OPEN SOFTWARE FOUNDATION: DELUSIONS OF CERTAINTY

When we first met Vittorio Cassoni he told us his major task as president of AT&T's Data Systems Group was "to absorb uncertainty" (Release 1.0, 87-2). Cassoni has returned to Olivetti now, but his talents are needed more than ever. The future of the coalescing UNIX standard (and its commercial promise) has been thrown back into a state of chaos and uncertainty by the launching of the Open Software Foundation last month.

Ostensibly created to counteract Sun and AT&T's "closings" of the UNIX "standard," OSF has proclaimed its intent to absorb uncertainty by creating a standard -- attempting to usurp the market's unpredictable role by fiat. But you can't reduce uncertainty by saying so, any more than you can make a doctor with a degree or guarantee love with a marriage certificate.

This group is attempting to turn a dynamic process -- the creation, evolution and succession of standards -- into something more fixed and controllable. In fact, standards are a fundamental contradiction in the scheme of things: They're static, brittle space, a vacuum in a changing, fluid world. They don't exist in nature, and they take on life with their death already pre-ordained because they're too rigid to evolve.

The creation of the Open Software Foundation has raised hackles and skepticism throughout the computer industry. OSF is putatively an independent non-profit company sponsored and organized by seven hardware companies for the purpose of building an open standard based on IBM's AIX, which is in turn based on AT&T's UNIX. UNIX itself, they argue, is becoming closed, or proprietary, based on changes in AT&T's license terms for UNIX and the recent alliance of Sun Microsystems and AT&T to create a binary UNIX standard. What really gives them the willies, of course, is AT&T's investment in Sun, plus Sun's sponsorship of the SPARC RISC architecture.

In other words, now that AT&T is trying to wrest control of UNIX from the marketplace, competitors and customers are afraid that they will succeed wildly, and ultimately turn UNIX...
proprietary. Competitors include both other UNIX vendors who truly want UNIX to be open and are afraid that they will lag behind Sun/AT&T, and vendors offering proprietary systems, who hope UNIX will either go away or become "proprietary" quickly. (If UNIX were really proprietary, it would not be such a threat; Apple could compete with IBM more easily without all the IBM clones.) Customers are afraid that Sun and AT&T will succeed and jack up prices once they have gained "control" of the market.

In response, the group of seven has launched a countermove to take control into its own hands. Although the Open Software Foundation will ultimately become independent, subject only to board-level control from its sponsoring members, culture and history frequently count as much as legal charters in setting corporate strategy.

OSF certainly impedes what looked like smooth sailing for AT&T and Sun, but it's more an acknowledgement of schisms in the UNIX world than the creator of such schisms. Until AT&T made its moves to unify UNIX by asserting control, there was no such thing as a UNIX standard. Instead of one coalescing standard led by two companies and a large group of nonaligned players likely to adopt that standard by default, the UNIX arena now has two opposing forces, one dependent on UNIX and the other adopting it in response to user pressure. Call it Sun White and the Seven Dwarfs. But the story is not yet over: Sun and AT&T may join OSF; OSF may come to some accommodation with System V.4 when the rhetoric dies down, or it may abandon UNIX™ entirely; AT&T may spin off its UNIX business into a new unit or to separate owners (with the same independence OSF has from its board?).

THE MOTIVES

None of this would have happened had customers (especially the government) not started specifying UNIX, in the mistaken belief that UNIX was an it -- a standard. For customers who have spent years dealing with incompatible systems, painful upgrades and isolated data, and have grown skeptical of the promises of their existing, proprietary-oriented vendors, the lure of a standard is strong -- and deceptive. For "Third-World" vendors, UNIX provides a way in, as it did when Sun realized it had neither the time nor the resources to create its own OS and hired UNIX whiz Bill Joy from Berkeley as the fourth member of its startup team (and as it may do for the Japanese?).

For the OSF members, established vendors all, UNIX is an accommodation to customers -- on top of which they can still hope to lay proprietary value-added in the form of superior software tools, interfaces, applications and service. The OSF sponsors are firms to whom UNIX represented more a threat than an opportunity. (Even Apollo, where UNIX systems have grown to 60 percent of sales since 1985, capitulated to UNIX only because of insufficient demand for its own technologically superb Domain/OS.) Yes, it's true that they're doing UNIX because the customers want it, but that doesn't mean much. They charge reasonable prices because the customers want that; it doesn't mean that they like to charge reasonable prices. As the members concede, they're not about to stop their own development efforts.

OSF lets them hedge their bets: Either we win, or they lose. If it works, we want to be there, and meanwhile perhaps we can slow AT&T/Sun's momentum. In this regard, consider a joke told in another context by IBM's George Conrades about a veterinarian/taxidermist: Either way you get your dog back.
Yet, since it doesn't change our argument about OSF's flaws, we're prepared to take seriously at least some of the group's claims of altruism. Or to take seriously the notion that at least some people within each of the sponsoring companies have only the users' best interests at heart. Big companies such as IBM and Hewlett-Packard are simply too big to be all of one mind. As one member told us, yes, the project did begin in a defensive mode, but it's moved on to take the offensive, to propose standards well beyond the operating system, things such as SQL and communications protocols (reminds you of IBM's OS/2 Extended Edition, doesn't it?). Of course, should Open Software Foundation approach any sort of success, it will soon be riven with the same divisions that are now bedeviling UNIX overall. How happy will H-P and DEC be to rely on IBM for a software base?

IBM in particular

IBM has always swung back and forth between central control and decentralized, market-minded internal competitiveness, with neither side ever winning fully. The last decade has seen a swing back to a coherent strategy, from the days when the company was almost torn apart by the Justice Department. In the face of DEC's monolithic architecture and wrong-headed public insistence on compatibility and standards at the expense of optimization and experimentation, IBM has extended its Systems Network Architecture to Systems Application Architecture, under the guidance of Jack Kuehler, and focused on three major platforms (PS/2-OS/2, AS/400, and 370-33XX). The opposing viewpoint -- let a hundred flowers bloom -- survives under CFO Allen Krowe, who was instrumental in building IBM's support for OSF.

Now that the deed has been done, discussion isn't over within IBM. Some people think OSF was a good way of stopping UNIX; others are nervous that IBM will pay more than lip service to the effort. And some, such as IBM VP and Fellow Andy Heller, are sincere about spreading the cause of UNIX. Heller, manager of the company's RT PC team, apparently thinks software should be free and wants to win on the strength of his hardware. (DEC, HP, Apollo, beware! You ain't seen nothin' yet!) The company that can field the 370 and the Silverlake-AS/400, to say nothing of the 8100, the RT PC, and the PC, can handle a little divergence within its ranks.

Meanwhile, back at the ranch

To be sure, AT&T's and Sun's motives aren't "pure" either. Both companies have a fiduciary duty to shareholders to maximize profits, subject to constraints of morality, etc. Our only problem with Sun and AT&T is that they have been less than forthright. It's obvious to all but the naive that Sun and AT&T can gain an advantage by being the ones to define a UNIX "standard." Otherwise they wouldn't bother. But they also understand clearly that they are more likely to be successful if that "standard" is widely adopted. Therefore, they're willing to license it to other folks at prices that will encourage those other folks to use it.

But they want the ability to control the product definition, and they figure they have the right to enjoy at least initial rewards for their efforts. So far so good. The concern is that AT&T will later jack up prices or even (as IBM is doing with the PC and PS/2) charge retroactively, and that it will favor its own hardware at the expense of others'. That's natural; the antidote is that if AT&T goes too far, no one will buy UNIX anymore.

Release 1.0

27 June 1988
A brief geological survey of standards, layers and openness

Suddenly the value of standards has become conventional wisdom. Customers want standards to minimize disruption in a world of increasing numbers of users who are expensive to (re)train and linked machines from multiple vendors. (Multi-vendor sites were not so complex when each system existed independently of the others.) Vendors want standards because customers want them. And, of course, vendors want to own standards because they can make more money that way -- whether by charging premium prices or earning royalties.

Some definitions

Standards and interfaces are intimately connected. Standards, because they can't handle everything at once, cover defined portions of a system -- either horizontal, as in OS, network, hardware layers, in which the various components of a single system are divided; or vertical, such as protocols that enable different systems at the same level to communicate. Standards set up regions of certainty within a changing environment. In fact, since changes are always occurring, most standards specify only the insulating interfaces between components, so that people can be free to change the internals of each component without worrying about external effects. The promise is: You can rely on this program interface; we may change the implementation details underneath, but we'll always support the external spec.

Interfaces are the fault lines between standards. They are temporary locuses of rigidity that are bound to shift abruptly and painfully at some point; their stability is illusory. Unseen pressures are building up; needs are satisfied only partially, as vendors and customers settle for the standard because it is convenient, because it is there, because change would be too disruptive. Standards are an accommodation -- the triumph of the definite adequate over the shifting, confusing forces of improvement.

The unbearable emptiness of openness

The word "open" means too many things. In the UNIX community it has traditionally meant open to all, costing almost nothing and subject to modification by all, with source code available so anyone can extend the system. That, however, is a recipe for chaos (and the recipe more or less espoused by the Open Systems Foundation).

Users' goal is not openness in the sense of "free," although they'd like that; it's flexibility and choice. Customers want roughly the same products available from multiple vendors; they want multiple sources keeping each other honest. They also want some assurance that an architecture will last, so that they won't be left high and dry. In other words, they want defined, permanent interfaces.

