INTELLECTUAL PROPERTY: LAW COMES TO THE FRONTIER

Are you selling something you don’t entirely own? If so, are its components owned by a more foresighted competitor who took the trouble to patent them or by a previous developer who will sue you for copyright infringement? Or are they something that no one can own and that anyone can copy, taking away your customers and business with a cheap knockoff?

In the future, defining and asserting intellectual property rights will simply be a cost of doing business, as are contracts and title searches and such in the real-estate business. Many of the current problems and uncertainties in the field will be wiped out with time and experience, as vendors and users become familiar with the issues and as a body of case law and practice builds up. This edition addresses current problems.

It may sound as if all this legal stuff is simply unnecessary overhead, an impediment to the free flow of information and a guarantor of monopoly profits that exacts a net cost from society. In fact these legal entanglements simply reflect the need to define intellectual property so that its development and distribution can be fostered.

There is an argument that the best technical people don’t invent for money; they do it for glory or intellectual satisfaction or whatever. That may be so, but the salespeople who sell the product and the support people who help it get installed and keep it running and who clean up the code of the geniuses generally have more mundane concerns. It’s ownership of intellectual property that coalesces those support groups into the critical mass needed to foster adoption of innovation and ultimately the creation of new (if perennially superseded) standards.

There are certainly points in the cycle where vendors seem to have too much power, or market or profit share, but things usually even out in the long run. In the continuing cycling between standards and chaos, clever players such as Adobe (page 4) foster the standards that benefit users, and are then forced either

THE TRANSCRIPTS ARE COMING!
to innovate further, or cede their growing power to others who are jealous of it. In the midst of all this movement generated by self-interest, technological progress is a byproduct.

These issues extend beyond the software business. Or, software is part of the fabric of business in general. Can you get protection for the specifics of a cash management account? Well, what about a way of assessing credit risks, picking stocks, or even processing loan applications? All these techniques can be embodied -- and sold -- as software. Software, and its manifestation in other products, will provide our competitive advantage as a country in the Nineties...if we can properly define, protect and price it.

SEE YOUR LAWYER!

...although he will disagree with us and with any other lawyer! People who ignore the implications of intellectual property laws will suffer for it. This issue of Release 1.0 is intended to command your attention so that you may be among those who benefit, rather than those who suffer. It also suggests how the patent and copyright systems could work more efficiently and predictably:

A well-lighted field

Copyrights protect software against copying of code or "look and feel;" patents make the infringer liable for copying of functional elements whether he knew about the previous work or not. Our basic plea is that the playing field should be well-lighted (not necessarily even). Developers should be required to define their property promptly and clearly, and others should be able to determine whether or not they are infringing. Market inducements for licensing are probably sufficient, but we’re open to persuasion here.

THE STORY THUS FAR

The definition and protection of intellectual property embodied in software (both functionality and interfaces) will become an increasingly visible issue over the next decade. While it can be stolen outright, its value can also be "stolen" or infringed by external actions -- such as marketing of substitutes of varying degrees of similarity. Conversely, its value can be increase by marketing, customer support and other environmental factors that increase demand -- its status as a "standard," the existence of other products that it works with and third-party support such as training and books.

Intellectual property embodied in goods and the processes used to make them was generally protected by patents and copyrights in a reasonably understood manner. But times have changed. Software has become a big business, and vendors are interested in protecting not just their code but also their user interfaces, as software reaches a broader base of users and ergonomic factors take on substantial marketplace significance. From a community working for love (pc software) or by direct contract with customers (the mainframe software business), the software business has turned into a larger marketplace of anonymous, financially motivated buyers and sellers.

This reality has become unmistakable recently as a result of copyright suits -- Apple against Microsoft and H-P, Lotus against Mosaic and Paperback, Ashton-Tate against Fox and Santa Cruz Operation. A rash of patent awards
and suits has further confused the situation and alarmed developers and inves-
tors with the prospect of unanticipated infringement charges.

At this point there are no solid definitions of software intellectual prop-
erty, let alone of how the law applies to it. It's the uncertainty, rather
than any particular policy, that is the real killer, and it's worse with
patents. While people may argue about the copyrightability of something,
it's at least clear to a possible infringer (but maybe not to his boss) that
the danger exists: You have to have seen the original to be liable for copy-
ing it. By contrast, you can infringe a patent without even knowing of its
existence. Investors want to know that companies have clear title to their
products before they invest; developers want to know that they will reap the
rewards for their own work -- and that they aren't inadvertently illegally
appropriating someone else's work.

True, customers don't want to struggle with user interfaces made spuriously
different just to satisfy legal requirements, but we don't legislate free
potatoes either, even though that theoretically should alleviate hunger.
(Actually, it wouldn't; see page 20.)

The current situation

While copyright law is sorting itself out as cases wend their way through
the courts, developers have legitimate reason for worry about the potential
impact of patents -- as administered by a patent system that is slow to act
and has little experience in the area. Once granted, patents (like copy-
rights) persist way too long (17 years) beyond the reasonable economic life
of software. Patent applicants, moreover, aggravate the problem by their
own dilatoriness in responding to Patent Office questions and in failing to
use the office's expedited-service option. Finally, the validity of patents
is difficult to assess without expensive litigation. Meanwhile, the fixes
described below (page 14) are way too slow in coming. In a fast-moving in-
dustry such as software, the current patent system doesn't cut it.

Meanwhile, vendors who rely on trade secret protection (about 75 percent of
the total in a recent Massachusetts Software Council survey, versus 8 per-
cent using patents) are playing a dangerous game: If someone independently
develops and patents an invention you made and kept secret, the patent-
holder owns it. He may stop you from selling your own product. (Patents
are granted in the absence of public prior art; secrets are not public.)

By contrast, Adobe Systems is an expert player of this game...

--------

1 Of course, not all of the respondents surveyed owned inventions that they
could have patented even if they wanted to. The kinds of things one might
copyright -- code and visual displays -- generally don’t qualify for patent
protection, although there's argument about that too. For what it's worth,
25 percent of the respondents say they use copyright -- an amazing response
when you consider that copyright applies to any published material, and
presumably to all software sold publicly. This response indicates a stun-
ing ignorance of the law and its implications. For citation of the survey,
see page 21.
ADOBE SYSTEMS: THE VERY MODEL OF A MAJOR MODERN COMPANY

Adobe Systems is a model of the company of the future, a company that traffics in rights rather than things themselves. It doesn't ship much except contracts in its basic business (although it's also doing nicely in sales of shrink-wrapped applications such as Adobe Illustrator and the forthcoming Adobe Type Manager). It has two intellectual property lawyers on staff, and uses the equivalent of 10 to 20 more as outside counsel.

Adobe has de facto relinquished its rights to the PostScript language in an attempt to make it a standard. It claims copyrights in the list of commands and written specifications of PostScript, but states explicitly (in a document issued by its legal department): "Adobe does not intend to exclude anyone from writing...programs...drivers...or interpreters for PostScript." Third parties are free to use the command list for such purposes, but they may not copy the manuals or the code in the Adobe PostScript interpreter.

In essence, Adobe is asserting, "This is protectable, but we want you to use it so we're not going to protect it. Please use our interface and help make it a standard. We figure in the long run we can do it better and improve it faster than anyone else." Display PostScript was a major such enhancement, which DEC, IBM and NeXT have licensed. (Sun preceded them all with NeWS).

