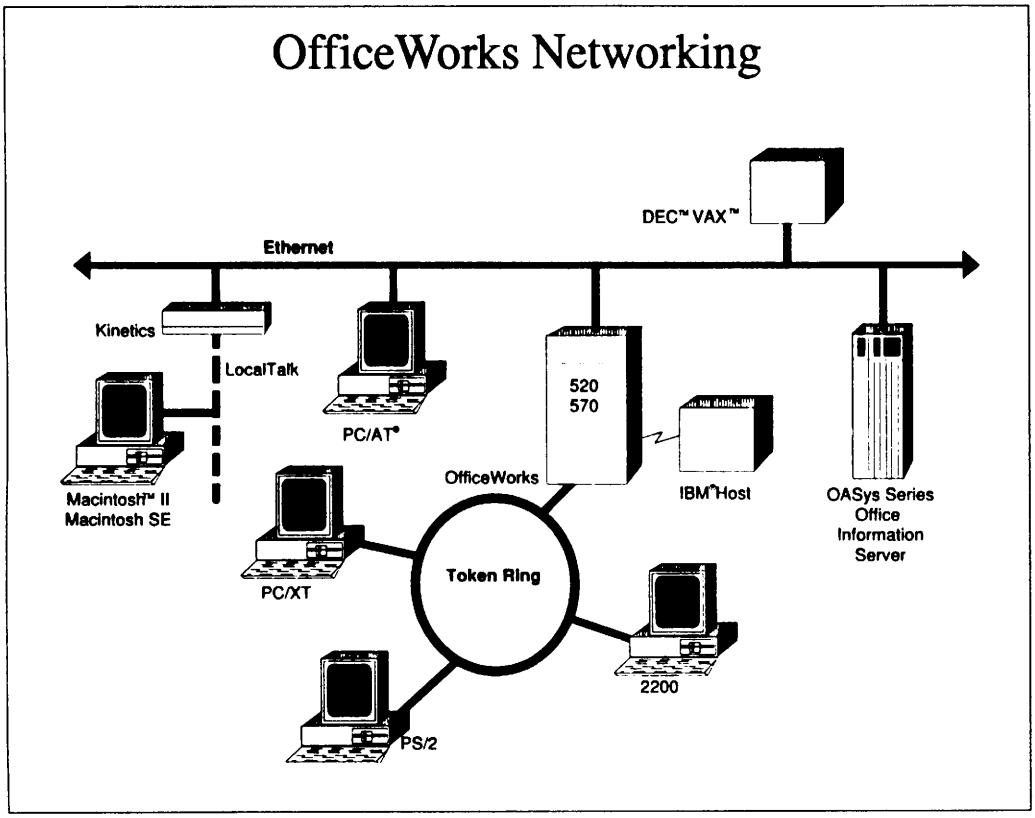


Users are free to use their favorite applications on their familiar workstations. So a single workgroup might consist of three PC users, one using WordPerfect and Lotus 1-2-3, another using Legend, and the third using XyWrite and Excel; a Mac user with Word and the Mac version of Excel; and several OASys users running NBI software. All members of the workgroup can access and edit each other's documents using the document processor of choice. To facilitate in document exchange, OfficeWorks incorporates NBI TIE, a document-conversion utility based on technology from KeyWord Incorporated. If a user tries to open a file created in a different editor, OfficeWorks gives the user a message identifying the original editor. The user can then invoke the document-conversion utility.

This is all well and good, but it takes a lot of steps. First of all, in order for OfficeWorks to report the application in which a file has been created, that information must be entered on an Electronic Index card (see "Document Retrieval" below). Then the user must request document conversion from the original format to his or her application's format. It makes more sense for the system to automatically notate the application of origin on the Electronic Index and to automatically convert the document. (Of course, the conversion will not always be 100 percent accurate. KeyWord does an admirable job of converting the integrity of a document, but there are always the problems of moving from a word processor that offers a specific feature—such as outlining—to one that doesn't.) This automation is possible using the Electronic Index and Project Templates (see "Project Templates" below). And, indeed, NBI will be providing a macro to execute this function. With this, the group editing process is truly transparent, and users can play in their own applications environments without any unnecessary hardship.

DOCUMENT RETRIEVAL. OfficeWorks provides full-text retrieval of documents—Content Retrieval Service (CRS). "Ho hum, that's nice," you say. But wait. There's more. OfficeWorks will search through all drives on the network, including PC hard disks. (Mac hard disks are not accessible. Apple server disks will be accessible in a future release.) To protect a user's privacy, only public documents (i.e., those registered with CRS either specifically by the user or automatically as part of a project template) are searched.

CRS supports standard Boolean searching, weighting analysis, frequency analysis, "fuzzy" matching, word-root analysis, and proximity analysis in order to determine the priority of a matching file. To ensure that the system understands the search query, CRS interprets the search phrase and presents what it believes is the query criteria back to the user for approval or modification. (This helps eliminate problems when the user enters an ambiguous search phrase.) CRS can search for documents in Legend, DCA, OASys, or ASCII format. The problem with this is that most PC-based word processors don't store natively in either DCA or ASCII. Perhaps a Project Template (see below) is necessary to automatically store documents in one of these formats transparently to the user. But then you are faced with the storage overhead resulting from storing a document twice.



But in addition to the full-text searching, the product provides an Electronic Index function. The Electronic Index is like an index card with which you tag a file (all file objects, including text, image, voice, spreadsheet, etc., are supported) from any application. The index not only identifies a file, but it also can define the file's behavior in relation to various OfficeWorks applications. For example, upon storing a file, the index card could tell the system to print a copy on a specified printer, check the file into the version tracking system, and automatically reindex the cards in the directory. The Electronic Index automatically includes certain default fields, such as creation data and author's name. Users may add their own descriptors, such as department or project name. If a specific file type is selected—for example, a memo—certain fields on the index card will automatically be inserted into the document, so the "To:", "From:", "Date:", and "Subject:" information will be filled in when the document is created.

OfficeWorks allows you to create index cards for off-line (in file cabinets) or archived documents.

You can search the index cards by fields or combine an index card search with CRS for a more targeted file search.

The major problem with the Electronic Index is the inconsistency of human nature. Users are likely to avoid filling in the cards, severely limiting the potential of the system. The more defaults entered by the system, the fewer problems with card fill-in. NBI does not require that a card be completed before creating a new file. So a user can quickly key past the index card, leaving all customized fields blank.

PROJECT TEMPLATES. Project templates are, in the simplest form, scripted macros which can be iconized. When used with the Electronic Index, TIE, and the office applications of OfficeWorks (see "Office Applications" below), they can be very powerful tools, automating routine applications for individuals or workgroups. Unfortunately, at this time, it takes an experienced user (with the level of ability to, say, write a batch file) to create a template, and the templates are restricted to OfficeWorks applications. They cannot go into another application and run an operation.

