OBJECTS IN PARADISE: OOPSLA PREVIEW

Conference-goers face a tough choice this week allocating their time among Stewart Alsop's Agenda 89, mingling with PC luminaries; the Conference on Computer-Supported Cooperative work (CSCW), research papers and a number of vendors there to learn; and Object-Oriented Programming, Systems, Languages and Applications (OOPSLA), papers but also exhibits and keynotes by two commercial characters -- Ingres and Berkeley's Michael Stonebraker and Lotus and ON's Mitchell Kapor. OOPS is an approach to programming and design, and so it's natural that it already has tools -- mostly languages and lately a few databases. CSCW, by contrast, operates at a higher level, and as yet has garnered some one-off applications but few dedicated tools -- especially any of commercial mien. Object-oriented programming ultimately affects users too, but its reach extends from tools to tools, while CSCW extends from tools to applications -- one level nearer to end-users and real-world models. OOPS, of course, is an excellent approach for the development of CSCW applications.

Object-oriented programming is gaining widespread recognition for a number of reasons, starting with: (1) the emergence of standard object-oriented tools and languages for standard environments, including HyperCard/Talk and most notably (and arguably!) C++, which will have its own conference sponsored by UseNIX in October, with gurus Bill Joy and Bjarne Stroustrup; and (2) the rise of graphical user interfaces, which are painful to build without object-oriented tools and class libraries. Once you let people use objects to make dialogue boxes and menus, they start to wonder why they can't use them for everything else. The recent struggles of certain well-known companies to get their products out on time show the need for some new approach to overcome the growing complexity of multi-tasking environments with graphical user interfaces and multi-vendor machine interfaces.

FORUM PLANS -- SEE PAGE 24
We won't explain object-oriented programming here (see Release 1.0, 87-3 and 87-8 for that), but will instead limn the emerging products and standards that will let it flourish. Like most "new" approaches, object-oriented programming has been around for a long time and experts have done it all along -- in out-of-the-mainstream object-oriented languages and in standard languages using their own constructs for objects. What's new is the proliferation of tools to make it easy for the rest of us.

Most of these tools address only part of the task: They are either languages, databases or environments. That's okay: Many customers prefer to keep using a familiar language or want access to an existing database. A challenge for OOPS vendors (as for AI vendors) is to lure customers into the new paradigm gracefully (although some OOPS fans disdain C++ precisely because it is an incremental solution and not fully object-oriented). Whereas AI is typically a set of techniques for specific problems, object-oriented design is ideally an approach that informs an entire development effort rather than specific modules within it -- although it's easy to use a mixture of C and object-oriented C extensions as desired.

The range of object-oriented tools includes C++ itself, developed at AT&T and now available on a variety of machines from a variety of vendors, other C pre-processors such as Stepstone's Objective C and Complete Computer's Complete C, C++ tools from ParcPlace Systems, non-C object-oriented languages such as Eiffel and environments such as Smalltalk, and object databases from Ontologic, Servio Logic and Grapheal, with others from start-ups such as Object Design and Objectivity on the way. Meanwhile, commercial developers such as Cadware and Virtual Machine offer object-based tools for tasks such as modeling and software and user-interface design. The target markets do tend to be design-oriented, but that will change -- or rather, more and more applications will include a design component so that customers can tailor the applications to their own specific needs or missions.

A quick message on computer-supported cooperative work (pages 16-19)

Although there are few commercial tools for building transaction-based groupware yet (see Release 1.0, 88-6), there is ferment in this area too. Just as objects are ideal for representing and manipulating elements of graphical user interfaces, so are they ideal for representing and implementing the complex data structures and data-driven execution sequences of groupware: individuals, workgroups, toolsets (or applications), e-mail messages and the content of the tasks themselves. Object Lens, an extension of the Information Lens project at MIT, is a case in point. Object-oriented databases come into play for sharing of groupware information, typically much richer than simple numbers and data strings: It could be software modules, designs and related documentation, or components of a richly structured document with complex cross-references.

The province of users and sociologists as much as developers, CSCW raises a wide variety of political, cultural, commercial issues. Culturally, it will require a substantial reorientation of people's thinking about the possible, proper role of computers, as will be discussed in Portland next week. Commercially, it will require extra support in the distribution channel. And practically, it will require system builders to understand and model their organizations better so that they can automate them effectively.
OBJECT-ORIENTED TOOLS

There are two schools of thought in the battle for supremacy between Smalltalk and C++. One says that the advantage of Smalltalk is that it forces you to work in an object-oriented way; the other says that the advantage of C++ is that it doesn't force you to work in an object-oriented way. Overall, the forces behind C++ are gaining ground rapidly, while the other C-oriented approaches are losing momentum: they inherit disadvantages of both approaches, alienating both the C adherents who want a "standard" (if an extended one) and the object-oriented purists, who want the real thing. Nonetheless, all the object-oriented C extensions generate ANSI-draft standard C code for compilation on the user's machine with his compiler of choice. (Yet Objective C's market position may improve substantially because of its adoption by NeXT and a license agreement with IBM.)

ParcPlace Systems, the biggest Smalltalk proponent of all, has bowed to commercial reality and is preparing to offer a suite of C++ development tools in response to perceived demand, but the tools themselves operate in the Smalltalk environment. Meanwhile, a small company in New York City offers Complete C, yet another C variant that is more object-oriented than C++.

INCOMPLETE C++ -- AND COMPLETE C

The major argument against C++ -- and in favor of "truly object-oriented" languages such as Smalltalk, Objective C from Stepstone, COP/TDL from Ontologic, Complete C and Eiffel from Interactive Software Engineering -- is that C++ allows you to create and use objects, but the programs built with it are still C programs with objects rather than object-oriented programs (unless the programmer builds in that structure himself). C++ programs don't all flow from one superclass in a hermetically sealed hierarchy, but rather comprise a mixture of object-oriented and regular C constructs that allow a programmer to break the object-oriented paradigm at will. In C++ (and C) the structure is arbitrary, implicit and difficult to restructure automatically. Of course, most C programmers like it this way; they can control the structure of their programs.

Second, C++ checks typing at compile-time, not runtime, making it faster but less flexible for the prototyping and just-in-time specification of details that object-oriented programmers favor. The system doesn't support factory objects or classes which generate objects of arbitrary kinds, a capability that gives object-oriented languages great power, but must deal with each object as an instantiated type at compile-time. For similar reasons, C++ can't easily handle collections of heterogeneous objects (nor can a do-loop in a procedural language, but iteration in an object-oriented language can).

C++'s classes, in a word, become sterile objects early in the development cycle. They have hidden behavior, but they can't effectively change it once a program is compiled (without recompilation). You can get the benefits of message-sending in C++, but you can't send a message to a generic object/class without knowing the specific subclasses you are (or will be) dealing with. That makes it difficult to prototype in standard object-oriented fashion. You can achieve some of the same power by using virtual functions in C++, but it's still a little awkward and needs more planning.