At the same time the company has asserted strong ownership of its font libraries (cf. class libraries, page 19) and the tools used to create them. Developers can create their own fonts, but it's very difficult to do so without using the Adobe tools and hint technology. Thus Adobe gets revenues from licensees for rights to resell a (trademarked) PostScript interpreter from Adobe in their machines, for use of the font tools, and for rights to sell font cartridges or software implementing Adobe's copyrighted fonts (many of which Adobe in turn has licensed from originators such as Linotype and International Typeface Corporation). But users and third parties do not pay royalties for writing and selling PostScript applications, drivers or interpreters -- all activities that indirectly help to strengthen PostScript's market position.

A seamless web

"Cloning PostScript is easy," says Adobe co-founder John Warnock. "But to get us you'd have to clone a business. Our business depends on so many cross-licenses it's a web that's virtually impossible to break or duplicate.

"As a small company," he continues, "you're better off not to patent anything and just rely on trade secret protection. You're too small to get noticed and you don't have the resources. But once you get larger, you want to get your rights defined and have something to cross-license."

Font wars: WYGFOPIWYGFA

At the moment, Adobe seems to have inadvertently created an "open font foundation" (not really, thank goodness!) in the companies of Microsoft and IBM, allying either with Hewlett-Packard or Bitstream. (Off on the sidelines, Apple is working on its own system based on QuickDraw.) As we understand it, Microsoft wanted a closed system that would help it and Hewlett-Packard corner the non-Adobe, non-Apple part of the printer market, while IBM, as a

Release 1.0 21 August 1989
bit player in this market, wants an open system where it could gain share, both for its own printers and by enhancing the appeal of its platform. Meanwhile Apple has an open format for fonts and font instruction sets (a.k.a. "hints"), but the interpreter for them is available only in the Apple 7.0 operating system.

By our lights, both Apple and Adobe are appropriately declaring their proprietary interests -- Apple in the font-interpretation system, available only on its own machines, and Adobe in the fonts and font-generation tools, available only from Adobe and protected by trade secret (encryption). Apple has open fonts and a closed machine; Adobe's PostScript has closed fonts and runs on any machine.

It's up to customers and third parties to decide what they can live with. To anyone's charges that they're fragmenting the market, the obvious answer is: "Make mine the standard." As with applications, font vendors now face the chore of developing compatible fonts for a variety of platforms.

Why do fonts matter? Aside from aesthetic issues of WYSIWYG and WYGFOPIWYG-FAP (Wigfoppy-wigfap! or what you get from one printer is what you get from another printer), if different letters of a font are of different widths on different printers, you can end up with messy hyphenation and justification incompatibilities. Font building is just one more issue that involves lots of dollars, lots of users and lots of variety.

HOW IT SHOULD BE

We don't intend to get into a full-fledged discussion of the law here. The less we say the less chance we have of being dismissed as ignorant and totally wrong-headed on a topic where even educated and active lawyers disagree violently. It's not, "Are you guilty?" but, "What does guilty mean?"

Instead, we intend to offer some recommendations. Then we provide a brief background to support our recommendations and assess certain specific cases.

We believe that genuinely novel and unique intellectual property deserves strong but limited-time protection (as provided in the Constitution), by copyright or by patent as appropriate. We would support a move to reduce the period of protection to five years, perhaps based on first marketing date, a time period more appropriate to this fast-moving industry. It might also encourage companies to finish their products more fully before they set the clock ticking by publishing and marketing them.

A five-year cycle might also foster attention to customer relationships and support from companies who want to retain customers after their five years are up. It might also raise prices in the short run, but that would bring in more investment and creativity, and fundamental improvements rather than knockoffs, in the long run. It would to some extent restart the race every five years, although dominant vendors would still benefit from a critical mass of funds and expertise. Would they, in fact, steal all the good five-year-old ideas that hadn't yet made it in the marketplace? Maybe that makes sense, since someone should commercialize them if they're that good. The remaining issue is the definition of software derivative works, and whether they could qualify for succeeding five-year periods of protection.

Release 1.0 21 August 1989
Meanwhile, the patent-granting process should be speeded up with requirements for fast action on the part of both inventor and patent office, and there should be better patent database facilities (pages 14 and 20).

In the copyright area, we support broad application of the laws -- extending to specific screen designs\(^2\) and even to languages and interfaces. However, any company that encouraged use of its language, interfaces or formats would have no recourse later. Meanwhile, restricted products would be unlikely to gain a broad range of users, and thus public-policy standards/monopoly issues would not arise.

In short, vendors should be free to set their own policies regarding things they have created (or caused to be created or purchased). The only unforgivable sin (which should be forbidden) is to change the policies midstream. You cannot make something freely available to all comers and sue them later. If you want to reserve your rights, you can license something almost for free, as Adobe did, with notice that you may charge later, and let people make up their minds what that means. (To us it means that you're not sure you will continue to have the best implementation of whatever it is that you're selling.) Charging for something and then giving it away may also create problems for the third parties who paid, but that's probably an issue of contract law. Dramatically raising prices may be okay from an intellectual property standpoint, but there are other kinds of laws regarding predatory practices that might come into play.

There may be reason for some kind of licensing requirement,\(^3\) although vigorous application of antitrust laws concerning predatory competition and monopoly might take care of this very tricky issue more effectively than any statutory requirements could. Where intellectual property law ends, antitrust and other business laws may still apply. Any company that uses ownership rights to engage in predatory practices, such as using a monopoly or dominant position in one product to sell another (bundling) or pricing competitors out of the market, can be attacked on those grounds instead.

Strong protection gives people an incentive to innovate, whereas a more limited time for software than provided in current law would keep innovators moving and reduce barriers to further innovation. As noted, this structure would also encourage strong customer relationships and support as a means of differentiation, and might make companies more willing to help their customers adopt technology, since the technology won't be their edge for long.

Note: All these proposals could probably be improved upon too!

\(^2\) ...but not to the notion of overlapping windows! A particular window design with scalloped edges, however, might qualify if it's original enough.

\(^3\) How do to do that is an exercise left to the reader. It's tough because making technology available at $1 million a copy shouldn't qualify -- but where do you draw the line? The music and cable tv businesses have their own clearinghouses to allocate and distribute royalties, but it would be far more difficult in this area. Simple royalties won't work because it's difficult to determine how much of a work contributes how much to the value of the second work. Software generally isn't reused whole.

Release 1.0	21 August 1989
Aside from code, copyright protects the part of software that most users see and promotes variety. Patents protect functionality, or how things are done -- when that way is new, nonobvious and useful.

To our mind, the distinction between what copyrights cover and what patents cover is clear, yet many lawyers working in good faith do not agree. A recent survey of the Computer Software Committee of the American Intellectual Property Law Association found that the 26 respondents (out of 44 sub-committee members specializing in the field) could come to almost no consensus on any aspect of the laws. But there is no disagreement at all that the current state of confusion is unhealthy -- except maybe for the lawyers. The issue is not that strict intellectual property protection may stifle people's creativity because they can't build on others' work (as many bleeding-heart free-software proponents would have you believe), but that uncertainty over ownership and the fear of legal entanglements will deter investors (both VCs and stock-market players) and would-be developers.

The problem with copyright is that it ostensibly covers an entire work, and it may take a lawsuit to determine what is sufficiently original and copyrightable. Patents are defined upfront, but they work retroactively, and infringers may not even be aware of their existence until it's too late. So both methods suffer from (or inflict) uncertainty. But uncertainty is a fact of life, as even digitally-minded software people will be forced to recognize. (The value of a real-estate parcel or the size of a market are not guaranteed any more than the value of a piece of software.)