That first goal -- choice -- is realistic. The second -- certainty -- is chimerical. Some change is necessary for progress (although perhaps not as much as IBM has inflicted on its customers). Openness and standards may avoid the pain of change from vendor to vendor, but it will not obviate the pain of change due to fundamental improvements. The smaller
changes may be hidden by interfaces, but long-run, when the earthquake of improvements finally surfaces, it will be monumental. (What did the earthquake June 12 mean?)

How standards get created

The standards lifecycle works best as follows: A new product is created, typically by a fanatic who doesn't listen to his managers or to his customers, and who wants to overturn the current standard. This first step happens hundreds of times; one in a hundred times, the fanatic creates something better than the current standard by an order of magnitude. Then, with some luck, this ten-times-better mousetrap attracts a committed marketing team and financial backing (sometimes from an existing company, such as Apple's for the Macintosh, but more frequently from outside sources with less vested interest in the old, as in the case of Tandem or of Apple itself for the Apple II). It takes a lot of effort to get this far, with natural selection weeding things out along the way. Natural selection is harsh, and it may cause good ideas to perish for senseless reasons. Perhaps the marketing effort was poor; perhaps the developer had a drug problem; perhaps another product almost as good hit the market earlier.

Finally, one product wins out and enjoys a brief day in the sun. Perhaps it's good enough and wins enough support to last a couple of generations. One way or another, it becomes a "standard." Other vendors support it, clone it, extend it. But eventually someone in a garage somewhere supersedes it, perhaps with a totally different system, perhaps with a system that builds on the old standard, and the cycle begins again. Since the creator of the original standard usually spends a lot of effort (emotional, political and financial) to maintain it, the new standard is likely to come from a new source.

Typically, standards are created and invested in by a single vendor. (We do not consider the current array of UNICES a standard.) At some point they are widely disseminated, through licensing or imitation, and typically investment is spent on distribution rather than the product itself. Sometimes vendors throw products into the public domain -- as Xerox did with Ethernet. It's at this point, we'd argue, that one could make a case for abrogating rights of ownership, so that vendors would lose a vested interest in the past (beyond avoiding the cost of change), and investment could focus on the next generation. In fact, the market usually takes care of that by reducing the differentiating value of the original standard.

Ultimately, standards are set by customers who buy things. Customers are influenced by many factors: product quality; pricing; value-added items such as third-party products, support and similarity to existing products; value-subtracted such as a need for recompiling or file conversion, software modification and user training; identities of the vendors; marketing efforts (including those of "independent" bodies such as the Open Software Foundation); de facto government endorsements such as RFPs requiring UNIX; perceptions of products as to quality, openness and other characteristics; and even the opinions of journalists. Combined, all these constitute "the market."

Release 1.0 27 June 1988
Sun and AT&T might have won more support and perhaps staved off the Open Software Foundation had they been willing to make some formal representation about the future openness of UNIX, but we doubt the lawyers could have gotten around to it in time even had they wanted to. "In retrospect," says one AT&Ter, "we probably let it fester too long between the announcement [of the relationship with Sun on October 19] and Uniforum [February], where we thought we'd have a good chance to lay it all out and explain everything."

The Hamilton Group, a precursor to OSF, first met in January. In fact, probably the best thing both Sun and AT&T could do now would be to join the Foundation as sponsors and then continue to conduct business as usual, which is what its other sponsors will probably do, albeit $13.5 million poorer. So far, AT&T has refused to join, while Sun is considering the issue.

Certainly, AT&T has done a poor job communicating the terms of the UNIX license agreement, widely considered to be arbitrary and unfair. In fact, the only major change we know of is that as of Release 3 (1986), resellers must agree to support the SVID (System V Interface Definition). The license is still perpetual, although subject to price changes; resellers need a new license for a new release; and pricing is subject to change with 90 days' notice. None of these latter conditions is new.

But the real issue is hardware: AT&T will be producing (with Sun) hardware-specific versions of UNIX for the SPARC, the 386 and for its own 3B (as well as for Motorola's 8800, Intergraph's Clipper and the MIPS chip). For customers, the advantage is the ability to buy and run shrink-wrap software out of the box on a variety of machines. For other vendors the catch is, Yes, they can freely build or buy the SPARC for their own machines and offer the same benefit, but forsaking a proprietary operating system is bad enough; they don't want to switch to "open" hardware too. (We understand the group was originally the Open Systems Foundation, but thought better of that.)

Innocent bystanders

None of the seven sponsoring companies sell software to run on other people's hardware, but rather sell systems software as part of their hardware line. Because all the software vendors can sell is software, and they don't like the idea of giving it away, even for PR purposes. Nor, despite the sponsors' insistence on the venture's international flavor, does OSF yet include any Japanese companies. Microsoft, of course, would rather see IBM focus wholeheartedly on OS/2 at the low end, and couldn't quite bring itself to follow IBM's lead.

The Third World of computing -- which now sees itself as a Fourth World behind IBM, DEC, and Sun/AT&T -- is mostly non-aligned, or perhaps paying protection to OSF at $25,000 per year and paying license fees to AT&T. (What world OSF sits in we don't know.) Most of the vendors seem to share John Sculley's attitude: "We'll watch it just like [we'd watch any] tennis match." Another vendor told us, "We'll go with the winner." Those are the sounds of the market working.

Up and down the line

Although this whole fracas is depicted as a battle between Sun/AT&T on the one hand and the OSF gang on the other, the market is not so monolithic. For the moment Sun is going after the desktop, with its ABI and Open Look,
while most of the OSF vendors (Apollo excepted) are going after the server market (what used to be called minis). With its commitment to OS/2, IBM for one will be hard put to support a desktop OSF system vigorously; the others don't have the market strength at the low end to put up a convincing fight. Will a presence on the desktop (if they achieve one) help AT&T/Sun win elsewhere? Probably, since it's a visible position in a mostly muddy market.

THE MECHANICS

For the moment, IBM will hand over the specs and code of future-AIX sometime this summer, and then a number of teams will get to work embellishing it and turning it into a hardware-independent source code reference system (with at least one internal hardware implementation just for testing). IBM's release of future-AIX should be out by late 1989; the other vendors should be close behind, with a richer product including the OSF enhancements. (That's the theory, anyway.) Part of AIX's appeal -- aside from any value its adoption may have had in persuading IBM to join OSF -- is its modularity, since it was written with portability in mind from the start. It runs across IBM's array of 370, RT and Intel architectures. That modularity will allow OSF to pick and choose the parts it wants, and incorporate other technology.

For example, OSF will select the user interface as follows: This summer it will request technical proposals for a standard interface, built on top of X Window and incorporating a toolkit and style guide. Yes, AT&T and Sun are invited to submit OpenLook; Microsoft and IBM could offer Presentation Manager -- a move fraught with meaning and an unlikely boost to OS/2's major competitor. Since H-P is already using a PM-based interface for the DOS version of NewWave, H-P seems the most likely proponent of PM. These offerings will be published to the membership, who will be invited to a technical selection conference for three to five days, not to vote but to provide detailed input on the decision. At this point, the OSF staff will make a decision; if they can't, it will be pushed up the OSF hierarchy, but, pledges director of development John Paul (on loan from Nixdorf), it will not go to the board of directors. He hopes to have the user interface decision by Thanksgiving. (Take note: Here's a tangible measure of progress.) Whatever interface decision OSF makes, however, it has also promised to follow X/Open's standard, which should also be decided shortly.

OSF will release all significant information on a monthly to quarterly basis, promises Paul. Inevitably, early access means that people will be dealing with unfinished code and specs subject to frequent changes.

Other major issues are still unclear. That's fine; we'd be nervous if the group had everything fixed from the start. But there are a lot of questions to which there may never be satisfactory answers (just ask the folks from MCC): How much will OSF build products and how much will it simply endorse and integrate other vendors' products? (How much stuff and how much glue? Will all the talent be wasted on the glue, while AT&T and Sun are working on System V.5 in Menlo Park?) What will the licensing terms be? Why should a successful vendor want to sell its product through OSF? Why is IBM donating AIX, for example? And for how long and how much of each product will these license agreements last? In fact, why does the OSF product/spec need to be relicensed from AT&T? (Future versions may not be, OSF said last week.) How close will the OSF UNIX be to the merged AT&T/Sun UNIX? Will they both be supersets of the reasonably complete V.4 merger of V.3 and BSD, with dif-
ferences mostly in higher-level areas such as file systems, communications, database and user interfaces? All these questions have yet to be resolved.

Hardware and interface issues

The OSF "product" will be a portable source code reference implementation; it won't support the shrink-wrap portability of the AT&T ABIs. The notion of hardware independence has general appeal and specific limitations. Vendors who rely on the OSF source code will end up at the other end of the traditional trade-off from AT&T and Sun. Users of one machine do not want to suffer for the limitations of other machines; rather, they want the best that the particular machine they're using can support. On the other hand, OSF-based products will be a generalization and homogenization of products available on specific machines, so they can easily absorb new hardware technology. Any machine-specific optimizations OSF members build will be proprietary -- albeit supporting general standards. Any enhancements they build may be proprietary, or may end up in the OSF spec.

Finally, we'd like to make one point in OSF's favor: its focus beyond the operating system. As discussed on page 13, software historically advances the line between chaos and regularity, turning more and more functions into standard processes that can be fixed and built into a system substrate, leaving developers free to concentrate on the next level of functionality. Over the past few years, the OS has subsumed memory management and file management, and is beginning to absorb communications, database management and user interface. Appropriately, the OSF companies are talking about setting standards in these areas as well. (What about groupware?) All these ideas are commendable, but the group must also decide where to locate its headquarters and hire some permanent staff.