Release 1.0 22 September 1988
A lot of people are kidding themselves in talking about "when C++ gets fixed." In fact, these presumed deficiencies are conscious design decisions on the part of Bjarne Stroustrup and his team at AT&T. "I don't want the language bent so that it becomes superb for the single job of [for example] running an object-oriented database," says Stroustrup.

Flaws that do merit fixing include the lack of multiple inheritance which will be provided in release 2, due out "soon" (in the first half of 1989, we assume). Product manager Paul Fillinich notes that Brad Cox has called providing multiple inheritance in C an impossible task, and so he's reluctant to promise a date until the job is fully done. Besides, he adds, "It's unsupported, so we want to make sure it's as clean as possible."

Not in release 2 but acknowledged goals for the future include exception-handling (for errors that show up far from the object where they originated) and parameterized types (a quick-and-dirty way to specialize classes without actually creating subclasses -- another programmer-controlled capability that OOOPS fans will scorn). But the basic, hybrid nature of C++ is unlikely to change (it has multiple parents!). In short, C++ provides objects, but it leaves the control -- and the work -- in the hands of the programmer.

The argument of the object-oriented crowd (which we support) is that applications are getting so big and complex that it's impossible for programmers to maintain control effectively, and it's best left to a bullet-proof environment. In the end, programming is like skiing: You'll go faster if you can learn to lose control and let the environment carry you.

Complete C

Attempting to plug the holes he sees in C++ with his own product is Seth Goldstein, president of Complete Computer. Since he graduated from Princeton just three years ago (followed by stints at Sarnoff Labs and Datability), Goldstein has been a do-it-yourselfer. Finding Smalltalk slow and "closed," dissatisfied with C and unimpressed with the popular new remedy C++, he has built his own object-oriented C pre-compiler, Complete C, a PC-based product that delights the beta users we talked to.

Complete C is unlikely to stop the onslaught of C++ where others have failed (notably Stepstone with Objective C and Ontologic with its C Object Pre-processor, below). But he does offer a powerful, easy-to-learn language that generates ANSI-standard C -- limiting the risk for users to the need to rewrite their objects later to fit a new paradigm. "It's not an extension of C++ but an extension of C and Smalltalk," he says. "Also, it's more like an environment than a pre-compiler. There's this database hanging around that knows a lot about your program. It takes care of all the menial tasks relating to building an object-oriented program, so the programmer can spend his time designing."

PARCPLACE SYSTEMS -- BETTER ENVIRONMENTS THROUGH SMALLTALK

Created from the group within Xerox PARC that developed and sold Smalltalk, ParcPlace Systems (see Release 1.0, 87-3) has emerged as a company with a distinct mission (cf. Envos, page 22), one that both reflects and transcends its origins with Smalltalk. In short, says ceo Adele Goldberg, a longtime PARC employee who (under protest) gave Steve Jobs the fateful demo of the
Star back in 1979, "We want to provide environments for information management." In the first few cases, that information is mostly the computer environment itself, as in ParcPlace's Navigator user interface, first available as a UNIX user shell bundled with the Ardent Titan graphics computer.

The Navigator lets users see their files as a tree, and offers a friendly, graphical environment for them to examine and manage their files, do system housekeeping and launch their applications -- at which point they lose the Navigator's "protection," although Navigator is waiting when they return. Users can also associate icons with a variety of scripts, applications and system resources such as printers or other users. Builder/users can use the Navigator toolkit and its array of buttons, scroll bars and other interface objects to enhance their applications. Navigator makes the system visible to the user -- and saves even experienced users a lot of typing and searching through files and directories. The current version runs in X Window and can spawn X Window applications. Navigator as designed for the Titan has its own look and feel, but the toolkit could easily emulate Open Look or Presentation Manager or Macintosh (subject to legal restrictions). Independently of its instantiated appearance, the underlying facility will be ported to other platforms, notably the 386 using DOS or OS/2.

C++ tools from a surprising source

More interesting yet, from a company previously devoted to Smalltalk, is ParcPlace's announcement that it will offer a development environment for C++ in the first quarter of 1989. (It is testing customer demand and soliciting user input by announcing early.) Called Cynergy and developed by Ted Goldstein, late of Yates Ventures, VisiCorp and Computer Innovations, the C++ environment will be a Smalltalk application (i.e., it will run in the Smalltalk environment) with facilities for examining and browsing through C++ code and a proprietary incremental compiler-linker for debugging and testing. While Cynergy won't force C++ applications into the object-oriented mold, it will provide a facility for users to define their program structures explicitly so that the Smalltalk tools can monitor which modules affect one another and control the incremental compilation process.

Most users won't see the Smalltalk environment and won't have access to its self-modifying features; they'll simply have a nice workspace in which to manage and examine their C++ modules. Some would argue that they'll be wasting the potential of Smalltalk by working with a less than object-oriented language in a Smalltalk environment stripped of its power, but if that's what the customers want, so be it. Long run, says Goldberg, it may make sense to build the environment itself in C++, which would offer far greater integration, as well as the ability for users to alter their environment from inside -- an ability that is key to the appeal of Smalltalk.

Freed from its past, ParcPlace is showing a surprising marketing flair, starting with such ideas as the ParcRanger program, which provides unlimited copies of Smalltalk to universities at $2995 annually for one platform, plus $195 for each additional platform type. This includes access to ParcBench, a bulletin board service.

Release 1.0 22 September 1988
OBJECT-ORIENTED DATABASES

Once you've created all these rich objects, how do you use them without loading them into memory at runtime? At some point, even virtual memory becomes insufficient for large applications with lots of data. And how do you share the objects among people and across applications? (If you simply copy an object into another application or someone else's workspace, you lose its links and ability to inherit changes made to its superclass, among other problems.) These are not new questions, but only recently have they seen any commercial answers. Most design-oriented software vendors have painstakingly created their own proprietary file structures that store but generally don't manage their data; Cadware (page 14) is about to breach these limits with its own object base. What we need is standards, but we haven't had enough experience yet to build or commit to one.

The three key commercial players (Release 1.0, 87-8) are Ontologic, Servio Logic and Graphael. All have progressed since last year's OOPSLA, but none has caught the world by storm. Newcomer Object Design promises a standard-platform, high-performance object-oriented database with a C++-based development environment to match. (So does Objectivity, a similar West Coast company that almost was Object Design, but the principals split into two groups with different approaches to visibility.) OD's mission would seem presumptuous were it not for the track records of its principals. Servio Logic offers Smalltalk as a development environment, Graphael has a friendly user interface but no tightly integrated development environment, and Ontologic so far has a language, COP/TDL, but no environment. Much as the world at large has moved beyond sheer database functionality to focus on the quality of related tools and environments, so will the object-oriented databases soon be judged on the quality of their tools.

ONTOLOGIC: BOWING TO STANDARDS

Ontologic was first into the small market for object-oriented databases, and has installations of 60 to 70 copies of its Vbase at 28 sites. Ontologic has its own object-oriented C pre-processor, incorporating COP (for C Object Pre-processor) and TDL (for Type Definition Language), but vp marketing Robert Martin concedes, "C++ is just all over the place and seems to be winning." Ontologic now faces the tricky task of developing a C++ version and moving its users' applications over before they become too numerous and burden the company with a double support requirement.