This is where we see the most potential in OfficeWorks. The first step is offering NBI-developed scripts for common applications, and if these scripts could be easily edited, so much the better. But the goal is to make the template simple for an average user to develop. Until then, the potential of OfficeWorks will not be realized.

VERSION-TRACKING. OfficeWorks version-tracking is useful for organizations that must keep track of the status of a document. What's the latest version? Who updated it? And so on. The system keeps a log of changes to a document by date and user. Only the latest version is available for editing on that track. Editing an earlier version starts a new history track with that version as the original.

This is not a co-authoring tool. There are no redlining or commenting facilities. But it is an excellent basis for one. We hope that NBI will offer such a system by the next release. It seems a logical next step, and the preliminary tracking structure is already in place. Offering this kind of collaborative tool will add to the value of OfficeWorks as a workgroup solution.

OFFICE APPLICATIONS. Both electronic mail and calendaring are offered as part of OfficeWorks. NBI Mail is a compound-document, E-mail system that provides both local and remote services. Any DOS, Mac, or Unix file can be sent as an attachment (including text, data, graphics, voice, etc.). The system offers all the standard mail features, including guaranteed delivery, mail store and forward, and a variety of print, save, and notification options. NBI Mail also provides transparent interfaces to DEC VMS Mail via the Simple Mail Transfer Protocol (SMTP) and Unix Sendmail.

LATE BREAKING NEWS

The Open System Foundation

by Judith S. Hurwitz

The Unix world has become a battlefield with two opposing armies poised for war. The gauntlet has been dropped, and the cannons are aimed. In the middle are the huddled masses, wondering what all this fighting will mean to their livelihood.

Although it may seem overly dramatic to characterize the establishment of the new Open Software Foundation as a war, in the software standards arena, it is indeed a dramatic occurence.

A BIT OF BACKGROUND. As you probably know, AT&T and Sun joined forces a few months ago to work on the next release of System V. This version is being developed for the Scalable Processor Architecture (SPARC) chip that both companies are committed to.

Although it is nice to see vendors working together, this alliance infuriated the rest of the Unix competitors, including big shots Digital and Apollo. They, along with a number of other vendors, created an ad hoc organization called the Hamilton Group to jointly confront AT&T with their concerns. The group contends that the intent of AT&T and Sun was to make Unix proprietary since the operating system would be written for a specific chip—SPARC. Therefore, Sun would have access to beta code as long as six months before any other vendor. Additionally, they feared that the new operating system would not be as efficient on standard platforms such as the Intel 80386 and the Motorola 680x0 as it would be on a SPARC platform.

AT&T has assured the industry that this is not the intent and that all vendors would have equal access to Unix. But the rest of the industry isn't buying it.

THE FOUNDATION. The Open Software Foundation is comprised of some of the most important computer vendors, including U.S.-based IBM, Hewlett-Packard, Digital, and Apollo, and European vendors such as Siemens, Nixdorf, Honeywell Bull, and Philips. These vendors will all be sponsors of the newly formed consortium. Each has agreed to provide $4.5 million a year for three years. Therefore, OSF is starting life with a bank account of more than $90 million. The goals of the organization are clear: to "define specifications, develop a leadership operating system, and promote an open, portable application environment." The principles of the foundation include:

- Offerings based on relevant industry standards
- Open process to actively solicit input and technology
- Timely, vendor-netural decision process
- Early and equal access to specifications and continuing development
- Hardware-independent implementations
- Reasonable, stable licensing terms
- Technical innovation through university/research participation

In addition, the OSF has taken on a motto of sorts: "Portability, Interoperability, and Scalability." This motto indicates the direction the group plans for Unix.

In order to spring into action, each sponsoring vendor has contributed some of its technology to build the foundation. Therefore, this standard will be based on a conglomeration of some of the best Unix technology the industry has to offer. IBM's AIX operating system will serve as the core operating system. This is quite a coup for IBM, which has just decided over the last year that it would pour some substantial resources into the Unix marketplace. It is not the current version of AIX that will form the foundation, but rather its successor. Other vendors have also added their pet technologies. Apollo has offered its Network Computing System (NCS), Honeywell Bull has offered its system-based multiprocessor architecture, Digital brings its user interface toolkit and style guides for X-Window, Hewlett-Packard added National Language System (NLS), Nixdorf proposes its relational database technology, and, finally, Siemens is offering its OSI protocol support. Philips, a late entry, has not made its offer public as of this writing.

CONCLUSION. Had AT&T created an industry council back in February when the so-called Hamilton Group began to complain about the AT&T/Sun plans, OSF would never have been formed. But rather than encouraging open participation by the rest of the Unix vendor community, AT&T tried other ways to sooth ruffled feathers. These approaches were simply too little too late.

In one respect, it is too bad that things did not work out differently. The notion of a Unix industry where there was indeed a single version of Unix adhered to by all vendors in the marketplace is very powerful and positive. The newly formed foundation is destined to throw Unix users and software suppliers into turmoil. They will have the sense that Unix is moving away from a positive direction that has been set over the past few years.

On the other hand, the amount of money and the clout of the vendors that are coming together to form OSF cannot be minimized. If they do as they promise and work in concert, the potential exists for coming out with an even stronger standard, incorporating the best that this industry has to offer. It should be able to pull in some advanced research from the university community as well as from the proprietary world.

In the next issues of *Unix in the Office*, we will keep you informed and help interpret just what the Open Software Foundation will do and what it will mean in the long term to the computer industry and the future of standards.

The calendaring system, Time Management, integrates with NBI Mail. It includes features such as public and private calendars, meeting scheduling, and resource calendars.

Both E-mail and calendaring can be integrated into Project Templates. And, incidentally, users can choose their own Unix-based mail and calendaring systems instead. These will not, however, share the Windows-based interface.

NETWORKING ISSUES. PC and OASys-attached workstations are connected to OfficeWorks (running on an NBI 520 or 570 Unix server) on Ethernet via TCP/IP or on a token ring LAN. The system supports the Microsoft network protocols for PCs. Macs plug into LocalTalk using AppleShare. The networked Macs connect to Ethernet via a Kinetics gateway. Wide area networking is supported over RS-232C dedicated lines.

OfficeWorks also offers connectivity to IBM and DEC VAX mainframe environments and to other NBI systems. It also provides users with asynchronous communications capabilities and connectivity between PCs and Macs.

The product provides 3278/9, 328X, and 3770 Remote Job Entry (RJE) terminal emulation and transfer of DCA-formatted files for communication to IBM hosts. Communications with Digital VAXs is provided via VT-100 and VT-220 terminal emulation, as well as by VMS Mail Exchange for file transfer and bidirectional mail. Access to outside sources is provided via asynchronous communications using X-Modem, Kermit, or UUCP.