Compare with DOS

Suppose IBM/Microsoft had decided not to build OS/2, and had instead freely licensed the DOS source code and said to everyone, "You do it"? There would be multiple versions of DOS approaching the functionality of OS/2, but each in its own way -- with unique approaches to memory management, multitasking, subroutine calls, pipes, file structures, window management, all called "DOS." We could lump together The Software Link's MOS/386, Graphic Communications'/Softguard's VM/386-plus-DOS, A.I. Architects' OS/386 DOS Extender, Wendin's Wendin-DOS, Rational Systems' DOS/16M, Concurrent CP/M, among others, and call them all DOS/2. (We could even throw in a couple of 68000-based ones for good measure.) Some of these systems might work quite well, but how much investment would they (did they) attract from application builders? Fundamentally, except for a different genesis and a smaller variety of hardware, the situation we've just described is where UNIX was until last year. Variety is the spice of life, but not the grease of commerce.

OSF says it doesn't want to reinvent the wheel, but given the level of resources that will be devoted to the task, maybe it is time to reinvent the wheel, or at least license the underpinnings, from, say, Key Logic, page 27. Sun and AT&T, for one, will be doing a major object-oriented rewrite, System V.5, for release in the early 90s.

Release 1.0	 27 June 1988
THE METAPHYSICS

Our overall conclusion is that OSF probably won't matter much. All in all, we'd agree with Ken Olsen of DEC, who told BusinessWeek last month (perhaps slightly out of context, to give him benefit of the doubt) that standards are "snake oil" with all the appeal of "a Russian truck." While the notion of open standards is philosophically and morally appealing, it doesn't make much business sense. This group of companies is not doing anything wicked; they're just self-interested, as Sun and AT&T are. What's new is the perception of how best to serve that self-interest, and a switch in approach from a group of vendors who by and large sell proprietary systems.

OSF is attempting to groom the ski slopes of progress into a wide, even terrain, but the forces of individual effort will keep on producing moguls. Ultimately, the market works, just as gravity pulls skiers down the slopes, but it doesn't always work smoothly. It works in fits and starts; companies go bankrupt; good ideas get lost on moguls or run into trees, but the fundamental direction is constant, determined by natural forces.

OSF may have some impact, but it will be the impact of one more mogul, not a fundamental redirection of the UNIX market -- let alone the computer business as a whole. In the long run, neither side will "win." OSF may bless products built by others or even build its own, and AT&T may control the evolution of UNIX (and even get other companies besides Sun to join it), but neither can control the market itself, nor force customers to buy any product that they don't want.

No single company, even a non-profit one set up by a large group of competing vendors temporarily working in concert, can hope to create a standard lasting for more than a generation or two. If it is successful on the first round, market forces will overthrow it a season or two later. There's no reason that a committee should be any better at choosing, let alone creating, software than the best of a group of companies. That's right: The market lets a number of companies try, and then selects the best one, rather than ordaining from the start who should create the standard.\(^2\) To the extent that OSF is content to play that role or simply endorse the work of other committees, fine, but then it becomes just one more standards body.

In sum, the Open Software Foundation is just one more industry player. While its broad parentage has helped in gaining it notice, that same broadness of interest (read: interference) is likely to hamper it in getting its job done. The OSF staffers are sincere about avoiding creeping committee-ism, but the organization's devotion to openness and consultation with members may make that inevitable. Meanwhile, Sun and AT&T (if they can avoid distraction) will be selling a real product with a binary standard interface to customers and ISVs, who ultimately buy a product, not a standard.

Thus, until OSF produces a product some 18 to 24 months from now, there's not much reason to take it too seriously. It's not stopping anyone (AT&T, say) from doing anything, it's not selling anything, and it has pledged to support standards. Who could argue with that?

\(^2\)Yes, yes, we know, IBM is pre-ordained to set standards by now -- but that's an arguable proposition. Look at the 8100 (remember?), the RT PC, AIX itself. OS/2 would not be a standard without Microsoft.

Release 1.0

27 June 1988
COUNTERPOINT: IBM'S OBJECT-CODE-ONLY POLICY

Since long before OSF was even dreamed of, another standards/openness battle was raging, over IBM's object-code-only policy instituted in February 1983. This is the other side of the coin, where IBM's proprietary stance riled a number of folks (particularly at a time when it's bending over and selling the stuff to Fujitsu -- a special irony since the policy was designed to protect IBM from Japanese inroads). In fact, just as we think that the notion of OSF is fundamentally flawed, so do we think that IBM should be able to do as it pleases with its source code. Adapso, a leader of the charge, recently withdrew from the battle after surveying its members and discovering that only a couple of them care, but the battle still rages.

In fact, IBM is trying to do precisely what Sun and AT&T are attempting with UNIX -- to make sure that there's only one version of the product around. Formerly it licensed source code to interested parties; now it will distribute only object code for new products (such as DB2 and OS/2) and significant upgrades to new ones. (Obviously, you can always argue whether new releases qualify as new products, cf. the changes in AT&T's licensing of UNIX.) The underlying assumption is that "Improvements" by third parties are unlikely to provide benefits substantial enough to offset the additional support costs, lack of standardization, migration problems, etc. IBM doesn't want to support their changes: People who take object code and muck around under those layers have to do all the work themselves when they change something, and are subject to a heavy maintenance burden when IBM changes something. (IBM's own programmers on OS/2 Extended Edition, for example, get no special access to the OS/2 source.) At a time when users are complaining about the proliferation of minor differences among supposed standards, and most users want to be insulated from systems issue, this attitude seems to make sense. "We used to have big problems between releases," says an IBMer, "but with the new policy the upgrade from MVS/XA to MVS/ESA is going pretty smoothly."

Yet some third-party vendors (mostly systems software) and a couple of customers complain that they can't get their hands on IBM's software to make changes as they used to. Of course, everyone who sells a fix to MVS and indirectly deprives IBM of some potential business is also making MVS that much more attractive to customers, which is why IBM is working with a small number of vendors to solve their problems without giving away the store. In short, this issue is losing its vigor -- not because it will ever be resolved, but because it will be handled, case by case. That's the only way we know of to deal with such the inevitable conflict between all or nothing.

Not in conclusion...

The point of being a grown-up is learning to deal with uncertainty, with the knowledge that there is no right answer -- always give out source, never give out source; license everything, license nothing; keep full control, give it away. The mature company (and person) must have the courage to be arbitrary -- and to take responsibility for the consequences. The smart guys make decisions with favorable (or manageable) consequences more frequently than the dumb guys, that's all. As for the truly wrongheaded, Vittorio Cassoni got it right: Darwinism will take care of them.

Release 1.0
27 June 1988
WIRED APPLE: THE MAC AS INTERFACE TO A NONSTANDARD WORLD

As half the world tries to deal with incompatibility and achieve interoperability by decreeing global standards (page 1), Apple takes another tack: retrofit technology. In essence, the company says: "Look, don't worry, keep what you've got back there. Just let us be the front-end. If you believe with us that the ultimate consumer of computing effort is the user, then let us be the funnel that delivers all that effort -- in a consistent, comprehensible way." And to vendors: "Just work with us to support our hooks into you, and don't worry about anything at all." (This falls flat when machines do need to talk to each other; see page 13. Where IBM once thought people could communicate most naturally via mainframes, Apple seems to think computers can communicate most naturally through people.)

In the last few months Apple has made a number of "communications" moves, seemingly similar to Sun's acquisition last year of Centram/TOPS. But their motivations are quite different. Sun believes that the network is the computer, and wanted a network that could hook up Macs and pcs as well as its own workstations and larger machines. (It also wanted a presence in retail distribution, although it hasn't done much with it yet.)

By contrast, Apple places the locus of power in the front-end. Apple wants the Mac to be everyone's window, connecting people to the rest of the systems world. Thus Apple can be a systems integrator with a standard product offering connectivity to all major environments: PCs, IBM SNA, DEC VAX, UNIX, SQL, TCP/IP and OSI...and you've pretty much got everything covered.

Other people's problem -- lack of standards in a multi-vendor world -- is Apple's opportunity. Let's put it this way: Apple doesn't need standards, and it doesn't need to be Number One. Its edge lies in its ability to let users hook into a diverse world of multi-vendor systems, and deal with them all intelligibly.

Setting up the links

Hence the deal with Novell. Novell will provide support for AppleTalk within NetWare rather than the other way around. That is, Mac users can keep their familiar icon-based environment to communicate with the growing installed base of NetWare-linked PCs or gateways to hosts. They can now download files or, using CL/1, execute transactions on remote databases. To be sure, NetWare does nothing to enable Mac applications to manipulate data in DOS files -- still a problem outside a few cases such as Excel and PageMaker which run in both environments. (Mac users who hook into 3Com networks must use a foreign 3Com interface on their Macs, whereas NetWare's SPG (Service Protocol Gateway) accepts the AppleShare calls and converts them. SPG will eventually be extended to handle NFS and other calls in the same fashion.)

The Novell event, complete with neon, photo opportunities (Ray Noorda and John Sculley spontaneously shook hands several times), and cadres from Regis McKenna, which happens to work for both firms, far overshadowed the deal with Orion Network Systems, announced the same day. Orion built Apple's Mac APPC, and also supplies a long list of other OEMs with SNA utilities that enable them to communicate with IBM equipment (or with each other using IBM protocols, for that matter). Customers include Wang, Banyan, Olivetti, AT&T, and Apollo. But Apple didn't want just Orion's SNA-protocol fish; it
wanted Orion's fishing school. Orion knows about not just SNA but communications issues in general: network management, directory services, etc.