Under pressure from customers and would-be customers, the company is trying to speed up its own development cycle to have the C++ version ready before late next year as originally planned. It will also have to provide tools to move its users' applications from COP to C++ -- a challenge considering that COP offers features missing in C++ such as iterations (for handling collections), exception handling and robust transaction management. Like everyone else, Ontologic is lobbying AT&T to put in these facilities post-haste. Currently Vbase is multi-user on a single cpu, but not yet on a network of homogeneous, let alone heterogeneous, machines. The system currently runs on Suns and VAXes, with Apollo on the way. Pricing starts at $15,000 on a Sun workstation with runtime licenses negotiable.

With the departure of Bob Berkowitz, Ontologic is looking for a full-time ceo; sales and marketing vp Si Lyle has taken over on an acting basis.

Release 1.0

22 September 1988
SERVIO LOGIC: OBJECT-ORIENTED TO THE CORE

While Ontologic is focused on engineering and CAD/CAM markets, Servio Logic has a higher proportion (although a smaller number) of customers in CASE and text-oriented applications, including configuration management and documentation support. It has 30 installations of its GemStone, 19 of them paid so far. Of the 30, 17 are commercial companies.

The typical configuration uses Sun workstations both for users and servers, although it also runs on the VAX. Servio Logic has gone the full object-oriented route, eschewing such half-measures as C++, and offers Smalltalk as a development environment, although it's written in C and can interface with C applications too. For data access, Servio Logic offers its own proprietary language with a Smalltalk-like syntax, OPAL. (It's an object-oriented equivalent of SQL -- a data-manipulation language for persistent objects.)

Because it uses a multi-user virtual-machine architecture, Servio Logic is likely to have an easier time supporting heterogeneous environments than its competitors, which have to deal with a different C compiler and instruction set in each target environment (until we all have unified UNIX?). That fact underlies Servio Logic's strength in office automation and software engineering environments rather than the dedicated, focused engineering groups its competitors are targeting. GemStone costs from $10,000 to $100,000 on a VAX server, and from $18,000 to $80,000 on a Sun server.

GRAPHAEL: OBJECTS FOR USERS

A company of French parentage, Graphael offers a rich LISP-based system targeted at design applications and gaining increasing acceptance, with 58 copies installed worldwide, all paid for. It has no development environment per se, and is written in LISP, but it comes with a user-interface usable by real people, not programmers. The installations are mostly on Symbolics or Explorer platforms, says president Joe Hancock, but new-sale activity has shifted to newly released versions on Sun and Apollo. Prices range from $9500 on the MicroExplorer to $14,000 at the high end. Graphael is big in aerospace and other large companies, mostly for storage and retrieval of documents and diagrams. In other words, Graphael stores objects and their relationships but doesn't support vigorous manipulation of them.

OBJECT DESIGN: OBJECT-ORIENTED BY DESIGN

Frequently the first company into a market -- VisiCorp in spreadsheets, Apollo in workstations -- isn't the one that flourishes later on. Object Design, the latest, most promising object-oriented database start-up, will have the opportunity to learn from its competitors' experience (read "mistakes"). Indeed, the company is "release 2" for several of its people.

Object Design has no product yet and isn't about to announce one anytime soon, but it bears mention here because its prospects are so bright -- and because it's looking both for a ceo and for prospective customer input. The company was founded this summer by Tom Atwood, until 1986 engineering manager of Ontologic's effort, to build what he calls a "second-generation" object-oriented database based on UNIX and C++ for engineering and design environments. Funding has been committed by Harvard Management and others.
The product, in two components called ObjectStore and ObjectDesigner, is designed to be a substrate for large-scale database-intensive applications in CAD/CAM, CASE, publishing and the like. The company will focus not just on the database itself, but also on the related development environment, including its own incremental C++ pre-processor. Other pieces include graphical diagramming tools for designing and evolving complex class lattices and facilities for managing class libraries.

Atwood points out, "We're competing with systems where 30 percent of the code is devoted to reading data into memory in acyclic graph structures, then flattening it out and shipping it back into storage when the session is over. That limits the size of the problem you can handle, but it accustoms users to the speed of using C structure graphs in memory. For raw performance, you have to own the compiler technology, not just the data structures." He has consciously assembled a team rich in both database and compiler experience: The company boasts an impressive roster of employees, including Dan Weinreb, chief architect of Symbolics' Statice just-released object-oriented database (for Symbolics platforms only, Release 1.0, 88-4), Sam Haradhvala of Symbolics' "foreign languages" group, an expert on incremental compilers and design environments (including a C inspector, Release 1.0, 88-8), and Jack Orenstein, a key figure behind Computer Corporation of America's Probe object-oriented database project (below). The board of advisors includes such heavyweights as Brown University's Stanley Zdonik.

In essence, Object Design is trying to recreate much of what makes the Symbolics environment, including Statice, so valuable without the corollary disadvantages of a nonstandard platform. In addition, the group is paying particular attention to multi-user issues, studying patterns of sharing and cooperation among users. The goal is to be able to define transactions and concurrency differently by user group within the same overall object space. In other words, transaction management, reconciliation of multiple versions and concurrency control will be object- and application-dependent -- a wretched idea for database traditionalists but useful for dealing with rich data and heterogeneous object behaviors.

THE OTHER GUYS

Xerox's Custom Systems Division has just acquired CCA's Probe project, along with the rest of CCA's Advanced Information Technology group. CSD's current plans are to field Probe as a commercial product after working on some engineering and implementation issues (multi-user support, robustness) rather than the research that preoccupied the group before.

Other products on the horizon include Postgres from Relational Technology and likely offerings from Oracle and Sybase, which already handles triggers and other metadata in its SQL Server.

A number of companies such as Informix claim to offer object-oriented databases, but what they have is merely indexed storage and access to structured files containing images or text (a useful capability in itself to be sure).
BETTER THINKING THROUGH GRAPHICS

Many people first encounter object-oriented notions through graphics -- partly because trying to build and manipulate graphical interfaces without objects is such a painful task, and also because graphical objects afford the easiest way to explain the concept. (If we have to hear one more time about how "the circle knows how to draw itself"")

Consider the classic joke: "Ready! Fire! Aim!" It's a joke about sequence -- doing things in the wrong order. It's not a joke a computer could understand. But it's one a computer can easily represent -- even without the bit-map displays, fancy software or any of the stunning image systems we saw last month at SIGGRAPH, the annual festival for graphics zealots. In fact, there is a whole world of graphical thinking outside the ray-tracing, radiosity, scientific visualization and 3-D animation that informs SIGGRAPH.

This other world is based on structure and sequence, concepts and relationships, as opposed to images and visualization of massive amounts of data.