Patents apply only to strict copies of a defined process (which is why it's easier to see if you infringe, but harder to determine if a patent is valid), and other inventors may sell or patent improvements. Copyrights, however, apply also to derivative works -- whether copies in other forms (different language, say) or enhancements or other modifications. In either case, you need a license (or resale agreement) if your work includes the original. A patent-holder may frequently execute a cross-license with the inventor of an improved product, leaving them both free to sell the improved product. (Relative value and market factors determine the share of royalties each receives.)

Copyright: The law ain't broke; it just ain't working yet

On the copyright front, things will resolve themselves (slowly) as the copyright cases wind their way through the courts. (Ashton-Tate's suit, which may decide the copyrightability of a computer language as opposed to the "look and feel" of a product to the user, will be a landmark if it's not decided on a technicality.) Whoever suffers in the short run, everyone will benefit in the long run as the courts make clear what is protectable and what is not, and as judges become better-educated.

Some lawyers such as Pamela Samuelson, a widely published lawyer in the field, are uncomfortable with the idea of copyrighting software because of the uncertainty of its applicability, the long period of protection (50 years or more) and the long-tail way it works. Moreover, she notes, it has no doctrines for dealing with technology issues such as compatibility or sheer functionality.
Software, even user interfaces, is functional material. But while the patent/copyright line is fuzzy, we believe it's real: The two means of protection apply to different aspects of a program. First of all, we believe, copyright protects what you see on the screen, the structure and sequence of the program (how it's organized), and the actual code. These may be functional in the abstract, but the function of how you achieve an effect -- say, how you highlight a particular section of the screen, as opposed to the description or effect of the display itself thereby created -- is the kind of function protected by a patent. These are equivalent to the facts in a textbook, which are not protected by copyright, but may be protected by patent (especially if it's a technical textbook).

Of course, there may be large parts of a program not protectable by any means -- vanilla expressions of vanilla ideas using technology and structures in general use.

Vanilla vision

At this point, the general direction of the courts is unclear. "The recent decisions in the Ninth Circuit\(^4\) appear to be knowledgeable, not prone to sweeping phrases such as 'look and feel.' They're deciding cases along traditional lines: Look at the facts, see what you have to decide, and then decide that, without making speeches," says computer lawyer Esther Schachter approvingly. Thus, many court decisions are being based on other issues -- indicating a judiciary that is reluctant to set general precedents. For example, most of the recent Apple decision was based on contract law, when the judge found that the displays in Windows 2.03 were covered in the 1985 contract between Apple and Microsoft, and did not rule on whether they were covered by copyright. Open to question is the specific issue of overlapping windows -- which we consider an idea -- and some specifics of icon behavior -- which may well be copyrightable expressions.

Nonetheless, there seems to be some movement towards supporting copyright protection of software look and feel, but not in a way that unduly restricts new development (as opposed to knock-offs). Reverse-engineering (as in the Intel case and most likely in the language portion of the Ashton-Tate suit) seems to be okay.

Take a legislator to lunch...

(...preferably before you have a case pending in his jurisdiction!)

There are still a couple of things the industry can do to speed improvement the situation. One is simply to educate the courts and legislature. You can do this by talking with the ones you know, writing to the ones who represent you, speaking to the press, and getting active in your trade associations (ADAPSO and SPA). The problem with the trade associations, however, is that they can't come to agreement among their own members, and thus are

\(\text{--------}\)

\(^4\) The Ninth Circuit handles the Silicon Valley area. Recent cases there include NEC vs. Intel; Data East, which absolved a second karate game of infringement because it was based on the game rather than the first program; and Apple vs. Microsoft and Hewlett-Packard.

Release 1.0

21 August 1989
not terribly effective as lobbying forces on this issue -- although they've been effective in raising consciousness on outright duplication of software. But it's important that government officials at least understand the issues.

Make yourself clear

One obvious immediate response is for vendors themselves to provide guidelines themselves, as noted above, rather than rely on uncertain interpretation of possibly inapplicable copyright laws. Software vendors should make it clear whether they consider their products proprietary, just as they do with patents. In short, they should provide an explicit unilateral contract (not a legal notion!) to guide customer and third-party behavior and legal interpretation of their rights. (ADAPSO is currently discussing a position along these lines; see page 21. Adobe, page 4, is exemplary in this respect. By contrast, one big issue in the Ashton-Tate case is that the company encouraged third parties to use its language for many years before deciding recently that it wanted to retain that privilege for itself.) It's no fair waiting for your product to become a standard before you turn around and sue many of the very people who helped it become one.

Users and third parties have a responsibility too: They should avoid using the products of, and rewarding, vendors who haven't made a clear statement of what they consider proprietary.

FOR EXAMPLE: THE APPLE SUIT

As we've said before (see Release 1.0, 88-4), we think Apple should have tended to business rather than file a suit which it appears to have lost on the basis of a carelessly drawn contract. According to the most recent ruling by Judge Schwarzer, the contract basically allows Microsoft (and thereby Hewlett-Packard) to use anything that was in Windows 1.0 in any way and any product it pleases. Apple's argument was basically: We can't have meant that; we wouldn't have been so stupid. But if you read that contract carefully, that's basically what it says.

This leaves unresolved for now only the issue of overlapping windows and some details of how icons are labeled and move around the screen -- and of course the slightly broader issue of whether Apple would have had any claims at all even if it had not given them away.

A basic tenet of copyright law is the separation between ideas and expression of those ideas. To us, windows, even overlapping windows, are an idea, while the precise arrangement and design of scroll bars, window icons and the like, all together, might be an expression of that idea. It might seem a little silly to avoid a copyright lawsuit by moving a few icons around and changing their (visible) design, but that's what expression is. The fundamental idea is precisely what underlies the window, and the component that is not protectable.

Thus, we figure a simple solution would be for Microsoft to change a couple of pixels on a couple of icons -- and to fight fiercely for the notion of overlapping windows. Aside from being a staple of the Xerox interface (long before it was ever called Viewpoint) that's an obvious idea -- and one of two basic feasible ways of covering a screen.
However, Microsoft isn’t planning to back down, and neither is Apple. "We haven’t had any substantial settlement discussions in months," says Microsoft senior corporate attorney Bill Pope. "Apple is still trying to fight the case on the overall concept and feel. They want to get rid of the overlapping windows and I know we won’t." They will meet on September 8 to plan further actions (more meetings?). The judge has clearly indicated that it’s something the parties ought to be able to settle by themselves, but don’t hold your breath. All this smacks of a situation where the judge thinks he knows the score, but is reluctant to set a precedent until he fully understands things. We hope he tries a few more such cases and we start getting the benefit of an educated judiciary.

Xerox’s Viewpoint: A "safe haven"

Meanwhile, the granddaddy of them all, Xerox, explicitly views itself as offering a "safe haven" for developers who are uncertain of their rights, says spokesman Peter Hawes. Xerox has signed licenses for the use of its Viewpoint interface so far with Sun and with Metaphor. Those rights extend to (and royalties are payable by) AT&T and IBM, among others, when they use OpenLook or the Metaphor interface. The royalties are "reasonable but not nominal," says Hawes. Which is as it should be. The only argument we have with Xerox is that it didn’t act sooner, but at least it seems to be acting with appropriate restraint.

Meanwhile, Xerox, a company founded around a patent, is still talking to other people, presumably including Apple. Should Apple pay royalties to Xerox, it would strengthen its moral case for ownership of its interface -- even if that ownership went partially to Xerox. Apple would lose the battle against Microsoft, but it might win the war of principle.