At a higher level, Apple is relying on Network Innovations (NI, acquired in March) and its CL/1 connectivity language for connection to SQL databases of all stripes. CL/1 lets the Mac, using a number of Mac applications such as spreadsheets and file managers, act as a front-end to almost any SQL database. Thus the Mac can now operate as a front-end to, say, Sybase's SQL Server on OS/2, any Oracle database on anything, Cullinet's IDMS/R on a VAX, or IBM's DB2. The value here is that CL/1 avoids the wholesale transfer of files that most other methods require. CL/1 allows the Mac user/application to run a database transaction remotely. The CL/1 user, without knowing it, is operating as a client of a remote database server.3

Windows on the world

All these tools link Macintosh users to a more powerful world outside. Coming from its background of incompatibility and focus on the user interface, Apple understands not just the mechanics but the metaphysics of the client-server architecture, perhaps better than any company around.

Strategically, Apple has been awesomely successful because the battle has moved to terrain for which it is ideally armed: the user interface. When computers weren't very powerful, it took every bit of their strength to do calculations, and there wasn't much interest in "user-friendliness." Users were cheaper than MIPS, generally speaking. With the advent of more power and of pcs -- basically, the last decade -- the investment, interest, resources, effort and payback were concentrated on the users' desks. There was enough power to make these things accessible to real people -- and the power these machines gives to people provides genuine economic returns.

But long-run, the value-added -- and the profits -- in the computer business will again shift away from the interface. Now, with even more power and yet further advances in design, we're moving our attention to what goes on between computers, over the wires. More tasks are being automated, with hardware action now taking place in back-end systems (for database, graphics, expert systems, etc.) too powerful and costly to work one to a seat. At the low end the action is in groupware, for which the Mac currently provides an inadequate server, without multi-tasking or a standard database server.

The question is: Is the front-end terrain large enough for Apple to carve out a secure future? Or will the company have to move ahead and compete with the big guys on their turf: industrial-strength back-ends?

3To put these situations in perspective: Apple would not want to own Novell, since it's far more valuable as an independent third party supporting an installed base of NetWared systems for Macintosh users to hook into. By contrast, Apple does not need to keep Orion and NI out in the world to provide an installed base of SNA and SQL systems for Mac users to hook into; IBM and the SQL vendors can take good care of that (with a little help from Orion's OEMs.) Likewise, IBM is disinclined to own either Intel or Microsoft, both of which serve a world of third parties that IBM wants to work with -- since it can't own them. On the other hand, IBM bought Rolm because Rolm represented valuable expertise that could be brought in-house rather than an independent power base IBM wanted to be close to.
GROUPWARE: A FRAMEWORK

The "purpose" of computers is to automate work; for that, the work must be defined explicitly and routine processes and data structures must be identified. In the beginning, we discovered the routine tasks underlying accounting, transaction-processing, and the like. With the advent of PCs, we progressed to financial model-building (Lotus 1-2-3) and text-processing (database management had already been addressed on mainframes). With the current world of networked PCs and departmental servers, we are beginning to understand the processes of people working together well enough to automate them, using tools and structures loosely classed as "groupware."

What is groupware? You could argue, as many marketing people do implicitly, that it is any software that supports group work (such as a multi-user database or at a higher level a graphics package which aids communication). We prefer to think of groupware as software that does something to information rather than just store it or pass it along. That is, groupware is active in behalf of a group, as opposed to agents, which are active in behalf of a person. (Groupware, of course, can coordinate agents and resources, such as schedules or data or access to physical objects such as meeting space.)

The appropriate metric for groupware is not, "Is it true groupware?" That's like asking if something is art. The proper question is, "Assuming it's groupware, how much of the work does it automate? And how much flexibility must the users trade off for its power?" The ideal tool does all the work but gives the user total control. (Yes, we know, we haven't seen it yet.)

Groupware is changing the industry's focus from the machine-user interface to the machine-machine and person-person interface. It needs powerful structures for data representation rather than data presentation, with data structures that can be manipulated by a computer rather than the conversion of information into images that can be interpreted by a human. (That's valuable too, and can of course vastly improve communication between people, but it's not what we're discussing here.) Groupware supports exception-handling; indeed, the major function of groupware is to treat exceptions as routine events that can be handled by a computer. The best groupware handles the routine stuff and thereby frees people to exercise individual initiative and flexibility. Groupware acts without explicit instruction from its users (except for the initial set-up process).

Nouns and verbs; things and actions

To return to our original premise, automation requires discovery of the routines or structures underlying work, and groupware requires discovery of the patterns underlying group work. Structure is regularity superseding anarchy, making apparent disarray orderly and subject to computer manipulation.

In groupware (as in any system), there are two basic focuses of routine: Transactions and information. Thus, groupware comes in two forms: trans-

4 Take, for example, the airport control room described on page 24. Each night the computer builds a daily schedule of gate assignments; the work of the people in that room is to manage the process of changing those assignments through the day, all of them due to unexpected, unplanned problems. Without those exceptions, the people in that room would have little to do.
action-oriented, and information-oriented. The underlying technology may be the same (typically a database, not necessarily relational), but the focus and the logical constructs are different. Transaction-oriented groupware tends to be automatic, since it handles the activity of the system, routing items and managing the workflow (or transactions). Information-based groupware generally does its work at a user's direction (either at runtime or through canned scripts), with a focus primarily on organizing information for group access and reconciling and assimilating individual efforts into a joint work product or information base. The distinctions break down in extremis, and the best groupware does both, but these notions provide a useful taxonomy, as shown below.

### A framework for groupware

<table>
<thead>
<tr>
<th>Structured information</th>
<th>No structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Server (86-6, passim)</td>
<td>SQL*Mail (88-5)</td>
</tr>
<tr>
<td>generic dbms, o-o dbms (87-8)</td>
<td>Info. Lens (86-10)</td>
</tr>
<tr>
<td>Staffware (87-12, 88-2)</td>
<td>Nastec LifeCycle Mgr. (86-5)</td>
</tr>
<tr>
<td>Lotus Notes (88-3)</td>
<td>Context DME (88-2)</td>
</tr>
<tr>
<td>calendar/E-mail/OA#</td>
<td>generic project managers Expedition (87-6)</td>
</tr>
<tr>
<td></td>
<td>FileNet WorkFlo</td>
</tr>
<tr>
<td>NewWave (87-12)</td>
<td>ViewStar (88-3)</td>
</tr>
<tr>
<td>Concordia (88-4)</td>
<td>DocuForum (87-6)</td>
</tr>
<tr>
<td>TEAMate (88-4)</td>
<td>Atherton BackPlane (88-5)</td>
</tr>
<tr>
<td>multi-user hypertext (87-11)</td>
<td>multi-user hypertext (87-11)</td>
</tr>
<tr>
<td>ForComment (86-11)</td>
<td>E-mail files Workhorse (88-2)</td>
</tr>
<tr>
<td></td>
<td>WP files Scaffold (88-6)</td>
</tr>
<tr>
<td>screen-sharing tools</td>
<td>screen-sharing tools Coordinator (86-10, 88-2)</td>
</tr>
</tbody>
</table>

The diagram above represents a preliminary, fuzzy assessment of a variety of products, many of which have not yet even been formally announced (although you can find most of them described in recent issues of Release 1.0, as noted). We welcome comments, disagreements, and most of all additions.

# A large group of office automation systems that provide mail service and multi-user calendars fits roughly here. They include DEC's All-in-One, Data Acess's Office Works, WordPerfect Office, Conetics' Higgins and Lifetree's Net Results.

### Information-based groupware

Information-based groupware deals with the sharing of information, wisdom, models, diagrams, etc. Except for technically complex screen-sharing tools, E-mail is conceptually the simplest of these. Although superficially it looks transaction-based, transactions are not the fundamental unit of an E-
mail system; the messages (or information) are. There’s generally only one
kind of transaction: send-receive (or forward); reply-requested is the start
of a transaction orientation.

Basic E-mail delivers messages; groupware E-mail systems with information-
oriented extensions do things with the messages, as in Information Lens or
Oracle’s SQL*Mail. They can file information appropriately or, if pro-
grammed, take certain actions in response to it. Databases for CAD/CAM,
multi-user hypertext systems, bulletin board systems (including TEAMate, an
elegant BBS-plus) and on-line documentation (cf. Symbolics’ Concordia) also
qualify as information-oriented groupware.

Such information-oriented systems generally don’t care much about people or
timing (big issues in transaction-oriented groupware), although they may
well maintain categories of people to manage access rights, and manage mul-
tiple versions of the data/objects they store. But their concern is with
the information, not with how it got there or who owns it. The wheel is
owned by the truck, not by Alice, its designer; a picture may belong to
several documents and it is also linked by cross-reference to a mention in
the text of one of those documents.

So far, most of the groupware we see for sale is information-based. It
provides passive support, rather than active management, of group interac-
tion. Fundamentally, this kind of groupware is about communication -- en-
abling people to share information, build information bases (including docu-
ments, designs, and the like) and explore ideas. Rich displays, data manip-
ulation, object-oriented data structures and editing tools -- all these make
it easier for people to create and share a vision. This is a valuable func-
tion, and one that will continue to grow over the next decade. But it’s
fundamentally a more mature area, and better understood, than transaction-
based groupware, which is the major focus of our discussion here.

Hybrids

Some of the more powerful information-oriented systems use a client-server
architecture and incorporate the notion of transactions to help maintain the
integrity of their databases. These include such systems as Lotus Notes,
Context Document Management Environment, NETI’s DocuForum and Atherton’s
BackPlane, which support the creation and sharing of richly structured text
and images, using data structures such as hierarchically structured text,
hypertext links, or an object-oriented database. These systems also manage
some user interactions, and can be programmed to support and define workflow
transactions.