Such structural information comes through clearly on a plain old text pc, starting with the lowly outline. An item's place in an outline -- three indentations to the right, five sections down -- shows its relationship to the other items. Or take something as simple as a pro-con chart: The mere placement of an item on one side or other of the page (screen) says it all.

Think of the meanings you can express with graphics. Arrows can show cause-and-effect, sources of anything (advice, raw materials) or the transfer of property. You can represent simultaneous events far better with a PERT or GANTT chart than with words: It's not just shorter to say; it's simpler to understand. You can see how all the events are related, in time and precedence, and which ones occur simultaneously without being related. Our own family structure, with four parents and numerous step-people, is best rendered with a diagram (or left discreetly unexplained). Want to describe a market? Draw a grid, with different shadings for the different players. (So far the best we can do is generally two axes, which is usually enough for the basic point -- anything more means a complex image and muddy thinking.) How does a company communicate? See the InFlow chart on page 18.

"Bartender schedule"

<table>
<thead>
<tr>
<th></th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Juan</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alice</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rachel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(We may have a problem around 11.30, for the end of the Olympic broadcast.)

Sequence indicates time: That's the principle behind the ready-fire-aim joke. Have you ever seen a time line that began with the future to the left? (Perhaps in the Middle East, where they read right to left.) With arrows added, it also indicates flows of information, cause-and-effect relationships and the like, with PERTT charts, Gantt charts and histograms. (Some of these approaches can be used to specify groupware work flows, as discussed in our June issue.)
Organization charts are another example of graphical structure, showing relationships among data (which could be the components of a task or project, a "work breakdown," as well as tangible things). An org chart, of course, is nothing but an outline (a hierarchical structure) in a different form, as shown for a typical five-person, one-employee start-up below.

The wonderful thing about this stuff is that it's meaningful to the computer too. You can give it simple data and have it generate the displays, or, with a variety of new CASE and diagramming tools, create the images and have it understand (and perhaps validate) the relationships they express. It won't get the jokes, of course, but it will get the tangible meanings.

Moreover, a number of tools can now parse code and generate flow charts, tree diagrams, instruction-frequency histograms and the like. Both Clear Software's Clear (see Release 1.0, 87-10) and Quantum Software's ASMflow are worthy of note as examples of how graphics can be used to display meaning -- and how valuable that capability is when it can automatically translate complex, unintelligible data into a form instantly meaningful to humans. Clear generates flow charts and structure diagrams from Dbase source code. ASMflow does the same for Intel 86-family instruction-set assembly language -- the sort of tool every assembler coder could use. The program itself, says developer and Quantum founder Mike Schmit, "practices what it preaches" in support of tight, pruned, elegant code: It's fast and compact with 15K of help text and 5K of menu setup code in a total of 50K of assembler.

Blue jeans in state transition

factory --> in stock
  ↓
  sold --------------> worn by Juan, 1988-1990
  ↓
  returned*
  ↓
  reticketed*
  ↓
reentered in
stock database*

*This portion of the system could also be represented by a set of groupware transactions modeling store policies and procedures.

Release 1.0
22 September 1988
In short, the real value of graphical thinking is not that you can use a bitmap of a trashcan to represent the act of deletion, but that you can use a sequence of arrows and objects to represent a sequence of actions or a series of relationships that's too complex to express in words. A diagramming tool is far more than just a tool for software engineering: It could represent a factory, the sequence of actions that occurs when a customer calls for a product that's out of stock (to be implemented as software -- or groupware), or the flow of information and funds between marketing partners (also a candidate for groupware). In fact, graphics can be used to model almost anything, and software to implement almost anything.

The question that intrigues us now is: Do people naturally think this way? Will they immediately start thinking graphically once they have the tools to do so? Or do some of us think visually while others think in words? (Some people like directions that are procedural instructions: Turn left at Kendall Square, go three blocks, turn right into the quadrangle; while others prefer maps: It's two blocks north of Ashby, one block west of Shattuck.)

Personally, we rarely draw because our efforts are so clumsy (across), but we'd love to use a tool that made it easy. We rarely use slides, but we always talk with our hands.

VIRTUAL MACHINE'S CHIPS: INTERFACE TOKENS

As we pointed out, graphics are underused because they are difficult to create and time-consuming even for people who are good at it. But if you think of graphics as symbols of objects and relationships rather than bit-by-bit images of a real item or scene, then reusable generic graphical objects are quite sufficient.

A host of systems and application developers would like to follow the guidelines (where available) of the various interface standards, but too frequently it's a lot of work. Vendors need toolkits not to build graphical objects, but to reuse, edit, assemble and link them. Among them are ParcPlace's Navigator (page 4), the Whitewater Group's Actor with a class library for Windows, and MacApp, an Apple tool with objects for the Macintosh. The others one hears about, such as X Window and NeWS for UNIX, are low-level environments concerned with windowing and network management, and don't provide resuable graphical objects. Windows and Presentation Manager likewise are window managers but don't have the ample libraries or the support for inheritance and editing of an object-oriented system. Open Look toolkits from Sun and AT&T should be out early next year.

Virtual Machine Corp. has built one of the first such object-oriented toolkits, called Chips¹ and has sent its specs to the Open Software Foundation, more to gain visibility than in sincere hopes of the OSF's blessing. (However, the OSF is willing to use components, and it could well find a use for Virtual Machine's interface-building technology along with someone else's interface specs.) Although pricing may change should VM be lucky enough to get a nod from the Open Software Foundation, current plans are to charge $5000 for the developer's toolkit, plus a negotiable runtime charge.

---

¹"Chips" as in components in the semiconductor world or Stepstone's Software ICs.

Release 1.0

22 September 1988
Chips runs on LISP machines currently; a new version written in C will be out for OS/2 in October. It will create calls to the GSS graphics sub-system, with support for PM when PM is ready. Chips will support Open Look or whatever emerges as the standard(s) for X Window. (Once you have the object-oriented development system, it's relatively easy to tailor an interface to suit. The tough part is building the tool and the object classes for each target hardware environment or window/network manager.) Chips' objects retain their power even after they're created, and can link to C routines that can change them dynamically at runtime (turning a face from happy to sad according to the results of a query, for example).

The company was founded last year by Jeffrey Bonar, formerly a professor at the University of Pittsburgh (no, not Carnegie-Mellon) who had done consulting work for Carnegie Group, Sun and IntelliCorp (visible in IntelliCorp's KEE on the 386 under Windows).

V. I. CORPORATION'S DATAVIEWS: DYNAMIC DISPLAYS

Scientific visualization, a hot topic right now, is about using graphics to display data -- red for hot areas, blue for cool ones; 3-D blow-ups of molecular structures that can't ever be "seen" because they're moving too fast and are smaller than the wavelengths of visible light; and computer models computed from limited data or simulations of air turbulence, geographical terrain and the like. To make it look good, there's the emerging standard from Pixar, RenderMan (more on that in the future).