Future releases of OfficeWorks will include support for LU6.2/SNADS, X.400, and X.25. Having these protocols is vital. We hope that "future release" in this case means "next release." NBI is already late by not providing these. It mustn't get further behind.

IN TOTO. OfficeWorks is really proof that the whole is often greater than the sum of its parts. Just another file retrieval/electronic indexing/version-tracking/scripted macro facility isn't amazing news, but put it all together with a desktop environment and the ability to work on your native system, exchanging files transparently over a network, and you've got something.

Our impression is that NBI's development team is just beginning to recognize the possibilities, and they seem to be a bit awestruck by them. We recommend that, instead of just focusing on finishing the product by the scheduled delivery date, they take a brief step back, take a deep breath, and let their imaginations soar. Who knows what they'll come up with?

DISTRIBUTION. NBI's primary concern is to avoid repeating past mistakes. The last time the company came out with a standalone word processor (code-named the Black Hole Project), no cohesive marketing strategy was developed. And the product, which was ahead of its time, disappeared. This time around, NBI is carefully planning all aspects of distribution for its new products.

OfficeWorks. Besides sales through NBI's direct force and exclusive dealer channels, OfficeWorks is targeted as a product for value-added resellers (VARs), which NBI calls sales partners. NBI will sell the hardware and software to VARs, which will, in turn, team-sell with both the NBI direct sales force and dealers. This is NBI's way of selling to vertical, industry-specific markets without excluding its regular sales force.

The VAR arena is new to NBI. And this very competitive market can be cut-throat. NBI will have to learn quickly how to succeed in this market, especially with the team-selling approach the company is planning.

The direct and dealer sales force will also sell OfficeWorks into their customer bases.

Availability and Price

OfficeWorks will be released in August of this year, though the version-tracking facility won't be available until the end of October. Pricing is predicted as follows:

	Server Model 520 Up to 16 Users	**Server Model 570** Up to 64 Users *(direct connect— PCs may be indirectly attached through token ring or Ethernet)*
OfficeWorks	$8,900	$14,900
Document Interchange (TIE)	N/A	N/A
Version Tracking	$1,500	$2,400
Unix Operating System	$2,000	$3,000
Server Hardware	$17,900 - $26,400*	$42,900 - $110,000*
TOTAL	$30,300 - $38,800+	$63,200 - $130,300+

* depending on configuration of memory and disk
\+ pricing for TIE is not yet available and will add to the total cost

A Total Solution

With OfficeWorks as the umbrella environment (NBI Office, so to speak) and filing system, Legend as the document processor, and all other DOS, Unix, and Mac applications available to all users in their native environment, NBI has a total solution.

Existing OASys users can be part of the system, which offers them a migration path from a languishing product line. Indeed, we anticipate that, within three years, almost all OASys installations will have migrated out. Whether they migrate to OfficeWorks depends a lot on how NBI meets the potential of the system.

Conclusion

There are a lot of logical steps NBI can take. We would like to see OfficeWorks front-end all the popular Unix-based relational database management systems (RDBMSs) and SQL. The Electronic Index cards could be used to do simple querying and data entry. And, as databases begin to accept more and more data types (voice, image, etc.), OfficeWorks' ability to handle these data types would make it a valuable front end.

The first release will not even begin to fulfill the promise OfficeWorks, but it will lay the foundation. NBI has to build on that foundation, opening it up to more standards platforms as planned and providing transparent links between and among existing applications.

NBI is following a new vision, and we believe it is a healthy vision. No longer does the company insist on using only its own products. Like its competitors, NBI has bowed to industry standards. And the developers have come up with a far-reaching solution, reminiscent, in some ways, of Hewlett-Packard's NewWave environment. (NewWave offers much more on the macro—i.e., agent—side, and better integration between applications—two-way live links in compound documents. OfficeWorks offers more on the filing, retrieval, and document history end.) But NBI must live up to the vision. New products, commitment to standards, and new marketing avenues are all the right moves, but NBI has to execute its new strategy with finesse. End users and VARs need to feel confident that the company knows what it is doing and will do it with elegance. NBI cannot afford to make mistakes at this point. Unfortunately, this is NBI's last chance.

Now, more than ever, you need the right connections!

Introducing . . .

PATRICIA SEYBOLD'S NETWORK MONITOR

The Monthly Research Service that helps vendors and end users make better-informed networking decisions!

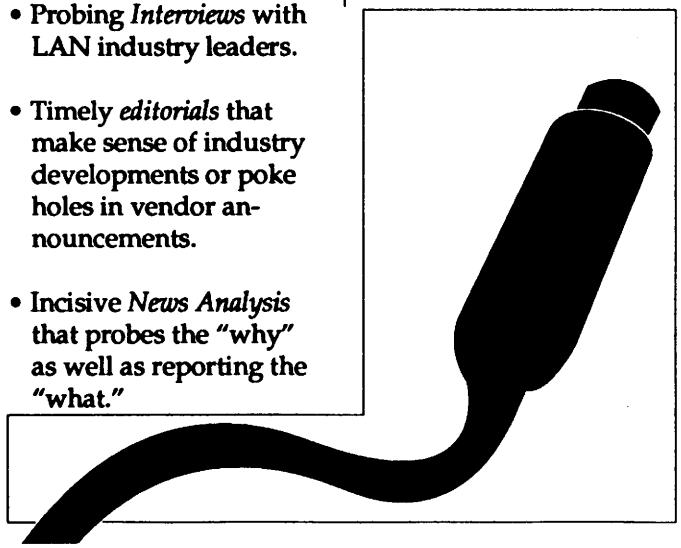
NETWORK MONITOR, a monthly research service, provides comprehensive analysis and objective evaluation of LAN products and vendor strategies. Edited by David L. Terrie, a leading LAN analyst and consultant, the newsletter assesses product strengths and weaknesses in terms of on-the-job performance and genuine user concerns.

Each Issue of NETWORK MONITOR treats its readers to a wealth of thorough research and commentary, including:

- *Feature Articles* offering analysis of leading vendors and products in a straight-talking, no-holds-barred style.
- Probing *Interviews* with LAN industry leaders.
- Timely *editorials* that make sense of industry developments or poke holes in vendor announcements.
- Incisive *News Analysis* that probes the "why" as well as reporting the "what."

Feature articles present broad-based product and strategy analyses of key industry players like IBM, Digital, Hewlett-Packard, 3Com, Novell, Banyan, AT&T, Contel, TRW, Wang, Ungermann-Bass, Sytek, Bridge, Apple, and other large and small companies on the cutting edge of LAN technology.

The analysis of a single issue might help you avoid wasting valuable time in a disastrous pilot program . . . let you get productive equipment on line *sooner* by narrowing your choices and shortening your buying cycle . . . make you aware of a market opportunity you could cash in on . . . or help you spot customer problems before they reach the "critical" stage.