A brief comment: The Lotus suit

The Lotus suit strikes us as a clear case against some rip-offs. A couple of companies said in effect, "We’re going to trade on the popularity of the 1-2-3 spreadsheet, and charge less." Neither of these vendors made enough performance improvements (not a copyright defense anyway) to trade on much other than price, and to rise out of the category of clearly derivative works. We prefer the approach of Microsoft, which genuinely advanced the state of the art with Excel (and hasn’t been sued for it).
HOW IT IS NOW: INTERFACES AND LANGUAGES

Despite the particulars of the Microsoft-Apple case, Apple probably does have the right to protect the Macintosh interface: The only issue is how to define it. On the other hand, it may not be able to prevent clones, an increasingly prevalent notion. Why?

In one case you're imitating the user interface, and in the other the program interface. At this point, we believe the law allows hardware/program interface clones, but is less hospitable to user-interface clones, although court interpretations vary. Hardware and program-interface clones are allowable as reverse-engineering of functional systems, as suggested in the recent Intel case. In these cases, the copiable functional "idea" is the ability to handle program calls, while the look and feel of user interfaces is a visual display that can be protected by copyright.

We feel that the law should be consistent and allow the protection of both (although we describe below our understanding of the current interpretation of the law). Nonetheless, as a business practice any vendor who wants to establish a product as a standard would be well-advised to license it freely -- notwithstanding Apple's success with a contrary policy.

Interfaces

Hardware interfaces, program interfaces and languages are fundamentally the same thing. They are just layers; the hardware is the one at the bottom (in common understanding). All receive electronic input and respond appropriately: whether by compiling a program or executing a sequence of actions -- everything from displaying a menu to performing a series of complex calculations on a set of data meeting certain criteria. Copyright or patents or trade secrets (if you never let out the source code) protect the code and specific algorithms that accomplish these functions, but what about the set of functions that constitutes the application or the language? What's wrong about responding in a predictable way to an expression written by the customer? In other words, it's the customer who's using the language putatively "owned" by Vendor One; Vendor Two is only interpreting or compiling it.

So now the question is, What if you never publish the specs of the machine or language? In essence, that would mean that you would sell only complete systems, not allowing users or third parties to modify or enhance your product except under strong, explicit contracts. (This is what Adobe is doing with its font technology.) We don't see anything inherently wrong with that morally or legally, but as noted, in many cases it's a dumb business move.

Stroke and feel

What about sequences of keystrokes, or feel -- which amount to languages in many cases? That gets more troublesome, and is probably the area where most of the controversy and divergent court opinions will arise. The ability to use the same keystroke sequences is important to users, who don't want to waste time retraining their fingers. But in the long run it's a convenience they should be willing to pay for.

In the short run, cases such as Lotus's or Ashton-Tate's against Fox (see Release 1.0, 88-12) will probably be decided on other issues, such as simi-
larity of look, which is clear in both cases. Our vague feeling is that it's okay to provide similar keystroke sequences as a convenience to users in an otherwise substantially different product, but questionable if the entire product accomplishes the same task and looks the same as the original. Thus, it's less of a problem to make your graphics package work like 1-2-3 than to make a spreadsheet do so. Obviously, these are rules of thumb, and liable to inconsistencies (which is why we need some clearly articulated guidelines going forward). What if Lotus's graphics package also works like 1-2-3? Does that mean you can't use those keystrokes after all, or on the contrary that they have now become a standard (whatever that is) that you may use freely?

Reverse-engineering: For example, the Mac

Reverse-engineering means using the original as a spec. In general, the current reading is that a reverse-engineered implementation is an expression of the idea of executing the commands interpretable by the original. As long as you develop the system to execute them independently, you're okay. This is the reasoning underlying the common practice of reverse-engineering chips, BIOSes and other hardware. An interesting notion we hear from time to time is that of making a Macintosh clone that can run Mac software. It would have a different look, of course, but much the same feel. It would still take mouse input, but its trash can might be a shredder, a black hole or an image of our office, and its scroll bars might work differently.

But the clones would "look" identical to the applications that ran on it. In other words, the user can adjust to a couple of minor differences, but the applications can't -- as many vendors of "almost-compatible" PCs found out in the early Eighties. (Remember the Dot, the Sirius/Victor, et al.?) Would such a machine be legal? Probably. Would it make business sense? Probably. Could someone do it? Apple thinks not without peeking at proprietary material, and copying instead of reverse-engineering. We're not sure.

However, we couldn't find much information about the rumored clones. Their products don't seem to be out yet (if ever) and thus can't be assessed.

Mac "clones": Old ROMs in new boxes

Right now the Mac "clones" one reads about aren't really clones. They're generally systems that use Mac ROMs or boards in new boxes. At this point there's not much Apple can do about them, other than to sue dealers who break their contracts by taking machines apart. "Mac ROMs belong in Mac products," says Apple spokesperson Carleen LeVasseur. Even switching a customer's ROMs at a customer's request "raises serious questions about patents and copyrights." (We're not sure what they are.)

5 We don't necessarily agree with this situation and believe companies such as Apple should have the right to stop clones (and Xerox should have the right to require licensing of Viewpoint). Nonetheless, as a business decision, we further believe that Apple should license clones and expand its customer base. Raise your hands if you buy (or support) Macs in the hope that there will be successful clones.
So far, Apple has done nothing, probably because the practice isn't widespread. Colby, for example, has shipped only a few hundred systems, and the other vendors aren't much in evidence. (Colby buys an entire board from dealers and assembles the system itself. Dynamac, which makes a licensed Mac portable, buys directly from Apple under a VAR agreement and sells its systems for a market-protecting $6,000 to $10,000.)

Yet Apple can still make life difficult simply by scaring off buyers and dealers as well as "clone" vendors. For the moment, it is merely making disturbed noises. It's "premature" to comment on the most promising newcomer, the Wallaby, says Jean-Louis Gassee.

But as a business decision, Apple might do well to welcome these pseudo-clones (to say nothing of real ones). Wallaby, for example, has built a stunning $3000, 9-pound laptop system (we saw a working prototype off the floor at MacWorld) that uses the ROM of a Mac Plus or SE. This system has several benefits for Apple if it could only see them: It gives people something to do with their old Macs and indirectly encourages them to trade up to Mac IIs. It opens up a whole new customer base of traveling salespeople and other itinerants to the Mac. And it spreads the Mac gospel. (Moreover, it needs a Plus or SE for initialization, so it can't use pirated ROMs.)

Meanwhile, the biggest complaint we hear about the Mac from developers and end-users is the lack of a second source. That's partly a complaint about price, but it also concerns market share and Apple's perceived attitude.) These same arguments would also apply to true clones, minus the how-to-make-your-old-Mac-born-again point.

Look ma! No machine!

Another approach, probably equally irritating to Apple, is that of Bawamba Software, with its Multiplatform Compatibility Package development tool. In essence, Bawamba is replacing the Mac program interface and subroutines with one of its own -- but there's no machine there, and no Mac-like user interface. Instead of a machine sold to users that interprets third-party Mac programs and presumably provides a Mac-like OS for a user to interact with, Bawamba sells a tool for developers that makes it easy for them to port their Mac programs to other machines (or vice versa).

It's a library of over 600 subroutines that allows a builder-user to compile and link C source code and port it from the Mac to the pc (as long as the Mac programs are as "clean" and hardware-independent as the kind of PC programs that could run under the original Windows). The resulting program retains most of the functionality of the Mac program. The original version of MCP, then called the Macintosh Compatibility Package, generated programs with the Mac look and feel as well as the Mac functionality, but conversations with Apple's lawyers and customer aversion to uncertainty prompted developer Steve Greenfield to switch to OpenLook -- a gain for Sun and a loss for Apple. (Note, however, that MCP ports code just to DOS and OS/2; a 386 UNIX version is coming soon.)