But even here object-oriented graphics plays a role. While Virtual Machine focuses on graphical objects and structures in relation to code and application elements, V.I. Corporation has developed an object-oriented interface tool, DataViews, targeted at commercial software developers and specialized for the display of data changing in realtime. Want to show pressure and 20 other parameters in your nuclear powerplant, or the volume of transactions in each of 20 branches, or the position of a hundred airplanes on a grid (with color to show altitude)?

V.I. Corporation's DataViews makes it relatively easy. It doesn't do anything a C programmer with lots of time on his hands couldn't do himself -- but he'd find the task pretty boring, and V.I. has already done most of the job effectively and elegantly. The tool is constructed to hook easily into a variety of environments, including Sybase, under a joint marketing agreement; Neuron Data's Nexpert Object, with a bridge sold by Neuron Data and Bechtel as a front-end for a variety of expert systems, especially monitoring applications; and Quotron, bundled as a runtime interface. Not only can the resulting DataViews interface represent the data; it can also easily follow rules (written in C, not within DataViews itself) such as "Turn red when the temperature exceeds 90°," "Ring a bell when when value x is greater than 1.3 time value y," and so forth. The basic objects, an array of 43 switches, dials, meters, gauges, bars and pie graphs, can easily be resized, recalibrated, rotated and linked to one or more sources of data -- real or simulated, or even a user's manipulation of some of the objects themselves with a mouse.

Of great value in this persistently heterogeneous world, DataViews also lets a developer write an interface once to run on 13 different hardware platforms from nine vendors. That advantage, piddling during initial develop-
ment in the face of V.I.'s $17,700-plus license charge for the developer's package, gains value quickly when the developer discovers that each subsequent enhancement requires recoding across that same number of platforms. "Often people don't use us at first," says V.I.'s Deborah Huisken, "but they come back to us later." The runtime module, which lets a user customize an interface but not create links to data sources, costs $3000 and down.

The company was founded in 1986 by George Brandt and a team from a predecessor company, Visual Intelligence, which sold similar products direct to engineering and scientific users. But the venture capitalists pulled the plug, and the company reorganized (with only its employees for funding) to sell through OEMs and VARs. It is now profitable with revenues heading for $5 million this calendar year.

**JUAN AND ALICE AT SIGGRAPH...** The visions were spectacular, the renderings were realistic, and the film show was terrific. It played during a stormy night to a self-congratulatory, enthusiastic audience, at least half of whom must have worked on one or more of the films shown. As we walked out into the storm afterwards, one wag pointed to a flash of lightning and echoed the refrain we'd heard all evening: "I wrote that software!"

** ***

**A NOTE ON NeXT**

It has become conventional wisdom since Steve Jobs (and others) said it long ago: The personal computer should be an appliance. Yes, indeed, the pc should be a small, friendly thing that you can have a personal relationship with. It does your bidding, stores your data, and you can use it to help you do your work. It has a user-friendly interface with icons and bit-map graphics, so it can show you pictures as well as text or even diagrams.

But this notion is in for an upheaval. The pc of the future is not going to spend most of its time talking with you; it's going to spend most of its time talking for you. Given the power it's going to pack inside that cute little form factor, it would be crazy to waste it all on communicating with a single slow-thinking, slow-typing human being. Instead, your pc is going to be hooked up to a network of other pcs and bigger machines, and it will spend most of its time interacting with them on your behalf. The shift is a sociological one: Your computer will actively look after your interests in a world populated with computers looking after other people's interests.

We don't know much about the NeXT machine. We haven't seen it. But that leaves us free to speculate. Based on what little we've heard beyond the usual (display PostScript, UNIX, the IBM and Stepstone deals), we suspect that the NeXT system will have the power and the software to be a personal agent (cf. Object Lens, page 17). It may not say so in the marketing literature, but it will say so in the positioning, which always been a strength of Steve's.

*Release 1.0*  
*22 September 1988*
At some point every component supplier wonders why his customers can be so successful reselling his products at a higher "value-added" price, and decides to add that value himself. Such is the case with Cadware, which for years has been a supplier of CASE-building tools to some of the better names in the upper CASE business including Applied Data Research and Yourdon. Using Cadware's factory, its customers can design diagramming tools to fit virtually any methodology, with whatever interface the reseller wants to incorporate. The users receive a runtime version configured by Cadware's OEM, with a library of objects embodying the precise rules and graphical objects that the OEM has specified. They can use and reuse the design objects, but can't create new ones or substantially alter their behavior. Cadware is also an example of a vendor reaching the limits of object-oriented programming without an object-oriented database...

Now, as the marketplace is growing more sophisticated, Cadware founder Bob Dunn sees an end-user market for his tool-making tool, Sylva, letting customers build their own design tools to implement the specific rules and objects they want in their own organizations. Moreover, he has the funds to implement the vision and support his own marketing effort, from a recent infusion of equity capital from Ameritech Development Corp. Although Ameritech has just sold its Applied Data Research dbms unit to Computer Associates, Dunn sees no change in the parent's interest in his own outfit, which targets markets beyond software engineering, starting with communications. The others are the modeling of manufacturing and workgroup processes.

Cadware's Sylva system now comes as the Developer, a full-fledged runtime CASE tool which the customers can use as is or, with the purchase of Sylva Foundry, customize to their own specifications, incorporating their own design rules and validity checks as object behavior. The heart of Sylva Foundry (from the Latin for forest, not trees) is its "icon drums," a set of graphical objects representing software elements and related rules (how they can be used; what other elements they must, may or may not be connected to; associated help and error messages, which can help promulgate corporate policies; etc.). Builder/users can build their own icon drums, with the specific rules, drawing styles and other options they prefer, using Cadware's patented Rule Tool, the heart of its offering.

Sylva is a DOS-based system, with objects but without a database to manage them. The system is focused around a set of discrete diagram types (objects) and the models each represents. Although the models can be linked when they get too large and complex or use different diagramming techniques, they can't be verified globally; that is, consistency across modules can't be checked because each module has its own data structures. Data can be transferred from one diagram paradigm to another only through a cumbersome
translation process. By mid-1989, Cadware plans to field a far more powerful UNIX-based system built around an object base with a new version of RuleTool, LogicTool, that will be able to handle more complex, conditional rules. The various diagram models will become applications addressing the object base, rather than applications that hold their own data.

POSE: PRICE IS NO OBJECT

While Cadware is going after the high end, POSE, a new CASE-market entrant from Singapore, is making it clear that traditional CASE diagramming tools have become a commodity. With Borland-style pricing and a set of full-featured modules for a PC with DOS, Computer Systems Advisors is about to bring CASE to the masses. For a couple hundred dollars, why not try it?

Computer Systems Advisors is the top software company in Singapore, which is mostly home to manufacturing facilities of Japanese and U.S. hardware companies. As such, it's a favorite of the Singapore government, which invested $2-3 million to develop POSE, for Picture-Oriented Software Engineering, and gave the rights to CSA. The money seems to have been well spent: POSE exploits the PC to its fullest, and comes in modules to reduce buyer resistance to a minimum. Anyone can afford to buy just a couple of modules at $295 each, or a starter kit of four for $885. With ads and a 315,000-piece mailing this summer, the company has already garnered 74 U.S. customers.