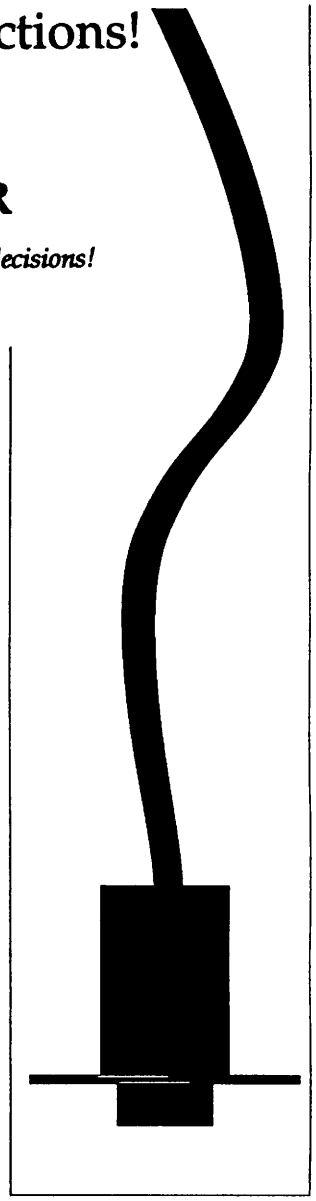


Whether you're a vendor or an end user, it pays to be well-connected . . .

GET ACQUAINTED WITH TODAY'S #1 LAN RESEARCH SERVICE

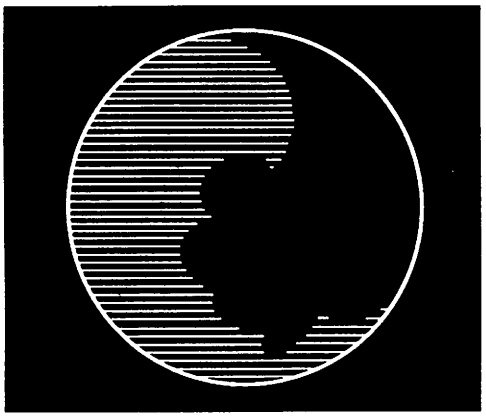
For the vendors, NETWORK MONITOR provides *monthly market research*: comparing your products and services against the competitions . . . assessing LAN-related market opportunities . . . charting industry trends . . . reporting on user concerns and expectations . . . covering topical issues . . . and analyzing what new developments could mean to you.

For the end users, NETWORK MONITOR provides *monthly consumer research*: identifying the best LAN hardware and software for your needs . . . protecting your investments with connectivity and integration guidance . . . presenting case studies that show how to proceed as well as what to avoid . . . reporting on vendor performance in areas like training, installation, service, and support.



Money Back Guarantee

What's more, each subscriber is offered a full guarantee. If you choose to discontinue your subscription at any time, you may do so simply by writing us. We'll send you a refund for all unserved issues. So send us your risk-free order today. Simply fill out the order form on the back cover of this issue. We'll send you your first issue immediately.



A N N O U N C I N G

The Sixth Annual **Seybold Executive Forum:** ***TEAMWORK, TECHNOLOGY, & ORGANIZATIONAL PERFORMANCE***

October 25–27, 1988
Royal Sonesta Hotel, Cambridge, MA

Forum Themes:

Moving Towards Real-Time Organizations

Teamwork & Technology: Making it Pay

Strategic Uses of CSCW

Information Systems as Change Agents

Please return form below to:

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

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

☐ My purchase order number: _____

☐ Charge to my MasterCard/VISA (circle one)

_____
Card Number Expiration Date

_____
Signature

_____
Name

_____
Title

_____
Company

_____
Address

_____
City, State, Zip

( )
_____
Telephone Number

As we look to the future, questions arise:

How will our organizations be structured?
How will we manage technology?
How will we manage our businesses?

As technology has evolved, organizations have had to evolve to meet the challenge. Now, technology is no longer a cause for change, but rather a tool to facilitate change. This year's Forum will focus on the urgency of evolving our organizations in cooperation with technology.

So join us as we ...

- Examine the impact of real-time systems
- Explore the relationship between technology and organizational performance
- Experience the catalyzing role technology can play
- Demonstrate how technology can be used to support teamwork
- Learn basic principles, practices, and techniques that sustain group cooperation and synergy

This year's Forum will include presentations by teams of senior executives from the foremost computer companies, such as IBM, Apple, Sun, and Digital, as they discuss their new corporate affiliations and articulate their visions for the future. These sessions will be followed by interactive Q&A led by the Office Computing Group consulting team. This highly-successful format has be acclaimed by past Forum attendees.

In addition, this year we will offer hands-on workshops on the Teamwork & Technology theme run by groups such as Action Technologies/ LogoNet, Digital Equipment Corporation, and Wilson Learning.

Cancellation Policy:

Should a registrant be unable to attend the Forum, the Forum Office will refund the full registration if notified before September 27. Cancellations from Sept. 8 to Oct. 11 are subject to a $50 service charge. There will be no refunds as of Oct. 12. Substitutions may be made at any time.

NEWS ANALYSIS

PRODUCTS • TRENDS • ISSUES • ANALYSIS

• INSIDE •

•RISC•

RISCy Business

Hewlett-Packard (HP) made a big splash in April with its latest implementations of its Spectrum Reduced Instruction Set Computer (RISC) platforms. HP has gone through the learning curve in production, etc., and is beginning to pump out a family of advanced products based on more sophisticated processor technology. Same RISC, just implemented with advanced technology rather than off-the-shelf stuff.

HP has done impressive work. The new Spectrum systems are small, and they are field upgradeable with a board swap. Cost of ownership is ridiculously low, and, combined with the low cost of the systems themselves, make HP a very attractive supplier of either Unix or MPE boxes.

HP isn't the only one hard at work in the RISC mines, however.

Motorola has finally unveiled its 88000 RISC-based processor, as well as a set of strategic alliances designed around the technology. Already, more than 20 vendors have committed to using the new chip. (Northern Telecom, for example, will be using the 88000 in the evolution of its DMS SuperNode switch, enabling better support for advanced and programmable services.)

The 88000, which is rated at between 14 to 17 MIPS, includes the 88100 and 88200 cache and memory units. Motorola also introduced "Hypermodules"—a series of 88000 multiprocessor boards that provide up to 50 MIPS aggregate processing power.

Motorola thus becomes a contender with Sun for providing a "standard" RISC chip for the marketplace. Vendors such as IBM and HP use their own RISC technologies in their internal products.

88open Consortium. The 88open Consortium is a group of hardware and software vendors pledging to promote the use of the 88000 chip. (Several weeks before the announcement of the 88000 and the consortium, Mips Computer Systems and its customers announced a similar body, designed to attract software developers for Mips' RISC platform.)

STRATEGIC ALLIANCE. Data General (DG) will be working with Motorola on creating an Emitter Coupled Logic (ECL) version of the 88000 chip, which DG will then use in a family of Unix systems targeted for next year.