FCMC’s Staffware is unusual in being a transaction-oriented system that
manages the data as well as the transactions; it uses the data as input for
its rule-based routing system, which triggers pre-defined transactions at
runtime. A new version, Calypso (Release 1.0, 88-2), will use a client-
server architecture so that it can attach to a user’s existing database.

Action Technologies’ The Coordinator is also an illuminating system to dis-
cuss in this context, because it’s easy to misunderstand. It is a trans-
action-oriented mail system, because the mail it manages is defined transac-
tions (or speech acts, to use the company’s terminology). It defines mul-
tiple kinds of transactions and monitors the status of those transactions.

Release 1.0

27 June 1988
The underlying development tool, Conversation Manager, is in use at a number of corporate partners and will eventually be released as a product; meanwhile, The Coordinator Release 2.0 has a new improved interface that should help the product appeal to a wider audience. Action's underlying mail-manager, Message Handling Service, is now a standard component of Novell NetWare, and provides a handy underpinning at a low technical level.

Transaction-based groupware

Transaction-based groupware is primarily concerned with people and their roles, rather than their actual work product. Rather than the product, it manages the process -- control of changes, transfer of selected data to those who need it, approvals and requests, submissions and deadlines. The system monitors who has what, who owes what, who may change it, who gets it next, etc. The system doesn't care about the underlying information; it could be text, code, images, diagrams, real-world objects, whatever. Like systems software, transaction-oriented groupware doesn't accomplish anything itself; it supports the coordination of the people who are doing the work.

Transaction-oriented groupware operates independently of the applications that do the work, whereas information-oriented groupware generally operates directly on the users' data, supporting such joint information-manipulation tasks as CAD/CAM, bulletin-board operation or editing of complex documents.

Transactions: the atomic particles of groupware

If data elements are units of information, then transactions are units of activity or process, or work. Just as you can build richer, more complex data elements, or objects, so can you build rich, complex, extended transactions that comprise a number of linked transactions, integrity rules, conditional branches, and other attributes and constraints.

Like data elements, transactions are discrete and internally consistent. They can be handled as units: sequenced, assigned, referenced, identified, queried, triggered, etc. One purpose of transactions is to maintain system integrity. Transactions are required to change the status of a system: Juan assigns a project to Alice, Alice delegates part of it to Hannah, Alice completes her work, Juan refers back one item to Alice for decision, Alice promises to have an answer by Friday. Generally, we are not here talking about simple transactions such as order entry or reservation-making; those are transactions between a person and a computer, with the person often acting as the intermediary for a customer. Groupware transactions happen among people, with the computer as intermediary.

Extended transactions

A transaction in groupware isn't a simple debit-credit sort of thing; it's an extended process that may involve several people and evolve through a succession of unstable, inconsistent states. In other words, most work is accomplished during the transaction, even though system designers traditionally figure that the only good transaction is a completed transaction. A

5Of course, in underlying technical terms, the computer attempts to maintain consistency throughout, but groupware tools let users and system-builders think about and manipulate familiar, meaningful abstractions rather than software elements.

Release 1.0

27 June 1988
defined sequence (with branches) of transactions can thus represent a "meta-
transaction," with requirements for consistency and completeness. (What
happened to Hannah’s part of the project?) The terminology isn’t yet fixed,
but the point is simply that defined sequences of transactions, with concur-
rent events and branches, are fundamental building blocks of groupware. The
groupware designer must make sure that the process completes, that the order
doesn’t end up in limbo in someone’s in-basket. Good groupware systems will
provide modeling tools and perhaps AI-based consistency checkers to assist
in the building of consistent processes. They will also provide tools for
builder/users to define work transactions particular to the business -- bug
fixes, subscription renewals, phone calls, sales call reports.

The resulting system manages two sets of information: transactions and
status. (In the same way, you have a general ledger, recording transac-
tions, and a balance sheet, recording status.) Groupware also keeps track
of the various incomplete "transactions" -- a request not responded to, a
promise unfulfilled, an alarmingly high inventory of unfinished tasks in the
virtual in-basket of a new team-member. How many fund requests are waiting
for Benjamin’s approval? What’s Charles’s ratio of cold calls to follow-ups
this week? Who’s responsible for the next contribution to the marketing
plan? Who gets the plan when Samuel approves it? Who keeps delegating work
to the clerical staff? What happened to the DRAM lease proposal?

Without transactions, in database systems or in groupware, we’d have no
assurance of integrity. For example, databases generally don’t allow people
to edit data; they must update it through approved transactions. (That’s
much of what client-server architecture is all about; more on this in a fu-
ture issue.) Someone could come in and erase our bank account without cred-
it ing us for the money; Juan could delegate a task to Alice without Alice
accepting responsibility for it.

To illustrate: We saw such a system recently, a project management system
wanting to become groupware. As a project manager it was handy: A manager
can define an objective, break it down into tasks and specific actions,
assign those to individuals, and monitor their progress. But he could
change those assignments by typing in new ones; there is no process to go
through. Moreover, there is no provision for interaction among the people
doing the work; they all report to the manager. The tool was a manager’s
monitor rather than a groupware tool.

A peek inside

The concept of transactions described here is a general one; over the next
few years, we’ll see a variety of tools and a variety of approaches. We’re
early in the process of defining what a transaction is and should be, and
there are likely to be furious arguments about it, rather than the gradual
uncovering of a single truth. It’s the same with object-oriented databases;
there’s still too much to discover before we can even hope to create stan-
dards. The things that need standardization are awesomely complex.

Aside from transactions, groupware also has its analogue to data types --
people’s roles, generally linked to the kinds of transactions they may or
must perform. The data structures are relationships among people: owner/
user, requester/fulfiller, coordinator/members, assigner/doer, proposer/
approver, etc. Multiple workgroups may co-exist, with the same people play-
ing a different role in each.

Release 1.0 27 June 1988
Underneath, of course, groupware relies on the same old technology -- networks, server OS (OS/2 or UNIX or a jury-rigged DOS, generally), E-mail, remote procedure calls, shared database access, relational or object-oriented or text database management systems, etc. Groupware tools build reusable logical constructs on top of these that make it easier to build groupware applications -- people, tasks, workgroups, transactions, etc. -- without worrying about the system underpinnings.

The better we understand group work and can make its activities explicit, the better we can model it on computers. As we discover the underlying concepts, it also becomes easier to build extensible tools rather than applications that model only a single situation (such as the early joint-editing tools). For example, Coordination Technology originally designed its product as a scripting tool, a language plus data structures; after working on the problems for a couple of years, it discovered that it had abstracted out enough common constructs to offer templates and mostly menu-driven procedures which users could customize and assemble into groupware applications.

VisiGroup?

Precisely because groupware is so abstract, it will be valuable to develop metaphors or graphical constructs that can render group dynamics clear and group interactions easy to model. FGMG’s Tony Kobine, for one, is talking to a number of CASE diagramming tool vendors to explore how their tools could be incorporated into his next groupware-development tool, Calypso. And EDS’s Apache system, built in-house to run on a Mac, lets builder/users diagram the transactions or flows of work among people as virtual paper shuffles its way through an organization. Perhaps the simplest rule of groupware design is: Follow the paper. Many interpersonal transactions occur by way of a buck slip. (Why is it that it’s easier to give someone a spreadsheet to work on than an order? Because you can pretend you’re sharing information rather than giving the order.)

Tools, not applications

As the press and commentators scurry around in the Easter egg hunt for the first few OS/2 applications, the smartest companies are working on the chickens that will lay those eggs -- perhaps not golden, but definitely shiny. The secret behind 1-2-3 and VisiCalc before it was that they were not truly applications but rather modeling tools -- suitable for users at the current stage of the market’s development. The task could be done with a financial planning language on a mainframe, but that took too much effort for a piddling result. PCs and spreadsheets changed all that. Likewise, the little notes we send to our secretary (and the follow-up) could go by pc and LAN, but that seems ridiculous now. Where is the tool that will make it worthwhile to automate all those pesky, repetitive little tasks requiring diligence but no intelligence?

------------

6In this context, we remember the commotion we caused every time we released an org chart at Computer Industry Daily, a previous employer. Somehow everyone imagined that they reported directly to the top; the clear, explicit information in the chart wasn’t always welcome -- but it ensured that misunderstandings were resolved early!
Now we have a new class (we hope) of users -- not spreadsheet-wielding staff analysts, but line managers, trying to get the work out. The hardware environment is also different -- not just more powerful, faster machines, but linked machines -- and a reasonable hope of finding (or putting) one on most people's desks. Indeed, much of that work is being done on computers, but it's not managed on computers.

The new software that will really establish OS/2 or UNIX, then, is not a super-duper, all-singing, all-dancing spreadsheet with icons and AI, but rather a sober, practical tool that will help managers automate their work.

Beyond bureaucracy?

How many of the business procedures that we've evolved are due to the difficulties of doing things, and how many inherently make sense? Users of Lotus Notes, for example, first appreciated its ability to help them schedule meetings. They soon found an even greater benefit: Frequently, it eliminated the need for meetings.

The long-term impact of groupware should be to reduce the height of corporate hierarchies, and all the overhead, miscommunication and heartbreak that they invariably engender. As workgroups extend outside companies (ask Novell or GM about this), we may see fundamental changes in corporate structure and the ability of small companies to operate in a world of large ones. Yes, groupware is about bureaucracy, but bureaucracy, like law, protects us as much as it restricts us. The problem with most bureaucracies is that they cannot handle exceptions, because the exceptions are not defined. The more we can define, the more we can handle. Good groupware is about coordination of ad-hoc activities, not rigid structuring of people's time. Good groupware will free us to address more interesting matters by handling the routine ones automatically.