The system itself is relatively inflexible (unlike Sylva, above), but a good way to get into diagramming. In a couple of years, CSA, which is not unsophisticated itself but now serves users who may be, plans to bring out UNIX- and OS/2-based tools with greater features and functionality.

MAXEM'S CAUSE: CASE WITHOUT TEARS

Programming has always been an abstract, conceptual exercise -- even when practiced by programmers who like to feel the bits between their toes. But the profusion of graphical tools is changing that. The most extreme example we have seen is Maxem's Cause, which you might call a direct-manipulation programming tool (see Release 1.0, 88-3), much as PageMaker is a direct-manipulation page-layout system. In fact, Cause is a database-based application generator in sheep's clothing, and provides a fairly rich development environment for novices who like to see the screens beyond their fingers.

Designed by Dave Thomas, who founded Cyma (accounting packages) and sold it to McGraw-Hill for $15 million, Cause lets users define all the things a database application needs: menus and dialogue boxes, forms for data entry, data records, computations, reports, etc. The system leads the user through the process step by step, and virtually guarantees an error-free application because there's almost no way the user can fall off the path.

However, the user can likewise not exceed the path laid out: An application that doesn't follow Cause's fairly simple model simply can't be created. There's no related programming language. We suspect that Maxem may reach quite a large market, however. The product runs almost identically on PCs with DOS and on the Macintosh, and is cleverly priced like a language to encourage a proliferation of third-party applications, called Effects®. The system costs $495 for end-users and $595 for authors, who may sell without royalties as many runtime applications as they can.

Release 1.0

22 September 1988
GROUPWARE ON THE SHELF

While pc software generally has become easier to use and install, and its users have become more adept at installing and using it, groupware is going to require renewed attention to marketing and support issues. Groupware products are bought not for individuals but for groups, and bought by... well, we're not quite sure yet. Nor do we know how to reach those buyers, or how to help them install the stuff.

Distribution issues are scaring otherwise eager venture capitalists away from investing in groupware. However wonderful the tool, it won't amount to much if it can't reach its users. The recent stunning growth of channels such as Egghead Software is of little relevance to this emerging category.

Groupware will be sold mostly as tools, not applications. Like 1-2-3, groupware helps people build models -- but not financial ones: Groupware models assist and execute workgroup procedures. Groupware is not a personal tool but rather a corporate tool. Yet the last place to try to sell groupware is be to the traditional MIS department. Ideally, groupware provides the system underware to help workgroups automate themselves (either democratically or at the leader's direction) without the aid of the MIS department. Perhaps the closest parallel is databases (or, not an encouraging notion for investors given the fragmentation of this market, accounting software). By this, we mean server databases storing data for groups, not the personal databases where individuals keep their rolodexes. Most multi-user databases are sold through VARs or through high-end dealers who also handle the LANs that make the databases accessible to multiple users.

Thus the prospective groupware vendor has only to find the resellers used by 3Com, Novell and Banyan, right? Or the dBASE VAR association? Or Businessland and systems integrators such as Arthur Andersen? Yes, but do those resellers want to take on the burden of learning and supporting a new groupware tool, let alone a bunch of them? Networks are bad enough, and they require half as much knowledge of customers' business as groupware will.

On the other hand, groupware tools are designed to make the development of such systems simpler than, say, a database used for the same purpose. We expect many groupware tools will in the end be part-application/part-tools built around databases such as Sybase, in which case they'll flow naturally through channels that those vendors build, along with other database tools.

Of course the prime determinant of resellers' willingness to carry any product is customers' clamor for it. How can we persuade people of groupware tools' utility? The simple answer will be a proliferation of success stories, which takes time (as for single-person tools such as Lotus Agenda). As for most tools except 1-2-3, the success stories will be about applications created with the tools rather than about the tools themselves. In fact, there are already a number of success stories about groupware, but they aren't presented as such: They tend to be task-specific systems built for groups that perform their missions on computers -- software developers, writers, designers. (They include systems for large publishing enterprises such as System Integrators' and Kodak's Atex, factory automation systems and custom-built software development environments.) Moreover, these early successes are mostly applications built from scratch or with a database, rather than with groupware tools -- and as such, they lack the easy enhanceability and customizability that will characterize tool-based groupware.

Release 1.0

22 September 1988
OBJECT LENS: INFORMATION LENS RELEASE 2.0

It's not every academic project that is so successful as to merit a second release, but MIT/Sloan School's Information Lens has just qualified. At this month's Conference on Computer-Supported Cooperative Work, MIT's Tom Malone will present "Object Lens: A Spreadsheet for Cooperative Work," a paper by himself, Kum-Yew Lai and Keh-Ching Yu on Object Lens, the successor to Information Lens (Release 1.0, 86-10).

Information Lens acts as a user's agent in creating, sharing, storing and filtering information. Its primary idea is the semi-structured message (an idea picked up in Oracle's SQL*Mail, Release 1.0, 88-5), a cross between a database record (with a linked form) and a regular e-mail message. As objects, such messages can be further specialized: a meeting message, for example, can be specialized to concern the regular Monday-morning meeting, with the list of attendees set to a modifiable default. There may be several kinds of subscriber-call messages, each including the subscriber's name with some data filled in automatically from the database and other information different each time -- lost-issue messages, missing-issue, multiple-subscription request, binder order, etc.

Each user of the system can build filtering rules governing the disposition of these messages, so that meetings could be automatically scheduled and responses generated, important messages could be selected and the like. But these messages are only semi-structured, so that they can also contain text or images unintelligible to the computer but meaningful to humans.

Objects for everything

That was Information Lens. It knew about messages, and held additional knowledge represented as filtering and action rules. Object Lens extends the idea further, with a seamless amalgamation of e-mail, hypertext, intelligent agents and objects. Now you can create semi-structured object classes (and knowledge) for anything: types of people (job titles and relevant skills or responsibilities), resources and other concepts needed to model a particular workgroup's operations. The actual subject of a business -- customers, products, projects, documents -- can also be incorporated into the model, although any heavy-duty data (such as accounting or transaction data) would probably be better represented in a traditional database with links to Object Lens.

Object Lens supports end-users in building group information-sharing applications, with the system automatically handling the details of creating data structures, handling messages and executing rules. Object Lens can also create and edit objects, do date and time arithmetic, create and manipulate links, view collections of objects in different ways (forms and reports, in dataspeak), and specify and execute rules. A rule might be, "If a message concerns a problem more than two weeks old, route it to the boss of the person to whom the problem was assigned." That would be pretty tough logic for any other system to handle gracefully.

Malone hopes to use Object Lens to implement quickly prototype versions of other groupware systems such as NoteCards, Higgins, The Coordinator and ForComment -- just as an exercise. Object Lens, like Information Lens, is currently implemented on LISP on a Xerox D-machine (the project is partially

Release 1.0	 22 September 1988
SAMPLE COMMUNICATION MAP
Showing Interactions In and Amongst Departments

ENGINEERING & RESEARCH

MANUFACTURING

person linked only to other group... should this job be transferred there?