Data General projects that the ECL chip set, when released in 1991, will provide about 100 MIPS of raw power. DG claims that the Scalable Processor Architecture (SPARC) chip, while adequate for workstation implementations, just didn't have what it takes to build a large system. (Sun has already addressed one earlier inadequacy of SPARC—its lack of internal memory management.)

COMPETITION. CCI is planning its own release of a RISC chip for use in its new Power 7/64 computers. CCI claims that its chip will pump out 25 MIPS sustained and 40 MIPS peak performance.

CCI claims that its chip will be able to run code compiled for Sun's SPARC architecture and will be source code compatible with the existing Power 6/32 systems that are resold by Unisys. (Unisys, by the way, has signed on with Sun for SPARC.)

COMMENTS. Thought of only a few years ago as a "far-out" technology destined only for a few specialized platforms, RISC has suddenly become everyone's darling. The technology makes sense. Much of the resulting marketing noise does not.

After the dust settles, we will probably end up with some of the larger proprietary vendors (HP, IBM) using their own RISC technologies internally, with two dominant "open" standards for the marketplace (Sun

SPARC and the 88000), and with a variety of smaller players scrabbling to maintain market share.

Differences from today's market will be that some of the larger vendors (DG) will also opt for standard chip technology, and, at least in this arena of the market, that Intel is not a player. Yet. ◐ —*M. Millikin*

•AT&T•

Goodbye, Vittorio—Hello, Bob

When Vittorio Cassoni took over the reins of AT&T's Data Systems Division, it appeared that, for the first time, a visionary was heading up AT&T's troubled computer division. And, indeed, Cassoni's leadership began to make a difference. He had begun to unveil a cohesive strategy that could lead to success in the marketplace. This has all changed now that Cassoni has announced that he plans to return to Olivetti as general manager. While this may be good news for Olivetti, it is disturbing for AT&T.

AT&T is in a tenuous position. It seems to have a knack for clashing with its most important allies: Sun Microsystems and Olivetti. Ironically, AT&T owns 20 percent of both companies. Strained relationships with its partners come down to a conflict of styles. AT&T has operated as a monopoly for so many years that its move into the fiercely competitive computer industry has been traumatic. For example, if you want to succeed as a manager in the AT&T culture, you'd do better starting out by stringing telephone cable. Also, old guard management still sees little difference between selling computers and selling telephones. So it is not surprising that Sun's freewheeling organizational environment would seem like an alien (as in outer space) culture to AT&T.

Olivetti presented another culture that AT&T was not quite in synch with. The European versus American orientation alone would have been enough to put some dampers on the relationship. More importantly, however, AT&T and Olivetti have very different ideas about what Unix should look like. For example, while AT&T's strategy is based on its System V, Olivetti bases its Unix strategy upon Berkeley Software Distribution (BSD) 4.3.

The bottom line, then, is that AT&T's future success in this volatile Unix market is not assured. With Cassoni's departure, the persona is in danger of again becoming a faceless monolith. What can AT&T do? First, it needs some bold leadership. If Robert Kavner, the new Data Systems Division president (see editorial), can provide this leadership, it will help the company immensely. AT&T needs to shake up its organization to make the necessary corporate changes. It needs to find ways to appreciate the strengths of its partners and to make peace with the rest of the Unix community. Vendors, such as Hewlett-Packard, Digital, and Apollo, are important to the future viability of Unix. These vendors perceive that AT&T intends to make Unix into a proprietary operating system that is closely tied to hardware and a proprietary interface. They also anticipate that they will not be included in future decisions about the functions and features included in new versions of Unix. These vendors believe that Sun Microsystems, as AT&T's partner, will have a competitive advantage. Ironically, AT&T is counting on Sun to provide the innovation and new features for the next generation of Unix products. AT&T needs Sun's technology. Therefore, AT&T must find a way to maintain a good relationship with Sun, on one hand, while keeping the rest of the Unix community happy on the other. It will not be an easy position. In the long run, AT&T needs to demonstrate that even though it controls this important operating system, Unix will remain an open standard. We expect that the next year could be a turning point for AT&T and, perhaps, Unix in general. ◐

—*J. Hurwitz*

•DATABASES•

Stratus Goes with Sybase

Stratus Computers Incorporated (Marlboro, Massachusetts) is positioning its recent introduction of SQL/2000, a relational database management system (RDBMS), as a platform from which to broaden its appeal in a multivendor networked environment. This announcement is also another strong vote of confidence in the Sybase RDBMS and its ability to effectively handle online transaction processing (OLTP) applications.

SQL/2000 is based on the Sybase DataServer from Sybase Incorporated (Berkeley). Both the Sybase DataServer and the Stratus operating system (VOS) have been tailored to optimize the performance and capabilities of SQL/2000. In addition, Stratus has extended the Sybase product to support its native VOS files.

SQL/2000 runs on the Stratus family of XA2000 Continuous Processing Systems. Of the eight different models, the first six were announced early last year, and the latest two high-end additions appeared this past February. Both the hardware and operating system (VOS) have been designed from the ground up to meet OLTP performance and availability requirements, with multiprocessors, fault tolerance and distributed networking capabilities.

SQL/2000, which has been in development for 18 months, is a good fit for both Stratus and Sybase. The Sybase RDBMS was also designed expressly for OLTP applications, and provides Stratus with a highly available database product on top of a continuously available hardware platform. OLTP is a fundamental requirement for Stratus customers, and the company

expects a majority of its customers to move to SQL/2000 over time. Stratus has packaged SQL/2000 attractively—the VOS Server (which allows SQL/2000 to access existing VOS files) is included with the operating system. Stratus also expects aggressive third-party support for SQL/2000. And finally, SQL/2000 neatly supports a new workstation strategy concurrently announced by Stratus (see "Client/Server Architecture" below).

THE SYBASE PERSPECTIVE. Sybase has always maintained that it would port its DataServer to only a limited number of selected platforms. In addition to Stratus, the DataServer runs on Digital VAX/VMS, Sun/Unix, and Pyramid/OSx computers. A version has also been announced for OS/2, the SQL Server, to be jointly marketed by Sybase, Microsoft, and Ashton-Tate.

Availability on Stratus now gives Sybase a chance to perform on an OLTP-focused platform with multiprocessors and built-in fault tolerance. (SQL/2000 is based on Version 3.0 of Sybase. Sybase Version 3.1, coming soon, includes fault tolerance in the form of disk-mirroring, a feature that is already standard in the Stratus environment.) We expect that Stratus will be the first platform on which Sybase will be enhanced to support multiprocessors.