COORDINATION TECHNOLOGY'S SCAFFOLD

Priming the market early, Coordination Technology Inc. is attempting to define groupware, in a brave (or is it foolhardy?) product launch starting now for a groupware tool promised next summer. Its use of OS/2 is a partial excuse for the timing. As with Lotus Notes, it's a pleasure to hear of a product that's fresh on OS/2, not a warmed-up leftover.

Take note: The product we're describing is a prototype; what you don't like can be changed. Developer Coordination Technology, with a budget of $4 million this year and $10 million for 1989 funded mostly by New York City investors E.S. Jacobs & Co., has ample resources to work on it with little outside pressure beyond the nudges of 15 unidentified pilot users.

CTI's tool has no name yet, so we'll call it "Scaffold." Scaffold is a tool for building work groups, complete with roles, transactions, and associated programs and data, along with a system to monitor the work processes (transactions) the builder/user defines. The builder/user can assign an arbitrary number of users to an arbitrary number of workgroups; end-users can also ask to join or set up workgroups. The user's activities are managed in centers, one for each workgroup he works in. As shown on the next page, the view of the person or resource is a row; the view of the workgroup is a column.

Release 1.0 27 June 1988
Each user (except for the lazy ones!) operates a number of centers, in which he assumes different roles and uses a variety of tools, data and other resources. While resources can be shared all around, for the moment data is assigned to a single workgroup, so that a user may not transfer his own information easily from center to center (but that could be fixed).

From the system’s point of view, all the work items and data are objects; in true object-oriented fashion, it ignores their internal behavior and characteristics. It takes no interest in the actual work product, but simply monitors the group members’ transactions, such as reporting completion of a task or assigning work to someone else.

Originally, designer Tolly Holt (who describes his work as the study of coordination mechanics and has written some illuminating papers about it) conceived of Scaffold mostly as "dynamic glue," an interface plus a scripting language, along the lines of H-P’s NewWave.7 (We wrote about it as such in Release 1.0, 87-12.) But Scaffold rapidly evolved into a set of script templates or transaction models ("diplans") that could be assembled and reused with variable particulars. Although the details vary from case to case, the fundamental dynamics of most situations follow a very few patterns (cf. The Coordinator’s speech acts, where the content is immaterial, but there are specialized transactions). For example:

---

7Unlike H-P, CTI realized early that it’s futile to do this sort of thing in a single-tasking OS. Waiting for processes to complete is tremendously annoying for users; refereeing their unpredictable actions is virtually impossible for the developers.

---

Release 1.0

27 June 1988
• Assignment of work. This can be a single person to a number of people, or can be a number of chains, where people assign work to people who in turn assign (or pass) it to others. (Don’t be so hierarchical!)

• Reporting completion of work.

• Approval of work.

• Management of resources where a single person controls them.

• Management of resources where multiple people’s needs or wishes must be reconciled.

• Management of resources where the focal point keeps an inventory.

• Scheduling or management of time.

• Monitoring of work (collection of transaction reports).

• Consensus mechanisms such as voting on a decision, which may be iterative.

Although the prototype we saw was just that, a prototype, it did a nice job of presenting information to the user. Without fancy graphics, the system displays to the user in outline form the workcenters he owns and his role(s) in each. He can go down a level for more detail, and select actual tasks to work on, at which point a pop-up menu starts to guide him. Various cues indicate incomplete transactions: He has been assigned a new task; something requires approval; a meeting has been scheduled and will he attend? This simple interface gives the user a clear sense of his (computer) roles and capabilities, and politely notifies him of all the work he has to do -- all in its proper place.

Scaffold takes care of all the workgroup transactions -- sending stuff to people, reporting on actions, etc. All the user has to do is his work -- not the coordination of it with others. This may sound like the trivial part of the task, but any activity seems trivial taken in small pieces; it’s Scaffold’s abilities to manage all the people, all the time, without slipping up, that’s so valuable. And yes, we forgot to mention, it will handle all those pesky problems of network administration, installing software, and the like -- or rather, it will let the system administrator handle them once, and make it all look easy to the users thenceforward.

PC/FORUM: VOICE ANNOTATION

Earlier this spring we saw a demo of Forum Systems’ PC/Forum, just about to be released to much fanfare, with initial distribution through selected ComputerLands, etc., etc. We and a few other observers were dubious for the reasons noted below. The company informed us that the product would not be released in its current form, and would we please hold our fire. We did.

Release 1.0

27 June 1988
We encountered Forum Systems again at the recent Brandon groupware conference. An all-new management team (including turnaround artist Dan Rasdal from investor Hambrecht & Quist) was showing the same product, but as a prototype. Forum Systems is now soliciting feedback before it launches PC/Forum, probably sometime in 1989. These comments are our contribution to the discussion. They are directed at a product concept, and are broadly relevant to more than just this particular product prototype.

Forum Systems’ PC/Forum sits at the leave-the-intelligence-up-to-the-users end of the groupware spectrum. It doesn’t do much, beyond providing the ability for users to embed voice comments (or other sounds!) into or onto files they send to others or want to review themselves later -- basically, it’s voice hypertext, links between regular data and voice data. This facility has enormous potential to enhance the value of an application -- especially if it is properly integrated with the application. Embedded voice annotation will let users carry on a conversation across time and space with reference to specific items within a data/application file. That is, the application data itself would provide a way of organizing the conversation. A particular value in a spreadsheet, for example, could be annotated with a voice comment (betraying the right level of ambiguity) about the reliability of the source or the implications of the ratio.

However, in the current version PC/Forum cannot attach its voice annotations to particular points within any file other than ASCII files (which it can manipulate with its own vanilla word-processor); it can simply attach them at the head of such application files. The original developers of PC/Forum seem to have expended more effort on creating an environment for this capability than on integrating it into and enhancing existing, already useful environments and applications. While a whole new environment could be considered value-added, we’d be more comfortable with PC/Forum if it promised less. Rasdal is currently in talks with the appropriate vendors, who should recognize the potential value PC/Forum could add to their products. We hope the company will now spend the next few months integrating it with 1-2-3, Excel, a couple of word-processors, a few database managers, and perhaps some mail packages. It’s a pity, of course, that PC/Forum can’t automatically work with all applications, but as usual the disparity will work in favor of the big guys -- if they are clever enough to help it happen.

Spreading it around

Meanwhile, Forum Systems is also working on a low-frills front-end, to replace its current baroque interface of expanding envelopes for mail and a profusion of buttons for the voice features. The screen takes a long, long time to refresh itself and probably gets extremely tiresome for frequent users of the system. Moreover, that user-friendly interface is deceptive, because it’s a front-end to a limited range of capabilities: In order to do any real work beyond the sending of messages, a user has to leave it and go into his other applications.

Finally, there’s the issue of pricing. As with any communications product, the value of PC/Forum’s voice annotation could best be realized if it were widespread, since users will be more likely to find other users they can trade voice messages with. Original (last-spring) plans were to sell the product for between $950 and $1250 per person at retail; now all bets are off. In addition, the product works best in a system with a total of 20
megabytes of mass storage. All we can say is, the cheaper the better. This is a technology powerful enough that it deserves a booster start, with hopes of making it up on volume later.

Forum Systems' history may explain its approach. It was founded as Santa Barbara Laboratories in 1982 by Mike and Charity Cheiky, founders of Ohio Scientific and speakers at the 1980 PC Forum. The Labs' first product, Centerpoint 1000, was a special-purpose hardware system running the CP/M-86 operating system and sold for $25,000 -- 90 systems to about 20 customers. Investors including Oak, Arscott & Norton, Hambrecht & Quist and Hewlett-Packard brought in a first round of new management in 1985, and scaled back the company's ambitions.

Versus HP's NewWave

For the sake of contrast, consider Hewlett-Packard's NewWave, which similarly creates a new environment, but uses rather than bypasses Windows. As it happens, H-P is using the Forum Systems voice technology, treating it as just one more kind of object in a system that supports multiple types of objects. Voice annotations can thus be embedded in NewWave's compound documents -- a sequence of objects that look like a document on the screen (and can be sent serially to a printer).

Incidentally, H-P faced the same integration problems as Forum Systems, and has gone to the trouble of working with a number of major applications vendors so that it can embed objects directly in their application files. (The most significant of these is Microsoft; see below.) Now, it's easy to extend that infrastructure to incorporate Forum Systems' voice objects. Moreover, NewWave lets users manipulate the system they inhabit, with programming facilities to create scripts (or agents) and integration with a number of leading applications such as Excel. NewWave extends other vendors' tools; Forum Systems attempts to replace them with a single feature that lacks the breadth of function most users need. If NewWave can just round up enough support, PC/Forum should be able to ride on its coattails, but despite HP's progress with Microsoft, that's not a very safe bet. A lot of people may want PC/Forum's voice capabilities without necessarily committing to the whole NewWave environment.

RELEASE 1.1: EXCEL SUPPORTS NEWWAVE

Microsoft has just announced that it will put hooks into Excel to support NewWave. This doesn't represent an enormous amount of effort or an overwhelming commitment, but it does indicate that Microsoft is sincere in its endorsement of the concepts. (See Release 1.0, 87-12.) Users will not have to buy a special version of Excel; the fixes will be incorporated in the next general release of Excel. Basically, this will allow users to put NewWave objects into Excel spreadsheets, as well as to put Excel objects into NewWave compound documents. With 1-2-3, for example, for which there is as yet no announced agreement with Lotus, NewWave users can dynamically import hot-linked 1-2-3 data. But they can't insert links to "foreign" data into a 1-2-3 spreadsheet, because that would involve a modification of 1-2-3 along the lines of what Microsoft will do for Excel (and what Forum Systems will be asking vendors to do to incorporate its voice objects).
NITTY-GRITTY EXPERTS: GATEKEEPER -- A SYSTEM FOR ALL PLATFORMS

The benefits of databases apply in almost any setting. As airlines are learning the benefits of expert system technology, expert-system builders are learning the benefits of layered architectures. GateKeeper, an expert system for airline gate management developed by Ascent Technology in partnership with System One, the dp subsidiary of Texas Air, uses an independent dbms that can also support other applications.