No direct link between Sales and Manufacturing...
Manufacturing can not react quickly to higher/lower than expected sales

SALES

why is this person isolated from everyone?
group has poor internal connections

why is this group isolated from the rest?
group has excellent internal and external connections

ADMINISTRATION

why is this group isolated from the rest?

V P MFG, ENG, & RES

too many connections... VP obviously came from Manufacturing
to other group...

V P SALES, MKTNG.

no connection between liaisons; is there a power struggle?

very well connected person
funded by Xerox PARC). In due course it should be moved onto a UNIX platform (probably the SPARC architecture that Xerox has pledged to adopt), but still with LISP. Some day perhaps a commercial version written in a standard language (C++) will make its way to market. The closest thing we're aware of, but more ambitious, is the ON Technology development environment, the specs of which are still shrouded in mystery.

A fundamental principle behind both Lenses is "semi-structure," the delicate balance between structure and nonstructure -- or information whose structure is not yet apparent. Creating semi-structured messages was a first step, providing a powerful way for people to manipulate the explicit structures in their messages and a pragmatic way to handle the unstructured parts -- all the while encouraging them to move more and more information into defined fields or slots in a semi-structured message template. Now Object Lens is consolidating and advancing further into the unspecified, unstructured territory of the everyday world, helping people to discern and model its structures so that more routine work can be handled automatically.

InFlow: FIRST THE MODEL...

A reader writes: "I agree [that] 'the better we understand group work and can make its activities explicit, the better we can model it on computers.' [Release 1.0, 88-6] ...we must do organization analysis along with systems analysis if we are to gain the expected payoffs from information technologies. Organization analysis includes looking at information flows & dependencies, policy & procedures, job design, coordination requirements, strategy & structure, employee skill base, etc. A technology-driven approach is not sufficient."

The writer, Valdis Krebs, has led the implementation of a number of human resources computer systems at Toyota USA and TRW, and is now helping to start up Toyota's luxury car division, Lexus, in Torrance, CA. Krebs has developed an in-house Macintosh-based organization modeler, InFlow, that does some of the ground work he mentions above. (He wrote the tool as a two-for-one project for separate UCLA classes he was attending on Prolog and Organizational Dynamics.) It starts as a questionnaire that quizzes each group member (typically comprising multiple groups) about his relationships with other people in the organization: Who gives you information? How much, how accurate, how timely, how frequent, in what form? Whom do you give information to? Sometimes answers don't always match!

From this simple data set (limited only by disk size, typically tens or hundreds of people), InFlow draws not an org chart but an information-flow chart (a hypertext display with typed links), where the width and intensity of the lines from node to node indicate the qualities of a relationship. The system also has a notion of hierarchy, and can indicate relationships between individuals, or between workgroups. If there's slow communication between two groups, for example, the system can descend a level to show exactly who maintains the relationship between the groups: Perhaps it's too few people for the amount of information involved, or perhaps one person simply slows things up. Once you know these things, says Krebs, you're in a much better position to install group applications. You know where improvements could have the most impact, and you know what problems you're trying to solve. (The illustration across is not an actual screen shot, but indicates the kind of information InFlow can reveal.)
OPEN BUS FOUNDATION

Like the Open Software Foundation (Release 1.0, 88-6), the group of companies who recently banded together to propose an alternative to IBM's Micro Channel Architecture are doing at least as much to bless the product they're fighting as they are to compete with it. Their very existence raises again the question: What are the benefits of the Micro Channel?

If the old AT bus isn’t a sufficient answer to that challenge, perhaps there are some advantages to the Micro Channel -- ones that will become compelling before the Open Bus people come out with a product a year or more from now. The most significant benefit is upgradability -- the same invisible benefit offered, interestingly, by Sun/AT&T's SPARC architecture.

However, the situation of the Open Bus Foundation is substantially different from the OSF. None of the sponsoring members of the bus consortium has its own successful proprietary offering that would dilute its commitment to the Extended Industry Standard Architecture (EISA) as the OSF founders do. The EISA offering enters a vacuum except for IBM's Micro Channel. The OSF will compete in a crowded market: "Open software" is a position in a market, not a market itself.

Timing

The timing of the announcement is somewhat strange. Evidently, the group formed rapidly and decided it should announce the effort since it was inevitable that the news would leak out. In one sense, this should have happened (and almost did) last year, right after the announcement of the PS/2. The specter of the open bus then might well have stopped the PS/2 in its tracks. The announcement now is an implicit endorsement of the PS/2's message: "The old AT bus ain't enough." If you want more, and you need to buy something now, the PS/2 is your only choice. If you can wait two years, then you'll have several choices. But you should buy an AT-type machine now only if you have no intention of upgrading for several years.

The likely immediate impact of the fracas is likely to be small: Some people will see it as an implicit endorsement of Micro Channel, and go ahead and buy that. Others will figure that the AT will live on, in one form or another, and they can buy an AT machine with comfort. (They may have to discard the machine-cum-bus itself, but their peripherals will live on. Peripherals frequently account for half or more of system cost.) And others will wait. The net near-term effect will probably be indistinguishable within the brownian motion of the total market.

Nonetheless, we expect the EISA to be a long-term success: It makes a lot of sense in an area where customers are too confused to ask too many questions. We certainly wouldn't venture to do a technical comparison of the two products, one not publicly specified and the other not yet built -- nor would we believe anyone else's claims. We expect that both buses will work, and that IBM will have Micro Channel II by the time the Open Bus Foundation has EISA I. That's what competition is all about. Note that although Intel will make the initial implementations of the EISA, anyone can do so for a nominal payment made through a law firm. Clone chip-maker Gordie Campbell of Chips & Technologies has already signed up.

Release 1.0 22 September 1988
Indeed, the market's recent fascination with "openness" (see Release 1.0, 88-6) raises interesting questions about the roles of Intel and Microsoft in the PC marketplace. Some people consider openness to mean "free;" others think it means "interface specs are available." The moderate view holds that "open" items should be available from a number of sources. Neither DOS-OS/2 nor the 386 chip (not required to run the EISA, but likely to support most implementations) properly qualifies under that definition.

Software for OS/2

Missed in all the commotion last week was a move of far greater significance -- the addition of support for RPG II applications on an OS/2 platform. In one fell swoop, this adds thousands of small-business applications to the almost empty OS/2 roster. It also repositions (or extends the positioning of) OS/2 into the server/multi-user arena from the prevalent notion of OS/2 as a high-end single-user workstation. This gives it greater firepower to fight off UNIX in the small-business market, answers criticisms of OS/2 as useless, and radically alters the positioning of IBM in the small-business market (see Release 1.0, 88-8, "Free advice") by extending the AS/400 line downwards into the low teens as well as extending PS/2-OS/2 upwards by providing a smooth migration path to the AS/400. Now IBM can offer small businesses a single-vendor solution.