THE COMPETITION. Stratus considers itself well-positioned to compete successfully with its main competitor, Tandem Computers Incorporated. Tandem also focuses on OLTP with its VLX family of fault tolerant multiprocessor systems and its Non-Stop SQL DBMS. Stratus claims that SQL/2000 offers superior features to Non-Stop SQL by providing extensive workstation support, database integrity in the data dictionary, the ability to perform system maintenance online, and integration with operating system files.

COMPONENTS. SQL/2000 is fully compatible with Sybase, and includes all of the Sybase back-end server components plus access to precompilers and VOS files. Availability dates and entry-level prices are also indicated below (pricing will vary depending on system size). SQL/2000 will be marketed exclusively by Stratus and its resellers (e.g., IBM and Olivetti).

Available in May:

- The SQL Server (the Sybase DataServer), which also includes DB-Library, a Sybase standard application language interface to the SQL Server (starts at $14,000).
- SQL/2000 DataWorkbench, including SQL, VQL (a visual query language allowing the user to compose and run complex SQL queries without any knowledge of SQL syntax), a visual report writer, and database administration utilities. The visual query and report writing tools are very important for ease of use and end-user applications development (starts at $12,500).
- SQL/2000 VOS Server for access to VOS files (included in VOS Release 7.0 at no extra charge to customers).

Available in the fourth quarter, 1988:

- Open Client Connection, software which provides the SQL Server with a network interface to Sybase-compatible requester (client or front-end) applications running on Sun workstations, IBM PS/2s, and other systems over Ethernet networks (starts at $7,000)
- Precompilers for embedding SQL statements in Cobol, PL/1, and C programs (start at $3,000 each)
- APT-Forms and Forms-Library, tools for creating applications (starting prices for these are $5,500 and $3,000 respectively)

CLIENT/SERVER ARCHITECTURE. Compatibility with Sybase means that SQL/2000 can be accessed from any workstation running the Sybase front-end DataToolset. Since Stratus itself is "workstation neutral" (the company only offers nonintelligent workstations), Stratus has chosen to extend this compatibility by announcing plans for a Stratus Workstation Architecture. This architecture will support sharing of information and services between its OLTP computer systems and PCs/workstations from IBM, Sun Microsystems, Apollo Computer, and Apple. Reading between the lines, we would also expect from this announcement to see Sybase offer at least its DB-Library module, if not the full DataToolset, on Apollo workstations and Apple's Macintosh in the not-too-distant future.

According to Stratus, specific products will be introduced over the next 18 months, and will encompass additional support for Transmission Control Protocol/Internet Protocol (TCP/IP); interfaces to Sun's Network File System (NFS) and Apollo's Network Computing System (NCS); and implementations of IBM's Token-Ring, LU6.2, NETBIOS, and MS-Net networking.

Workstation services will include sharing of Stratus-based resources (files, printers, disks), access to SQL/2000 databases, peer-to-peer networking, and presentation services.

Following the industry trends, Stratus is stressing adherence to industry standards and a client/server approach to allow its customers to effectively tie together both online business systems and advanced workstation functionality. ◐ —*J. Davis*

Ontologic Incorporated

Object-oriented data models are beginning to attract significant attention in the industry. While research and development efforts in object-oriented programming languages and database management systems (DBMSs) have been underway for some time, we are

just now starting to see commercial products that use objects as the underlying building blocks.

Last month, Ontologic Incorporated (Billerica, Massachusetts) introduced Vbase, one of the first object-oriented database systems to appear on the market. (Two other such systems are available—Gemstone from Servio Logic Corporation and Gbase from Graphael). Object technology is designed to manage and model large volumes of highly complex data and has long been of interest to organizations involved in applications such as computer-aided design and engineering (CAD/CAE), computer-integrated manufacturing (CIM), document management systems, computer-assisted software engineering (CASE), and knowledge-based systems. Object-oriented systems can also improve the software development process by increasing developer productivity and reducing the cost of maintaining applications.

SOME BACKGROUND. Several disciplines are gradually moving toward objects: database management systems, programming languages (e.g., Ada and C++), artificial intelligence languages/engines, and even user interfaces (witness the Apple Macintosh and Hewlett-Packard's NewWave—although we are loath to mention them both in the same sentence, what with Apple's suit and all). An icon-oriented graphics user interface is designed to present familiar objects (files, disks, trash cans, applications) for manipulation by the user.

BENEFITS. Why objects? What is so special about this approach? A major benefit is that it provides a simple data model that closely mirrors the world as the user actually sees it. This is sometimes referred to as "natural modeling." Most of us don't view the world as a set of two-dimensional tables, which is the model presented by the relational DBMS (RDBMS). While the RDBMS allows the developer/user the flexibility to determine the relationships between tables of data, the relationships still must be defined at the application level.

In an object-oriented system, the developer defines and manipulates the natural objects present in an application, rather than trying to fit the application into a predetermined set of DBMS concepts. The object-oriented database includes both data (properties) and code (behavior) in the definition of an object. Properties include two types of data—attributes (the familiar fields or columns of data in the RDBMS) and relationships. Code includes operations and constraints that apply to an object. Constraints can be used to enforce data integrity within the database. Thus, an object is an entity that encapsulates both data and procedures.

For example, the definition of a person-object type could contain:

- Attributes such as name, address, and age.
- Relationships such as spouse, children, or "member of" something, such as a project team. You could also relate a person object to a picture of that person (e.g., a bit-mapped image object that has been scanned into the system).
- Operations such as "marry" or "take on a project."
- Constraints or rules.

In addition, in an object system, if you specify that an inverse relationship exists (e.g., the children relationship is the inverse of the parent relationship), the system will maintain referential integrity.

Unlimited Data Types. A second major advantage of the object-oriented approach is the ability to define your own objects, i.e., to create user-defined objects or data types. Most RDBMSs are limited to the specific data types they support—integer, floating point numeric, character, date, etc. Even those that now support what are sometimes called "blobs" (basic large objects) may not incorporate the ability to differentiate among several types of blobs (e.g., bit-mapped image versus structured text versus graphical data), or to define special operations to manipulate such objects. An example here would be to "rotate" an engineering drawing.

Inheritance. Another advantage of object-oriented systems is the notion of inheritance. As Ontologic points out, "People attack new problems by seeing similarities and differences with old problems. Often, new solutions can be produced by making slight modifications to old solutions; and similar solutions to several different problems can be generalized to a 'generic' solution."

Once an object is defined, new types of objects can be defined that take on (inherit) certain specifications, relationships, and operations of the original object, and that add on their own differentiating characteristics. Thus, the application developer can reuse code, both within and among applications, to improve productivity. As an example, a part is an object type that has a part number and a supplier. From here, the developer could define "subtypes" of a part, such as pipe and valve. Each of these takes on the characteristics of a part, but has additional data that differentiates it. A pipe has a diameter and a length; a valve might have different specifications.