This all makes the point of how trivial look and feel as a technological issue -- as important as it is to users' experience and a program's appeal. With the new generation of development tools, look and feel can be transformed automatically, unlike a program's underlying functionality.

Release 1.0 21 August 1989
The Patent and Trademark Office de facto provides a registry of novel, useful technology with the purpose of rewarding inventors and preventing redundant invention of wheels and other existing technologies. The problem with patents from the software perspective is that they take too long to grant -- two to three years versus an average of 19 months for all patents, according to computer lawyer Gary Hecker. Moreover, because it's hard to find and filter all relevant previous patents and other "prior art," it's difficult to know when you deserve a patent for a truly novel invention, or when you're infringing, even inadvertently. Many patents granted (about half of those challenged, says Hecker) don't stand up in court, but most alleged infringers find it too costly to contest them and simply pay a protection -- er, license fee. In software, where patents and cases are too few to provide a good sample, the quality of patents may well be lower, since patent examiners have limited experience with software.

Nonetheless, it is misleading to liken (even figuratively) the software patent situation to the possibility of patenting the first four notes of Beethoven's Fifth Symphony, as The New York Times did recently. Aside from being nonfunctional, they would have insufficient novelty, content or nonobviousness to merit a patent. Yet there certainly is room for some tangible government-funded improvements to the process and substance of the system that could substantially reduce unnecessary uncertainty due to inefficiency and inaccessibility of information.

- First of all, only patents with some likelihood of being upheld should be granted. Patent claimants have to negotiate a trade-off between breadth and acceptability: Claim too much and you'll get nothing at all, because your claims won't be upheld (either by the patent office or later on by a court when they're challenged). The patent examiner should be sure that the trade-off is tight.

- Second, it should be easy to find out whether a particular idea has been patented.

- Third, patents should be granted quickly or not at all.

Achieving the first two is difficult, since it requires a clear statement of the problem and of the solution (and it may be a solution to some other problem as well), but it should be possible to create a meaningful, automatically updating electronic database that would help in this process.

Indeed, it's a perfect application for some of those text tools we keep writing about (and for which the Patent Office itself is now issuing patents). We can imagine many tactics beyond the Office's current move towards a full-text electronic database. For example, you could build an interesting citation index, linking patents to each other by citations and allowing examiners to follow those links easily. Alternatively, you could use similarity rankings to match a patent application to other patents on file. Thus your query would consist of all the words (minus stop words) in the application instead of a few key words. And some day the examiners and outsiders might have better access to other sources of information. (Right now, this is a special problem in software since there are so few software patents, but a considerable body of published prior art.) See Release 1.0, 87-9, 88-1, 89-3, 89-7, among others; and also page 20 of this issue.
The patent office in (slow) motion

The typical time for a patent to issue is about 19 months, including about two weeks for the filing to be logged, copied and distributed to an examiner, three months for the applicant to respond to the patent examiner's initial findings, and a couple of months for the documents to be printed. The Patent and Trademark Office's goal is to reduce that to 18 months, says internal information systems expert Lee Skil-lington. That strikes us as underly ambitious.

The Office could benefit from an automated paperflow system such as FileNet's or the many modular systems such as IBM's and Wang's. The actual time an examiner spends on a single patent application may be only 15 to 30 hours; much of the rest is moving from queue to queue. Everything moves on paper: The application comes in on paper, and its folder gradually grows as related letters, other patents, examiner's comments (dictated and typed) are added.

The Office currently has a tracking system into which Assistant Commissioner Tom Giammo hopes to fold more and more of the process, ultimately allowing for electronic filing and shuffling the application through the entire process electronically: "Everything an examiner did would be controlled within a software shell and logged in. You could collect statistics on where things are, and assemble all the documents and letters and change. Then at the end something would plop out that we could go directly to photocomposition with."

While the Patent Office has installed an electronic full-text 30-plus-gigabyte database of all 1 million-plus patents issued since 1975, the examiners usually use it only to search for previous relevant patents in assessing an application for a new one. Once they find them by Boolean and proximity search (and criteria such as date, author, assignee, number and classification), the examiners pull the paper files. Subsequent events -- challenges, litigation, amendments and the like -- aren't necessarily appended to the main document but are filed separately. Patents are classified into any of 112,000 categories based on the technology they implement, plus cross-references. The classification scheme is strictly hierarchical and lacks the richness or flexibility of some alternate scheme such as a citation index. The Office has about 100 terminals for its 1500 examiners, plus four for the public. (That number is limited so that the Office will not compete with private enterprise, and in fact the same database, with a few enhancements but the same data, is available to the public from Mead Data Central as Lexpat.)

Obviously, better information aces should reduce the number of patents filed and granted erroneously and speed up the rest, and also make it easier for filers and third parties who want to avoid infringement to assess the situation accurately and act with dispatch.

The Office's overall budget for its Automated Patents System -- a series of improvements expected to be completed by the year 2000 -- is running at about $30 million a year currently. We're not a big fan of government spending, but additional funds here might be a wise investment rather than a cost.
To be sure, the Patent Office is aware of the potential of technology. "Our premise is that there are better ways of doing it than we're doing it now," says Assistant Commissioner of Information Systems Tom Giammo. (See box.) But there will be no fundamental changes to the simple text-search system now in place until at least 1995, when the current project to get all the diagrams online is finished. The number of software patents granted so far is still relatively small, so now's the time to start.

Train a patent officer

The solution to the speed problem too lies partially in technology, as described above. But it also needs a larger, better-educated, faster-acting staff to use the tools. Its current staff of about 1500 examiners is ill-equipped to handle the coming boom in software-based intellectual property -- our strategic advantage in the Nineties. In all there are about 100 patent examiners devoted to "information storage and retrieval," one of 16 groups, but few of them have a deep software background.

True, some vendors prefer not to have their patents granted right away (why alert people early?), but speedy action is of benefit to the industry as a whole. Applicant response times could also be shortened.

Overall, much the same problems arose in the early Eighties in the biotech business, Skillington points out, and they were ultimately resolved as a new class of biotech-trained examiners entered the system, and as experience built up. The difference is that biotech products have long leadtimes and long lifetimes; software is ephemeral, decomposable and copyable in every sense of the word. When your product is ready (if not before), you ship; there are no factories to build, and there are lots of substitutes. As pointed out below, the advent of modular software will mean a quicker time-to-market for software components.

PATENTS ON PARADE

Although companies such as IBM and Xerox have been obtaining and licensing patents for years, software patents are relatively new. The Diamond vs. Diehr case in 1981 was the first broad acknowledgement that software could be patented. Two patents recently awarded to Advanced Software Inc. and Quarterdeck Office Systems have made the headlines, since they refer to broadly used capabilities -- text comparison (redlining) and window management. Even broader (but more questionable in its claims, we think) is Refac International's patent-infringement lawsuit against spreadsheet vendors.

Advanced Software Inc. and Quarterdeck

This spring Advanced Software won a patent for the methods used in its text-comparison software, which originated in inventor Cary Queen's work on comparing gene sequences. We don't claim to know the merits of the patent, but the folks at Advanced Software seem eager to avoid litigation in favor of license fees from companies such as CompareRite vendor JURISoft, Lotus and WordPerfect. Complicating matters is JURISoft's own pending patent on what looks to be similar technology. Maybe they could cross-license each other and cancel things out. This sounds overly cute, perhaps, but it's the way things often happen when companies have business, not law, on their minds. (You can get a patent on a specific implementation of the ideas in someone...
else's valid patent.) As for the other, bigger guys ASI is talking to, they'll probably try to wait and see how the smaller guys sort it out.