To the naked eye, GateKeeper looks like any other expert system: It has a fancy interface, it solves complex, constraint-bound problems, and it answers questions about its reasoning. Like its brethren, it doesn't so much replace people as assist them, assembling all the information they need and presenting it in handy digestible form. If the users ask it to, it will suggest courses of action, but it doesn't give excessive amounts of unsolicited advice. GateKeeper happens to manage the assignment of Continental's gates in Houston (an application we personally find meaningful, given our travel habits), but that's almost beside the point. What makes GateKeeper interesting is its architecture, which will let it be customized easily for a variety of clients and extended to handle a variety of applications.

The heart of GateKeeper isn't its expert system at all, which is just a front-end that represented only 10 percent of the programming effort (very roughly 200,000 lines of code in all). Its heart is an Oracle database that underlies a variety of front-ends, including an expert system that assigns and reassigns aircraft to gates in real-time according to changing conditions and (soon) low-end terminals that will tell passengers where to pick up their bags and caterers where to load the mystery-meat specials.

Ascent Technology, a Cambridge start-up co-founded by Karen Prendergast, former business manager of MIT's AI Lab, and her husband Patrick Winston, the LISP guru, designed GateKeeper for System One with the goal of remarketing it to other airlines very much in mind. "We estimate it took an extra 50 percent of code and halved our performance to make the system portable," says the third co-founder, Philippe Brou, formerly a consultant to Symbolics (and a loyalist who's delighted to see Symbolics focus on its superior software development environment). GateKeeper is written in Common LISP, developed within the Symbolics development environment Brou loves but constrained to run on any Common LISP platform, including the 386, its major target. "We've talked to ten airlines," says Brou, "and not one of them wants a LISP machine." The display is X Window-based, also for machine independence.

Parse me some messages!

The most valuable and unusual part of the system is a parser written in C, 30 percent of the coding effort, that catches and interprets the steady stream of messages issuing from Continental's central operations. Some of these come direct from a computer or automatic sensors in airplanes; others are typed in by aircraft routing personnel. In the old days, they went to printers where someone would tear them off and shout, "118 delayed half an hour out of Omaha!" or, "865 just in at Gate 2!" or, "Tail number 4870 will fly flight 445 to San Jose!"

This data -- real-time flight information, passenger counts and the like -- is loaded into GateKeeper's Oracle database. Longer-term, Ascent and System

Release 1.0

27 June 1988
One will build a ramp-management expert system to control assignment of ground personnel such as baggage-handlers, cleaners, and ticket-takers -- people who will shortly use the Oracle database to find their aircraft but are not yet scheduled and rescheduled by it. Eventually, they hope to build a "hubkeeper," which will use a similar system to handle aircraft routing.

An interface a gate controller could get really excited about.....

GateKeeper models its user interface on current paper systems, and shows a grid (above) of the gates through time, with the presence of aircraft noted by colored bars. Conflicts show up in red; other colors indicate the type of aircraft. (Ascent originally considered a chart which showed a bird's-eye view of the airport layout at a given point in time, but the controllers found it pretty useless. "Whaddaya want us to do?" they asked. "Bring our Crayolas?" They preferred the higher information bandwidth of a time-based graph and familiar if cryptic messages to the prettiness of the chart.) The display can explain its reasoning (why did you select this gate?), show all the gate changes made during a specified period and rank the efficiency of gate assignments as well as answer typical database queries.

Unlike some other systems, GateKeeper is fairly passive, offering advice on gate changes only when asked. Its value is due not to its expertise alone, but to its value as an information-display system that can perform some optimization queries on request. (The technology for those optimizations happens to be rules and constraints rather than linear programming.) In fact, the decision-making gets pretty easy when the information is easy to see. This reminds us of American Express's Authorizer's Assistant, another system where much of the value came from its ability to collect information from
disparate sources (IMS and TPF, mainly) and present it intelligibly to human authorizers. Like GateKeeper's, the expert system component was an extra tool to help users pick "intelligently" through huge volumes of information.

On request, however, GateKeeper can reconfigure gate assignments while following some 40 constraints or rules per gate. These include: assignment of international arrivals to the four gates with customs facilities in Houston; required time periods to clear and clean an aircraft, depending on configuration and passenger load; various rules aimed at limiting the distance passengers must walk to switch flights (usually abandoned in a pinch); and a rule that always places the New York City flights at Gates 1 and 2 (because the New York passengers tend to be...outspoken, the Continental representative explains delicately).

Each night, the system (in batch) builds a plan for gate assignments for the following day, based on scheduled flights. By the next evening, that plan is usually totally rejiggered as the result of unexpected events during the day. (Only their details are unexpected; it's sure that something will be cancelled, something will not work, etc.) On the day we visited the control center for Continental's gates in its Houston hub, a flight to Cancun had been delayed by mechanical problems. An alternate aircraft was selected, causing a gate change, but in the end the original aircraft was repaired because the captain who could have flown the replacement aircraft (a different type from the original) was not due in until 15 minutes before the scheduled departure, too short a turnaround time.

While the system's real-time rule-based powers generally aren't used more than a couple of times per flight bank, or 10 to 15 times a day, they can save a lot of valuable time and effort on those few occasions. System One estimates the system will save $800,000 to $1 million per year, based on more efficient gate assignments and quicker reassignments, faster aircraft movement, reduced baggage and passenger misconnects and reduced labor costs. Better yet, it will let airlines change schedules more easily, both day-to-day in real-time and month-to-month in response to changing market patterns.

Sharing the wealth

As noted, the two companies will market GateKeeper to third parties after installing it at several System One (Continental and Eastern) hubs this year. Like most successful expert systems (especially ones that can be sold outside the location where they were developed), GateKeeper solves a discrete problem, relies on existing data sources, and fits in neatly into an existing work environment. That is, it's not disruptive, either to people or to current systems. It certainly doesn't replace people (although that may be management's long-term goal), but it makes their jobs easier.

Ascent Technology acts as an R&D department to organizations who can help develop products. Both outfits share ownership, and co-market the resulting products. That way Ascent's 15-odd people can concentrate on what they like doing, and needn't worry (once the partner is found) about their credibility in the market. System One is Ascent's only current partner, although it has worked with McDermott on a CAD/CAM system for designing offshore drilling platforms. Its technical people have worked together on projects for Chase Manhattan, Polaroid and Norton, while Philippe Brou designed the Symbolics system that tuned the car that won the Indy 500.

Release 1.0 27 June 1988
RELEASE 0.5 -- HALF-BAKED NEWS

Wordbench: Remember the DOS community! Wish we'd had this when we were in school! Wish we'd even had a computer when we were in school... The folks who brought you Bank Street Writer have now brought out (via Addison-Wesley) Wordbench, a nicely integrated set of writer's tools including an outliner, speller, thesaurus, automatic indexer and table of contents generator, etc. It runs in 256K on a PC, a nice counterpoint to the monster OS/2 programs we're all waiting. Low-end users still have high-end needs.

WordPerfect: How comes it's so successful? We found a hint in a recent company bulletin, which instructed all employees to take phone orders and help customers in any way they can directly, rather than switch them somewhere else. WordPerfect is undergoing an especially painful upgrade to WordPerfect 5.0, and we're sure this will cause confusion in the back office. But it will certainly make customers feel loved.

Typeless? Or type-cast? We keep hearing about these wonderful systems that allow you to write instead of type. Aside from their slowness and questionable accuracy, we can't help thinking that they serve a temporary market, one that will disappear as the world gets more automated, as everyone learns to type in grade school, and as more and more people's penmanship gets vestigially illegible. But our main beef with these systems is that their software is dull. Either they do data-entry only, or they offer some limited DOS environment. As we've noted before, connectivity, not compatibility, is key. What is it that you'd want to do on an electronic notepad? Aside from Alan Kay's famous grocery list (he proposes that no such Dynabook system will succeed until it's so convenient you'd take it to the grocery), we're not sure. But until someone figures that out, we don't see a huge future for these things.

RELEASE 1.5 -- UPDATES

Verity Systems We wrote about Verity last September (without the name) when it was a division of Advanced Decision Systems. The company has since spun off into an independent company headed by Mike Pliner, founder of Sytek, and Cliff Reid, who started and ran the business for ADS. Funding of $3 million comes from U.S. Venture Partners, Olympic Ventures, and Growtech. Verity's product, TOPIC, is a tool for creating subject taxonomies (or topics) that can be used to search text databases. In essence, they are rich compiled queries that search an inverted file of the text for a selection of words that represents the topic requested. Most of the processing takes place locally, at the user's Mac or PC, avoiding a heavy load on the central text server. (See Release 1.0, 87-9).

Transform Logic We wrote about Transform Logic in December 1986. Under new management (a recurring theme, eh?) since mid-1987, the company has just signed an agreement with IBM whereby it will incorporate some of its COBOL program generation facilities into IBM's CSP (Cross-System Product) CASE tool. (See Release 1.0, 86-13.)