Navigation. Ontologic also describes the ease of navigating through an object-oriented database like Vbase. For example, to describe Jim's grandfather's age (assuming the necessary data is in the database) might simply entail the following "dot" syntax in an object-oriented database: jim.father.father.age. Each dot represents a connection in the database. To do this with SQL would require a fairly complex nested query.

ONTOLOGIC. Ontologic Incorporated was founded in 1983 as Mosaic Technologies, a developer of hardware and software for graphics workstations. In 1985, the company changed its name to Ontologic and decided to focus exclusively on its object-oriented database

software. Since then, the company has received $10.5 million in venture capital funding.

Ontologic has built Vbase from the ground up as an object database system with a C language interface. The Vbase system environment consists of the following components:

- Vbase database. Objects are stored in the Vbase database.

- Object language. Vbase has two languages. The Type Definition Language (TDL), used to define object types in the database, is based on the object model to define properties, operations, and inheritance. TDL is similar to data definition languages in other DBMSs. The C Object Processor (COP), an object extension of Kernighan and Ritchie standard C, is used to implement the operations of the object types defined using TDL.

- System Type Library. Vbase includes a library of over 60 object types that have already been defined for the developer. These include object types such as integer, character, date, string, set, array, dictionary, stack, and graphics support types.

- Object SQL. Vbase's query facility provides the retrieval features of SQL with extensions relevant to an object database system.

Ontologic believes that Vbase has a leg up on its current competition (Servio Logic and Graphael) by focusing on production applications (versus prototyping), performance, and the need to be compatible with already existing database applications. Vbase provides an interface to the C language, a more mainstream development environment than that provided by either Gemstone (SmallTalk) or Gbase (Lisp). Ontologic also stresses its strong SQL bent as well as the means it provides to eventually tie into existing applications and SQL-based DBMSs.

Ontologic also states that it expects to enjoy a two- to three-year head start (especially in the area of performance) over the traditional RDBMS vendors like Oracle and Relational Technology. The company feels that it will take that long for these competitors to fully implement object functionality. And those that take a layered approach (extending their relational product to also handle objects rather than building an object system from the ground up) will get there faster, but pay a penalty in lower performance.

In terms of marketing, Ontologic is looking for "high-performance C applications that need database functionality." Other characteristics of appropriate applications include very complex data and a need to improve and enhance developer productivity. Many customers are CAD/CAE companies who have had to design their own customized database systems to handle these requirements in the past.

COST AND AVAILABILITY. Release 1.0 of Vbase for Sun Microsystems workstations was delivered in February, and Vbase for Digital's VAX/VMS will ship this month. Cost depends on the hardware installation, ranging from $10,000 on a workstation to $40,000 on a minicomputer and up to $80,000 on a large minicomputer. Ontologic sells Vbase directly to end users and applications developers. Sales offices are located in Boston and Santa Barbara. In Europe, Vbase is available through distributors.

NOT YET READY FOR PRIME TIME. It is still early in the game for object-oriented tools, particularly for mainstream commercial applications. However, the signs are there that the evolution will continue as object systems prove capable of solving some complex data management problems. These systems are also attractive because they require less abstraction for the user in mapping the application to the database management system, and less code for the developer to write.

The major drawback for widespread use of Vbase is the lack of easy-to-use tools for creating and modifying applications; the developer must use the C language to make effective use of Vbase. Currently, Vbase does not support networking; all processes must be resident on a single machine. The company is now developing networking support for Vbase on homogeneous machines. Future steps will include heterogeneous networking of Vbase databases, and then access to non-Vbase databases. ◐ —*J. Davis*

Ashton-Tate Joins the Fray

With interest in database management systems (DBMSs) growing by leaps and bounds, the industry has been waiting to see what Ashton-Tate, the leader in the PC environment with dBase III Plus, would decide to do. Well, over the past few months, the company has finally laid some cards on the table. Users are now eagerly awaiting dBase IV, scheduled to ship this summer, and the SQL Server, to be jointly marketed with Microsoft and Sybase in the OS/2 marketplace.

The latest play for Ashton-Tate is the company's recently announced agreement to jointly develop workstation database products with Interbase Software Corporation (Burlington, Massachusetts). This technology agreement with Interbase will lead to development of Ashton-Tate's "next generation" of workstation products, and it clarifies several major directions for Ashton-Tate.

WORKSTATION PRODUCTS. It is clear that Ashton-Tate, at least for now, will concentrate on developing workstation-based products (the client, or front-end side, of the client/server relationship). The company obviously realizes that the client/server architecture for database products is critical to success; the first step here was the SQL Server (based on the Sybase DataServer), a solution that allowed Ashton-Tate to get into the game quickly. The company will market the SQL server in

the retail market and is adapting dBase to provide the client interface to the SQL Server.

What is not yet clear is whether the agreement with Interbase will also eventually result in the development of a back-end server for Ashton-Tate, or whether Ashton-Tate will be content to stick with the client side and provide interfaces to other vendor's servers.

New Platforms. Ashton-Tate is already moving beyond its traditional DOS environment with products for OS/2. The work with Interbase will focus on the OS/2 Presentation Manager as well as other operating environments such as Unix, Apple's Macintosh, and Digital's VAX/VMS.

Graphics User Interface. Future products will be graphics-based to take advantage of the coming trend in all major computing environments to implement a Mac-like windowing interface.

SQL. A major requirement for a DBMS today is SQL compatibility, and Ashton-Tate will have to implement this to become a successful player on multiple platforms.

Distributed DBMS Capabilities. As organizations look to integrate PCs and workstations into existing computing networks, distributed database capabilities and heterogeneous connectivity become more and more important.

WHO IS INTERBASE? James A. Starkey, president of Interbase, founded the company in 1984 with the objective of developing advanced database management software. Starkey brought with him 10 years of experience at Digital, where he played a primary role in designing Datatrieve and created the Digital Standard Relational Interface and Rdb/ELN.

The company's product, InterBase, is a relational distributed database system developed for the technical engineering and scientific workstation environment. It currently runs on workstations from Apollo (under Aegis and Domain/IX) and Sun Microsystems (under Unix) as well as on Digital's VAX (VMS and Ultrix). The product is designed to serve database environments characterized by a high degree of data-sharing among multivendor networks, complex data such as CAD images or digitized voice, and large volumes of accessible data. Version 2 of InterBase was introduced last October. Major features include:

- Support for "blobs" (basic large objects) such as graphics images, executable code, and text. (Coming is support for voice and extensibility for user-defined data types.)
- Multi-generational data structures to manage concurrency and ensure data integrity.
- The ability to store in the data dictionary both triggers (to specify inter-record constraints, such as those used to ensure referential integrity) and validity conditions (to specify field constraints).
- A two-phase commit protocol for updating multiple databases in one transaction.
- The ability to back up files while other users are using the database.
- A fourth-generation language (Query Language Interface, or QLI) that allows the developer to use either industry-standard SQL or InterBase's own proprietary language (GDML), or a combination, to access the database. In addition, products that conform to the Digital Standard Relational Interface (DSRI) can also directly access Interbase databases. This includes Digital's Datatrieve and applications from other third-party vendors. Version 2 also supports dynamic SQL, enabling users to build the content of a query at runtime.