Meanwhile, Quarterdeck has won a farther-reaching patent covering a variety of approaches to building and operating display-dependent applications in multiple windows concurrently. The patent is fairly broad, and Quarterdeck is actively talking to potential licensees such as Microsoft. Basically, Quarterdeck was surprised when it heard of the patent award; it filed and amended it long ago as a routine business practice (stemming from founder Terry Myers' background at a Citicorp subsidiary) and has been tending to business since. Yet the prospect of patent revenues for this company that has been living hand to mouth for so long a time is exciting.

This story counters the incentive argument, in that Quarterdeck wasn't lured to create its product by the promise of patent revenues; on the other hand, it certainly intended to protect its rights to its work, and it is a small guy fighting a big guy with superior market power over which it would welcome a moral (and financial) victory. If Quarterdeck does make it big, its example will help other vendors get venture capital, and will lure more developers away from cushy corporate jobs to try something new. Success stories have an incentive effect; it's all in the statistics.

Refac International Ltd.

Refac International has just filed suit against Lotus, Borland, Ashton-Tate, Computer Associates, Microsoft and Informix. Refac is in the business of acquiring and litigating patent rights -- perhaps not a practice to endear it to the software industry but a legitimate pursuit nonetheless. In other circumstances, it might be considered a benefactor, rewarding the creative efforts of a small company that couldn't afford to assert its rights against giants such as those named. In this case, however, the patent it acquired is of questionable provenance and value. It was first filed in 1970, and refused by the patent office, which at that point felt some hostility to the notion of software patents and probably didn't assess the application on its merits before denying it. The owners, a Canadian company called Forward Reference Systems Ltd., appealed the rejection and argued their own case sans lawyers. The Patent Office evidently didn't take the appeal seriously and rejected it again on the basis of its not being patentable subject matter, without bothering to give grounds such as prior art -- which of course it couldn't find in its patent database because there were few software patents at that time. In fact, the Diehr case which set the precedent for the patentability of software was decided 3 March 1981 -- just before the Patent Office Board of Appeals' rejection of the "spreadsheet" patent on 20 May 1981. Eventually, the Court of Customs and Patent Appeals remanded it for reconsideration to the Patent Office, which granted the patent in 1983. Refac acquired it last month.

As it happens, according to interested observer Dan Bricklin (who should know), there was indeed substantial prior art, even in 1970. According to Dan Bricklin, author of VisiCalc, the first mass-market spreadsheet in 1979, the Refac patent refers to a capability (now called natural order of recalc) that had common-knowledge implementations at the time the patent was filed in 1970. It refers specifically to only one algorithm, which he believes is not the one used by Lotus, but its overall claims are much broader. The issue is probably the validity of those broad claims rather than their applicability to 1-2-3.

Release 1.0

21 August 1989
ISSUES OF THE FUTURE

Once the country has resolved the issues of look-and-feel and interfaces, it's still not likely to suffer from hordes of unemployed intellectual property lawyers, because technology advances will provide them with two new problems to address -- on-line information packaging and distribution services, and reusable code in the form of subroutine and class libraries.

INFORMATION PACKAGING AND DISTRIBUTION

With some exceptions such as Dow Jones News Retrieval, most on-line services are packagers and retailers of information generated by someone else. They are charging in part for the information, but also for the service of finding, filtering and delivering it. The clearest illustration of this is the dissemination of public-domain information, which you can get from the government "free" -- i.e. for their cost and your time.

The information distributor's work consists of two primary functions. First he must line up the appropriate information sources and negotiate contracts with each of them. He must also charge the users appropriately. Second, he must provide an interface and tools that enable the users to find precisely the information they need.

Pricing

This part of the job involves negotiating contracts with each information supplier, and determining the appropriate pricing. Frequently on-line distribution will be only one means of dissemination of the information. The biggest problem is simply defining the product and the unit of measure. Is it connect time, access, etc.? What kind of use is to be made of the information? Can one corporate customer share it with the guy down the hall, his entire department, his entire company, his customers? And how can you control this?

Then there's deciding how to charge: Does the information supplier get an overall royalty, or is it feasible to charge per access? Or should you just sell it on a CD ROM and forget the details? What extras such as fancy filtering and presentation can you charge for? What's the value of a database access where the response is "Nothing found"? That could be useful information in a reference check where you were checking for negative information.

The same information can be resold many ways and many times -- frequently to the same customers. You can get data in real-time, in time-series, and integrated with other data in statistical analyses.

----------

6 The originator's first business question is: Will on-line cut into my print revenues, or will it in fact serve as a marketing vehicle that will attract users to other forms as well? In other words, does it satisfy demand or create demand? Varying proportions of both, of course. As a rule of thumb, we assume that it satisfies demand for timely information, such as stock quotes, and creates demand for long-term information, such as Release 1.0, by pointing out information they might have missed, or might not have needed when they saw it, but want to find now.

Release 1.0

21 August 1989
From Juan to Alice to Fred to Bill

And it can be reused in many ways. Who owns it, and what interest does he retain, after it's been filtered, massaged and turned into yet another database? Can a lawyer send out a fax of a patent he downloaded from Lexpat? What about the same patent he received from the Government Printing Office? All these questions will keep lawyers occupied for years.

Textual data raises other problems. What's the legal status of abstracting? For now it's legal, but what happens when we move towards automatic abstracting, assignment of key words and the like? Is it really okay to buy The New York Times or PC Week, say, for $100 or so a year, and resell summaries for hundreds of dollars?

More interesting in the long run is the notion of assembled texts: Users may end up reading only paragraphs here and there, following links set up by a human editor or an intelligent agent. Do we really want to pay royalties by the paragraph? Maybe we do, since it will be relatively easy to track usage automatically (see Release 1.0, 89-7). But how much goes to the original author (and his publisher) and how much to the person who -- or owner of the agent which -- created the links? That of course is for the market to determine, and it will be interesting to see how it shakes out.

Packaging

In essence, we're going back to the world of data-sharing instead of time-sharing. The reason for time-sharing was that it made more sense (in an age of large machines) to keep the machines centralized and sell their services. Now there are small machines, and everyone can afford the equivalent of what used to a large machine on his desk. But it still makes sense to keep the data centralized, so that it can be filtered and only the results of queries need be sent over still-narrow wires to the final customers. This is partly a way to monitor usage, but it also enables the data to be mashed and grinded centrally by huge number- and text-crunchers that even nowadays are be too expensive to deploy locally. After the text has been categorized and linked, then individual users will be able to follow specific typed links or queries, and get what they want.

The new model is client-server architecture, where both ends of the system are intelligent. Once the data is shaped and filtered for the customer, it's the job of the client workstation to make the resulting subset accessible to users for incorporation into reports or further analysis. Client software (such as IZE, Agenda or Topic, among others) lets the user generate queries, specify filters and manipulate the text and data he downloads.

REUSABLE CODE

Reusability is a fundamental principle of object-oriented computing. It also underlies the notion of dynamic link libraries, modular programs and interprocess communication found in OS/2 and UNIX (among others). If applications can talk to each other, they can use each other's services instead of doing everything themselves. Each developer does only the parts he can do best, and de facto reuses other people's modules for the rest.

As these approaches become fashionable, they will lead to a large secondary market in class libraries and application components. What kinds of con-

Release 1.0

21 August 1989
tracts will govern their reuse? And as they get modified and resold again and again, who gets paid and in what proportion?

In the end, who will make the investment to turn a module into a standard -- a public good the existence of which benefits us all? We believe the market -- with its trial-and-error, risk-and-incentive mechanisms effected through the application of intellectual property law -- is by far the most effective approach. In a world without incentives based on some notion of intellectual property, we can't see what forces would propel the creation of good products and of standards. Vendors would be unable to leverage their products, and would probably instead sell professional services or black boxes, all the while endeavoring to keep their techniques and technology secret.