Key Logic We mentioned Key Logic's new management in our May issue. As of this issue, the new management has let go most of the old; president Marc Butlein, an import from the Gartner Group in mid-1987, remains. Butlein is now looking to find a buyer or a corporate partner. "Save me before I have to go to overseas!" he says in only partial jest. (See Release 1.0, 88-5.)

Release 1.0 27 June 1988
RESOURCES & PHONE NUMBERS

Rick Mellinger, Action Technologies, (415) 654-4444
Linda O'Brien, Addison-Wesley, (617) 944-3700
Mike Homer, Les Szabados, Apple Computer, (408) 996-1010
Bruce Henderson, Ascent Technology, (617) 868-0765
Mike DeFazio, AT&T, (201) 522-6413
Roger Moody, Tolly Holt, Coordination Technology, (203) 268-4045
Tony Kobine, FCMG, (415) 549-7100
Dan Rasdal, Forum Systems, (805) 683-5661
Bill Murphy, Bill Crowe, Hewlett-Packard, (408) 447-1465
Marc Butlein, Key Logic, (408) 496-1090
Craig Burton, Novell, (801) 379-5900
John Paul, Alex Morrow, Open Software Foundation, (617) 250-0035
Paul Rampel, Orion Network Systems, (415) 649-4000
Scott McNealy, John Doerr, Sun Microsystems, (415) 960-1300
Cliff Reid, Mike Pliner, Verity Inc., (415) 960-7600
Ronald Davis, Transform Logic, (602) 948-2600

Two UNIX publications we highly recommend:

UNIX Review, from Miller Freeman, (415) 397-1881 (literate and thoughtful)
UNIX Today, from CMP Publications, (516) 562-5000 (newsy and lively)

COMING SOON...

- Active objects. (It is expected that passive objects will also be discussed.)
- Database styles: Oracle, Sybase, Interbase, Cullinet, IBM et al.
- Graphics standards.
- Novell’s strategy.
- Distributed database: A revisionist view.
- Nitty-gritty experts.
- And much more...

Release 1.0 is published 12 times a year by EDventure Holdings, 375 Park Ave.,
New York, NY 10152; (212) 758-3434. It covers the pc, software, CASE, group-
ware, text management and connectivity markets, and artificial intelligence.
Editor & publisher: Esther Dyson; associate publisher: Lisa Braden; secre-
tary & office manager: Eric McErland; copy chief & transcripts editor:
William M. Kutik. Copyright 1988, EDventure Holdings Inc. All rights re-
served. No material in this publication may be reproduced without written
permission; however, we gladly arrange for reprints or bulk purchases.
Subscriptions cost $395 per year, $475 overseas.

Release 1.0

27 June 1988
RELEASE 1.0 CALENDAR


July 12-15  CASE '88 - Cambridge, MA. Second international workshop on computer-aided software engineering. More academics and less hype than most CASE conferences, for better or worse. Sponsored by several academic institutions. Call Pamela Meyer, Index Technology (organizers), (617) 494-8200, x454.

July 24-27  IEEE conference on neural nets - San Diego. The second, because the first was so successful. Keynote by Marvin Minsky. Contact: Richard Rea (exhibits), (619) 222-7477, Sue Varga, (619) 281-8991, or Nomi Feldman (papers), (619) 453-6222.


August 1-5  SIGGRAPH - Atlanta. Sponsored by IEEE, ACM and SIGGRAPH. Contact: Adele Newton, (519) 888-4534, or (312) 644-6610.

August 2-4  UniForum DC - Washington, DC. Keynote by Bill O'Shea of AT&T, replacing Vittorio Cassoni, formerly of AT&T. Sponsored by /usr/group. Contact: Heidi Thorne at (312) 299-3131 or register at (800) 323-5155.


August 15-19  Fifth international logic programming conference - Seattle. Sponsored by the Association for Logic Programming. Contact: Diana Nielson, (206) 543-2300.


August 23-28  TechDoc Twelve - San Diego. Relax with something practical after AAAI; explore the nitty-gritty problems of large-scale document management. CALS, SGML, etc. Keynote by Dave Gater, manager of publications development at Hughes Aircraft. Sponsored by Graphic Communications Association. Contact: Marion Elledge, (703) 841-8160.

Release 1.0  
27 June 1988
August 30-31 Increasing management productivity with intelligent systems - Snowmass, CO. Sponsored by the University of Colorado's Center for Applied Artificial Intelligence. Contact: Melissa Eiles, (303) 492-8229.


September 7-10 SPA fourth annual conference - Washington, DC. With Gordon Eubanks, Symantec; Victor Alhadeff, Egghead (newly public; congratulations!); John Roach, Tandy. Contact: Jayne White, (202) 452-1600.


September 25-30 OOPSLA - San Diego. Object-oriented Programming: Systems, Languages and Applications. Sponsored by ACM. Contact: Allen Otis, Servio Logic, (503) 644-4242 or Barbara Noparstak, Digitalk, (213) 645-1082. (The conference section of OOPSLA is Wednesday through Friday (28-30), so you can catch most of CSCW first if you miss the OOPSLA tutorials.)

September 26-28 Second conference on computer-supported cooperative work - Portland, OR. Sponsored by ACM. Contact: Suzanne Sylvia, (617) 225-1860.

September 26-28 The software re-engineering symposium - San Francisco. Sponsored by DCI. With Rich Currier, Panoramic; others. Contact: Marny Peabody, (617) 475-6990.

September 26-28 CD-ROM Expo - Chicago. Sponsored by IDG Conference Management. Workshops, tutorials, etc. Call Lisa D'Ambrosia, (800) 343-6474 or (617) 879-0700.

Release 1.0 27 June 1988

October 3-5  Adam Green's dBASE symposium - Burlington, MA. Three-day conference of dBASE third parties, including clones, compilers and complements. Contact: Marny Peabody at Digital Consulting, (617) 470-3870.

October 3-6  Electronic imaging '88 - Boston, MA. Hear the speeches; see the products. Sponsored and managed by the Institute for Graphic Communication. Contact: Richard Murray, (617) 267-9425.


October 7-9  Hackers 4.0 - Saratoga, CA. Theme: "Connections." Forget E-mail; you have to be there in person! Sponsored by the Point Foundation. Contact: Glenn Tenney, (415) 574-0546.


October 10-14  Computer services and communications and information systems seminars - Baltimore. The ninth annual, sponsored by Alex. Brown. All your favorite information-industry vendors, and the VCs and investors who feed on them. Contact: Donna Campbell or Leslie Engel, (301) 727-1700.

October 11-14  Info Show - New York City. Contact: Frank Fazio, Cahners Exposition Group, (203) 964-0000.

October 12-13  Bioanalysis and computing conference and exposition - Burlingame, CA (SFO airport). "Computing: Its role in bioanalysis." Data analysis, imaging, instrumentation control, etc. Contact: Norm de Nardi, (415) 941-8440.

October 16-19  ADAPSO MANAGEMENT CONFERENCE - Dallas. The software and services industry's premier gathering. Contact: Sheila Wakefield, (703) 522-5055. Followed directly by...


Release 1.0  27 June 1988
October 17-20  C++ conference - Denver. The first full C++ conference, sponsored by USENIX, with topics and speakers inherited from last year's C++ meetings and workshops. Chaired by Andrew C++ Koenig of AT&T Bell Labs. Contact: Judy Desharnais, (213) 592-1381 or 592-3242.


October 23-28  Monterey Classic - Monterey, CA. Where investors and companies discuss common goals... Contact: John Baumeister, (408) 987-4200.


October 27-29  Northeast Computer Faire - Boston. Sponsored by the Interface Group, for small systems users. Contact: Andrew Shapiro, (800) 325-3330 or (617) 449-6600.


October 31-November 2  Autofact/SME - Chicago. Plenary speakers: John Sculley, Ken Olsen. The annual factory automation event, sponsored by the Society for Manufacturing Engineers. Contact: Jill Vanderlin or Becky Alsup, (313) 271-0023.

October 31-November 2  UNIX Expo - New York City. Managed by National Expositions. Contact Don Berey, (212) 391-9111.
November 2-4  Adam Green's dbASE\textsuperscript{R} symposium - Woodland Hills, CA. See October 3. Contact: Marny Peabody, DCI, (617) 470-3870.


November 14-16  The software re-engineering symposium - New York City. Sponsored by DCI. Contact: Marny Peabody, (617) 475-6990.

November 14-18  Comdex - Las Vegas. The one and only, sponsored by the Interface Group. Contact: Jane Wemyss, (617) 449-6600.


December 5-7  Strategic issues forum - Cambridge. Sponsored by Decision Support Technology. Contact: Donna Kacin, (617) 354-6400.

December 5-8  CASExpo - Anaheim. Sponsored by Arthur Young; chaired by Howard Yudkin, president of the Software Productivity Consortium. Contact: Rhoda Canter, (202) 956-6041.

1989


March 13-18  Seybold Seminars '89 - San Francisco. The place to be published...er, seen. Contact: Kevin Howard, (213) 457-5850.


Please let us know of any other events we should include.  -- Eric McErlain

Release 1.0  27 June 1988
SUBSCRIPTION FORM

Please enter my subscription to Release 1.0 at the rate of $395 per year in the U.S. and Canada. Overseas subscriptions are $475, airmail postage included. Payment must be enclosed. Multiple-copy rates on request. Satisfaction guaranteed or your money back.

Name
Title
Company
Address
City State Zip
Telephone
How did you hear about Release 1.0?

Please fill in the information above and send with your check payable to: EDventure Holdings Inc.
375 Park Avenue, Suite 2503
New York, NY 10152

If you have any questions, please call us at (212) 758-3434.

Lisa Braden
Associate Publisher

Release 1.0 27 June 1988