Version 2 of InterBase ranges in price from $5,000 to $75,000, depending on the hardware configuration. It supports the following communications protocols: DECnet, Apollo's Domain, and Transmission Control Protocol/Internet Protocol (TCP/IP) for Unix systems. The company currently has 17 employees.

WHY INTERBASE? One of the first questions we had when we found out about the agreement with Interbase was: Why not Sybase, since Ashton-Tate had already joined forces with Sybase in marketing the SQL Server and is adapting dBase to work with Sybase? As explained by Ashton-Tate, the company wanted a development partner that would not only provide the appropriate technology, but would also permit a very tight, long-term working relationship (and probably a higher level of control than would have been possible with Sybase, a much larger company than Interbase). Ashton-Tate described its investment in Interbase as "significant," and the president of Ashton-Tate now sits on the Interbase board of directors.

While Ashton-Tate is firmly committed to the SQL Server, as evidenced by its recent participation in equity financing for Sybase, the company stresses the SQL Server as primarily designed for the OS/2 platform. The agreement with Interbase, on the other hand, will focus on developing workstation products that will interface with multiple servers on multiple platforms, including the SQL Server. These workstation products will maintain full compatibility with dBase IV as well as with all current and future Interbase products.

Another factor influencing the decision to go with Interbase was Ashton-Tate's desire to provide a workstation product that would not only work with servers in a client/server environment, but would also permit the user to create and work with local databases stored at the workstation. Compact code is required to do this and still allow other applications to run concurrently on the workstation. Interbase has had extensive experience in developing stream-

lined code, since its existing products are targeted at this workstation market.

The benefits for Interbase, aside from the obvious financial ones, include the opportunity to collaborate with a major PC software company, and access to the PC environment and the huge installed base of dBase users.

Ashton-Tate will have exclusive marketing rights to the jointly developed technology, which it will co-own with Interbase.

Ashton-Tate is entering the race to provide the front end, workstation tools of choice in a multivendor, distributed DBMS environment. The buy-in with Interbase will provide technology in areas critical to Ashton-Tate's success, including SQL, distributed databases, multivendor networking, access to heterogeneous databases, and manipulation of complex data types.

—J. Davis

Patricia Seybold's Computer Industry Reports

Order Form

Subscription rates:		U.S.A.	Canada	Foreign
Patricia Seybold's Office Computing Report	12 issues per year	$385	$397	$409
Patricia Seybold's UNIX in the Office	12 issues per year	$495	$507	$519
Patricia Seybold's NETWORK MONITOR	12 issues per year	$595	$607	$619

SAMPLE ISSUES ARE AVAILABLE UPON REQUEST. **TO ORDER BY PHONE, CALL (617) 742-5200.**

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

Name: _____ Title: _____

Company Name: _____ Dept.: _____

Address: _____

City, State, Zip code: _____

Country: _____ Bus. Tel. No.: _____

Card #: _____

Exp. Date: _____

Signature: _____

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

Send to: Patricia Seybold's Office Computing Group: 148 State Street, Boston MA 02109; (617) 742-5200; FAX: 1-617-742-1028

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

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

Office Computing Report

1987—Volume 10

1 Jan. Predictions—Systems and Players in the 1990s *Other Features:* Using Technology to Competitive Advantage
2 Feb. Leading the Way—Will DEC stay on Its Roll? *Second Feature:* General, DEC, HP, IBM, Wang
3 Mar. Digital's All-In-1 *Second Feature:* Service and Support II: Honeywell, NBI, NCR, Prime, Xerox
4 Apr. LANs in the Real World: The Users Speak *Second Feature:* 3Com's 3 Station: A VAXmate Killer?
5 May Wang Tries a New Approach: Integrated Image Management *Second Feature:* Electronic Publishing
~~6 June Prime Information~~ **OUT OF STOCK**
7 July End-User Computing—Tools for Visionary Users
8 Aug. Will Wang Make it? Can Fred's Leadership Turn the Tide
9 Sep. Voice Messaging Systems—User Perspectives
10 Oct. IBM Office Software—Implementing A Cohesive Vision *Second Feature:* The Marriage of PC and FAX
11 Nov. User Developed Applications—They Know What They Need
12 Dec. Hewlett-Packard's New Wave

UNIX in the Office

1987—Volume 2

1 Jan. DOS/Unix Integration—Doing it Right with Multiplex *Second Feature:* It's...Super Boxes!—Disguised as Mild-Mannered Multiuser Systems
2 Feb. The Officesmith—The Keyword Is Structure *Second Feature:* Standardize or Bow Out
3 Mar. Unix Arrives!—And UniForum Is It's Coming Out Party
4 Apr. Apollo's Challenge: Bowing to Standards, Yet Remaining on Top
5 May Open Systems and Standards Fuel Sun's Fire *Second Feature:* A New Dimension: Plexus's Mixed-Mode Data Processing
6 June The Ingres Realtional DBMS—Riding the Wave
7 July Texas Instruments—Blending AI with Commercial Systems
8 Aug. Banyan Systems—A Unix Solution to the PC LAN Conundrum
9 Sep. AFCAC 251—Defining the Future Unix Office
10 Oct. The Oracle Relational DBMS—The Leader of the Pack
11 Nov. Expres Initiative—Is there an Office Use for University Communications
12 Dec. Unix Roundup—Unix-Based Integrated Office Systems

Network Monitor

1987—Volume 2

1 Jan. Wang's WangNet—A Product In Transition *Second Feature:* Broadband Technology Primer—Wang's Dual-Cable Approach
2 Feb. Sytek—Can It Make the Big Time? *Second Feature:* Fibronix Debuts FDDI LAN
3 Mar. Bridge Communications—Stalking the Competition
4 Apr. Networking the Macintosh: Apple Opens the Floodgates *Second Feature:* Hewlett-Packard's GlobalNetwork Strategy
5 May Integrated Voice and Data LANS: IEEE to Marry ISDN and LANS *Second Feature:* Fox Research's 10-NET: An Expatriate Returns
6 June Banyan Systems Incorporated—Networks for People Who Think Big
7 July Apollo's New Thrust—Shaping Computing's New Generation
8 Aug. Proteon Inc.—A Better Mousetrap Isn't Enough
~~9 Sept. Digital's DecNet~~ **OUT OF STOCK**
10 Oct. DataGeneral—Better Late Than Never
11 Nov. IBM and SNA—Meeting the Challenge?
12 Dec. Micom Interlan Inc.—Learning From Past Mistakes

Patricia Seybold's Office Computing Group: 148 State Street, Boston MA 02109; (617) 742-5200; FAX: 1-617-742-1028