Perhaps this market will work through the (not yet funded nor even proposed) National Software Registry and Licensing Bureau, a huge object-oriented transaction database, a sort of Library of Congress-cum-stock market of functional intellectual property. Of course, software is not as fungible as stocks are, but that's why it's an interesting technical challenge. (See Release 1.0, 89-6, "SQL and prices," page 19.)

POSTSCRIPT: SOVIET UNION AGAIN

Our visit to the Soviet Union recalls the story of the fish who climbed onto land and thereby discovered what water was all about. So it may be instructive to relate what it's like in a world without private intellectual property (or much of any other kind, either). People do have some incentive to create software, because the institutes and factories and offices where they (pretend to) work (pretend to) pay them to (pretend to) do so.

But once that code is created, it generally languishes. No one advertises it, improves on it, or resells it. No one supports it or writes manuals about it. No one proclaims its benefits. There's no reason to. Meanwhile, because there's no incentive to spread information about any new inventions in the software (an integral part of a market system), developers don't learn from each other. Sure, people trade programs and there's some interaction, but the free flow of free information moves slowly without market channels to flow in.

COMING SOON

- CompuServe, Prodigy, MCI Mail, USENET, Internet, et al. The only way we know to get around to getting online is to commit ourselves to writing about them...
- Network navigation.
- Transaction processing.
- Object-oriented database status report.
- And much more... (If you know of any good examples of the categories listed above, please let us know.)

Release 1.0
21 August 1989
RESOURCES & PHONE NUMBERS

Ron Palenski, ADAPSO, (703) 522-5055
Larry Lightman, Jeff Cherniss, Advanced Software Inc., (408) 733-0745
John Warnock, Colleen Poulion, Adobe Systems, (415) 961-4400
Carleen LeVasseur, Jean-Louis Gasse, Apple Computer, (408) 996-1010
Steven Greenfield, Bawamba Software, (818) 843-1627
Esther Roditti Schachter, Computer Law and Tax Report, (212) 758-1464
Britt Blaser, Dynamac, (303) 296-0606 or (800) 234-2349
Paul Goodman, Elias & Goodman (representing Refac), (212) 421-6000
Pamela Samuelson, Emory University, (404) 727-6821
Gary Hecker, Hecker & Harriman (representing Quarterdeck and Advanced Software), (213) 286-0377
Richard Anders, JURISOFT, (617) 864-6151
Michelle Love, Taylor Putney, Mead Data Central, (513) 865-6800
Lee Skillington, Thomas Giammo, Patent & Trademark Office, (703) 557-6000
Philip Sperber, Refac International, (212) 687-4741
Dan Bricklin, Software Garden, (617) 332-2240
Ken Wasch, Software Publishers Association, (202) 452-1600
Mike Higgins, Wallaby, (303) 444-4606
Peter Hawes, Xerox, (203) 968-4416

For further reading:

"The Computer Lawyer," special issue on software patents, May 1989, and Intel case discussion, March 1989, Prentice-Hall Law & Business, (201) 894-8538. This is a useful, solid publication in general (we sit on its board), and these issues are relevant in particular.

"Computer Law & Tax Report," edited by Esther Roditti Schachter, (212) 758-1464. While The Computer Lawyer is for lawyers, and runs to 48 pages per issue, CL&TR is for business people who are at the mercy of lawyers, and provides the news and the implications in half that number of pages of focused reporting.

"Software Industry Survey," by Price Waterhouse. Sponsored by the Massachusetts Computer Software Council, c/o Joyce Plotkin, (617) 437-0600. How, not how much. This report focuses on internal information -- how companies do things such as protect their software, what tools, languages and operating systems they use -- rather than predictions of market share. We'd like to see the same information for the other half of the industry in Silicon Valley. (And the third half elsewhere!)


Exposure draft of "ADAPSO Position Paper on Business Guidelines for Declaring or Waiving Proprietary Rights in Interfaces and Languages," ADAPSO, (703) 522-5055. This is a somewhat vanilla document that makes sense; i.e. it agrees with our position. Note that it has not been adopted by ADAPSO, but it has given the group a specific, reasoned point of view to focus on in an area of controversy and blather. Most inflammatory is its position in favor of copyright on "sufficiently original" languages and interfaces, which may well be watered down. Read it and comment!
RELEASE 1.0 CALENDAR

August 24-September 1 *Eleventh World Computer Congress - San Francisco. With a focus on tools and application software this year; in the U.S. for the first time in 24 years. Sponsored by 46 IFIP member societies. Call Nancy Dana, (303) 696-6100.


September 6-9 Breakaway '89 - Orlando. Sponsored by ABCD, the microcomputer industry association (mostly dealers). Keynote: Fran Tarkenton. Contact: Deborah Keating, (601) 977-9033.

September 7-10 Comtec '89 - Singapore. Regional micro exhibition. Sponsored by Microcomputer Trade Association of Singapore. Contact: Yong Mee Hiong, Singapore 2913238; fax 2965384.


September 11-13 Patent, trademark and computer litigation - Boston. "Trial techniques and strategies." Sponsored by Franklin Pierce Law Center. (Who should attend? "Any litigator whose competitor has already registered..." says the brochure.) Contact: Jamie Bulen, (603) 228-1541.

September 11-14 NetWorld - Dallas. Keynote by telecommunications vp William Hider of Gannett. Managed by H.A. Bruno. Contact: Adam Torres at (201) 569-8542 or (800) 444-EXPO.


September 13-15 Conference on Computer-Supported Cooperative Work - London (Gatwick). Inspired by the successful U.S. events, but likely to focus even more on social issues. Contact: Lorna Meek, 011 44 (753) 73232.

September 17-21 Managing the corporate personality for the nineties - Martha's Vineyard. Sponsored by Design Management Institute. With speakers from Xerox and Danish State Railways, among others. A broadening, useful conference. Call Nancy Barry, (617) 236-4165.

September 18-20 DataStorage - San Jose. The industry standard, sponsored by DISK/TREND and Freeman Associates. Storage issues, from massive disk farms to notebook systems. Call Darlene Flamondon, (408) 554-6644.

Release 1.0 21 August 1989

September 19 Interface Design '89 - San Francisco. Sponsored by Mac-Week, in hopes of consistency if not necessarily standards. Contact: Cindy Koral at (415) 243-3315.


September 19-22 Ashton-Tate Developer Conference - Anaheim, CA. With subconferences on Framework and Mac products. Call Brad Stevens, (213) 538-7348 or (800) 223-4898.

September 20 #Charles Wang at New York PC User Group - New York City. Call David Hoffman, (212) 674-2632, or (212) 533-NYPC.


September 21 #Microcomputer Managers Association meeting - New York City. Keynote by Esther Dyson. Contact: Priscilla Tate-Austin, (201) 580-9091.


Release 1.0 21 August 1989
<table>
<thead>
<tr>
<th>Date</th>
<th>Event Description</th>
<th>Contact Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>October 1-4</td>
<td>*ADAPSO Management Conference - Orlando. Mingle with your peers (and Disneyland's nearby just in case). With Alan Kay (Apple) and Robert Weissman (D&amp;B), among others. Contact: Sheila Wakefield, (703) 522-5055.</td>
<td></td>
</tr>
<tr>
<td>October 1-4</td>
<td>Alex. Brown Computer Services Seminar - Baltimore. The tenth annual... Contact: Rivka Hawk or Ellen Kempler, (301) 727-1700.</td>
<td></td>
</tr>
<tr>
<td>October 2-6</td>
<td>*OOPSIA - New Orleans. Sponsored by ACM/SIGPLAN. Come meet your fellow objects and share procedures. Send a message to Carole Mann, (407) 628-3602.</td>
<td></td>
</tr>
<tr>
<td>October 2-6</td>
<td>CD-ROM Expo - Washington, DC. Sponsored by IDG Conference Group. Contact: Dorothy Ferriter, (508) 879-6700 (registration), or Richard Winant, (617) 329-8090 (exhibits).</td>
<td></td>
</tr>
<tr>
<td>October 2-6</td>
<td>Interop '89 - San Jose. Interoperability made tangible, with tutorials, discussions, product demos and pitches, and speeches by Doug Engelbart and Vint Cerf of Corporation for National Research Initiatives. Sponsored by Advanced Computing Environments. Call Mark Belinsky, (415) 941-3399.</td>
<td></td>
</tr>
<tr>
<td>October 3-5</td>
<td>PC Expo - Chicago. Sponsored by PC Expo. Contact: Steven Faher, (800) 444-EXPO or (201) 569-8542.</td>
<td></td>
</tr>
<tr>
<td>October 5</td>
<td>The R&amp;D gold mine - Palo Alto. Sponsored by Regis McKenna. &quot;Improving your effectiveness in using nonprofit research organizations as a source of new products and technologies.&quot; Contact: Elizabeth Batson at (415) 857-9388.</td>
<td></td>
</tr>
<tr>
<td>October 5-6</td>
<td>Electronic messaging '89 - Chicago. Sponsored by Electronic Mail Association. With Warren Prince, Tymnet; Mike Zisman, Soft·Switch; others. EDI, X.400, and connected topics. Contact EMA at: telephone, (703) 522-7111; fax, (703) 528-4251; AT&amp;T Mail, !EMA; Dialcom, 63:PRD003; EasyLink, 62886257; jNet, ema.association; CompuServe, 70007,2377; Envoy 100, EMA; GEnie, EMA; On Tyme, EMA.SUP; MCI Mail, EMA/2544290; Telemail, [ema/associates] mail/usa.</td>
<td></td>
</tr>
</tbody>
</table>
October 9-12  ShowCASE IV - St. Louis. "Integrating automated development environments." Tutorials and speeches, with Ed Yourdon, Vaughan Merlyn, Larry Constantine, Capers Jones, Sam Hocman, others. Sponsored by the CASE Studies Consortium. Call Donna Skaggs or Kimberly Yourick at (314) 889-4556.]

October 10  *Massachusetts Computer Software Council fall meeting - Newton, MA. With a panel on "Object-oriented Everything" moderated by Esther Dyson and featuring Tom Atwood, Object Design; Tom Malone, MIT; and Burt Rubenstein, Index Technology. Also, discussion of membership survey results; see page 3. Call Joyce Plotkin, (617) 437-0600.


October 11-13  *Strategic directions in computing research - Washington, DC. Sponsored by ACM. Call Donna Baglio, (212) 869-7440.


October 14-21  International Computer Forum - Venice, Italy (the real one). Sponsored by Boston Computer Society. Seminars on desktop publishing, support strategies, computers in investing, etc. Call Beverly Kleiman, (617) 367-8080.


October 16-19  Scan-Tech 89 - San Jose. Sponsored by Automatic Identification Manufacturers. On beyond retail bar codes, including integration with EDI, tracking materials in offices, etc. Scan-Talk: Munster punster (and Wall Street Week host) Lou Rukeyser. Call Bill Hakanson, (412) 963-8588 or (800) 338-0206.

October 18  *Rod Canion at New York PC User Group - New York City. Call David Hoffman, (212) 674-2632, or (212) 533-NYPC.

October 20-21  *A delicate balance: Technics, culture and consequences - Los Angeles. Some thought-provoking topics, but will the right people be listening? Call Chantal Toporow, (213) 813-6194 or Nik Warren, (213) 392-6595.


Release 1.0  21 August 1989
October 23-26  Monterey Classic - Monterey. Sponsored by American Elec-
tronics Association. Where VCs, investors and enticing young companies can meet. Contact: (408) 987-4200.

October 23-26  Hammer Forum - Boston. "Reengineering the corporation: Information technology and business process redesign." With Allen Loren, Apple; Paul Chapman, Rank Xerox; and Michael Hammer himself. Contact: Pam Davis, (617) 354-5555.


October 26  The R&D gold mine - Ann Arbor. Sponsored by Regis McKenna. "Improving your effectiveness in using nonprofit research organizations as a source of new products and technologies." Contact: Elizabeth Batson at (415) 857-9388.


October 29-November 1  *3Com network systems forum - San Jose. Sponsored by 3Com. Contact: Cheryl Soderberg, (800) NET-3Com or (408) 562-6400.

October 30-November 1  *Seventh annual Seybold Executive Forum - Boston. Sponsored by Patty Seybold's Office Computing Group. Contact: Deborah Hay, (617) 742-5200 or (800) 826-2424.

November 1-3  *UNIX expo - New York City. Keynote by noted UNIX fan Ken Olsen of DEC; also speaking: Bill Joy and John White. Managed by National Expositions Co. Contact: Roger Haltigan or Heidi Dethloff, (312) 332-4650 or (212) 391-9111.

November 5-10  *Hypertext '89/SIGDOC 89 - Pittsburgh, PA. Much larger, for better or worse, than the first, wonderful hypertext conference in the fall of '87. Hypertext covers the first three days; SIGDOC the last three. Sponsored by ACM. Contact: Elise Yoder at (412) 327-8181 for Hypertext '89; Nina Wishbow at (412) 323-2600. (How about a joint committee on standards for the use of apostrophes?)

November 13-15  UIST - Williamsburg, VA. Symposium on user interface software and technology, sponsored by ACM SIGGRAPH and SIGCHI. Contact: John Sibert, (202) 994-4953.

November 13-17  *Comdex - Las Vegas. Also including MACdex. Contact: Jane Wemyss at (617) 449-6600 or (800) 325-3330.

November 13-17  Supercomputing '89 - Reno, NV. Conveniently located near Comdex, if you tire of small computers and big hoopla. Contact: Ron Bailey, (415) 694-4500.


December 4-6  *First international conference on object-oriented and deductive databases - Kyoto. Sponsored by IEEE, MCC, many others. Contact: Professor Kiyoshi Agusa, 011 (81 75) 256-1677, or Won Kim at MCC, (512) 338-3439.

1990


January 28-31  **EDventure Holdings PC (Platforms for Computing) Forum - Tucson, AZ. Sponsored by us! New speakers will include Danny Hillis, Thinking Machines; Mike Slater, Microprocessor Report; Rod Canion is returning. Note that it's earlier this year. Contact: Daphne Kis, (212) 758-3434.

February 6-9  Software development '90 - Oakland, CA. Solid information from practitioners and luminaries, including Larry Constantine, Ed Yourdon, Ken Orr, Bill Gates and Philippe Kahn. Sponsored by Miller-Freeman, publisher of Computer Language, AI Expert, UNIX Review and other magazines. Contact: Cheryl How, (415) 995-2471.


March 5-9  *Seybold Seminars '90 - Boston. ...moves east. Call Kevin Howard, (213) 457-5850.

Please let us know about any other events we should include.
-- Denise DuBois

*The asterisks indicate events we plan to attend. Lack of an asterisk is no indication of lack of merit.

Release 1.0  21 August 1989
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.

Daphne Kis
Associate Publisher

Release 1.0 21 August 1989