To be sure, these benefits redound in part to Compaq and the clones too, who should be able to run the same RPG II packages. But guess who controls the distribution of most of those packages -- IBM VARs. And guess who gets the upgrade business -- IBM VARs.

* * *

NITTY-GRITTY EXPERTS: IMPLEMENTATION EXPERTISE

We recently attended two AI conferences at opposite ends of the research-to-commerce spectrum: AAAI in St. Paul, and "Increasing management productivity with intelligent systems" in Aspen, sponsored by the University of Colorado's Center for Applied Artificial Intelligence. AI has moved out of academia, but it still lacks an infrastructure of craftspeople, not theorists.

We enjoyed AAAI far more than we expected to. The conference seemed far more at ease with its role in the world than it did last year: It no longer made a pretense of serving business needs, and the sessions were firmly research-oriented. (We spent most of our time at a couple of workshops, which had the highest concentrations of clever people around.) On the show floor the glitz and hype and pseudo-friendliness of previous years were gone, replaced by serious tools for serious people running on standard machines. (Also gone was the traditional extravaganza hosted by Symbolics, replaced, symbolically, by Sun's bash at the Science Museum. Meanwhile, Gold Hill kept up tradition with a beery bash at the old train depot.)

The tire kickers of previous years who incessantly asked "How many rules can it handle?" were replaced by more knowledgeable, open-to-buy customers who asked questions about the suitability of the tools for the particular problems they had in mind. We suspect a lot more business got done this year than last. Prices were much lower, offering encouragement to both serious...
buyers and experimenters. Part of the drop came from actual software and hardware price cuts, but much was due to a proliferation of low-end platforms, especially the Macintosh combined with chips from Texas Instruments and Symbolics, and even the Macintosh alone.

The Colorado conference, by contrast, was directed mostly at managers not yet convinced of the utility of expert systems. The sessions were effective at that purpose, with speakers detailing successes in underwriting (Fireman’s Fund), truck-loading (IBM), marketing (Balsams Grand Resort Hotel), personnel scheduling (Management Robotics Inc.), resource allocation (5thGen Technology) and the like.

But we’re still hard put to find the kind of conference we suspect is really needed -- one where practitioners can trade tips and hard-won lessons about the nitty-gritty of building expert systems: How should you organize rules? When is backward-chaining effective; when is forward-chaining better? What are the political issues? How diligent are various tool vendors at customer support? Can you improve performance by using an object-oriented system in place of data-defining rules? The closest thing we know is Decision Support Technology’s Expert Systems Implementation Group (see Release 1.0, 87-10), quite a commitment in time (ten days over the course of a year) and money ($30,000 for a team of two people). In a similar but less-intensive vein, there’s the TISC Seminar on Emerging Software Technologies this November (see calendar, page 29).

Technology transfer is happening, but what about expertise transfer? The lack of facilities to support such dissemination of experience is why we think Teknowledge may flourish as a consulting firm now that it has abandoned the tools business. And it is why tools vendors such as Coherent Thought (mostly a bunch of Teknowledge veterans; Release 1.0, 88-1) and Gen-sym, both with tools already tailored for specific kinds of problems with much application-specific control logic and data structures already built in, should likewise flourish.

ENVOS: XEROX AIS IN A START-UP’S CLOTHING

The biggest announcement at AAAI was the emergence of Envos, the Xerox unit 85-percent owned by employees that will take over Xerox’s AI systems business in an attempt to shield this small unit from its giant parent. Its dowry is a nice set of products, including the NoteCards hypertext technology (see Release 1.0, 84-16); LOOPS, the LISP Object-Oriented Programming System; and Rooms. Rooms is a clever idea: If we have all these windows that mess up our screen, why not organize them into rooms, or collections of windows, that represent the groupings of facilities that we use for particular tasks? This idea is about as protectible as the idea of windows itself -- but Envos has a nice implementation that enhances its offerings although no one’s likely to buy Rooms without the underlying software.

The real challenge for Envos will be to build mission and character -- precisely the strengths of the other notable PARC spinoff, ParcPlace (see page 4). Running a profitable business -- Envos’s immediate goal -- isn’t a mission; it’s the possible by-product of a mission. But Envos does not yet seem to have a strong enough personality or mission to guide it past the sirens of illusory market opportunities.

Release 1.0

22 September 1988
ORACLE IN SEARCH OF GREENER GRASS

While Management Science America looks sickly, selling into a mostly mature market for mainframe-based accounting systems, Oracle sees an enticing opportunity for accounting applications in companies downsizing onto departmental computers -- frequently with Oracle databases. Its new product line, Oracle Financials, was overseen by Jeff Walker, chief financial officer of Oracle since 1986. Now it's clear what Walker has really been doing.

Walker is the founder and former CEO of Walker Interactive, which is already implementing the same strategy across a variety of databases (catching customers as they move from IMS to something more modern, such as DB2) after a couple of disastrous years due to overambition and underperformance; see Release 1.0, 88-7. Like Walker and Cullinet, Oracle is using its own tools to build the system, and then providing (versions of) those same tools to its users so that they can customize their systems and build ad hoc queries and reports.

Aside from the finer control and reduced porting problems that Oracle gets from owning the underlying database (although it does have to port the database!), Oracle also has the advantage over Walker of a 500-person salesforce and an installed base of 12,000 Oracle sites to sell into. Walker, of course, addresses a wider marketplace comprising users of IBM's DB2 and other databases as well. Although Oracle can address these customers as well, both from a marketing and from a technical (with SQL) perspective, for practical purposes it has a smaller but probably more welcoming target market.

Indeed, the essence of the Oracle approach (as always) is brilliant marketing. The accounting product will be sold by teams of traditional database salespeople joined by accounting specialists so as to trade on existing account control extended by specific knowledge that the generic, Oracle salespeople lack. Both team members get commissions, so that the database salespeople aren't tempted to stick to what they know.

Back to basics

But what of the basic premise? Are companies really downsizing? Yes indeed. Probably not enough to solve MSA's problems even if it were to adopt Walker's or Oracle's strategy, but enough to add a nice source of incremental revenue for Oracle as it faces increasing competition in its basic markets. As companies start to automate and link not just accounting but also mission-critical applications (such as using customer data for promotions and customer service, paying royalties to third parties based on sales by particular products, and tying commissions to profitability rather than sales), they see the need for more local control of and access to data, by way of databases that are both less centralized and more flexible and connectible to each other than those underlying traditional accounting systems. Generally, only relational databases such as Oracle's and IBM's DB2 running on a departmental machine offer these benefits. In other words, all the benefits Oracle promises from its database management systems now apply to a line of accounting packages as well.
1989 PERSONAL COMPUTING FORUM: GET SET FOR THE NINETIES

The Personal Computing Forum will take place a month later than usual in 1989 (no nefarious reasons, just the availability of suitable hotel space), from March 19 to 22. We are moving back to our old haunts, Palm Springs, in search of good weather and morning alertness among West Coast attendees. Invitations won't be mailed out until after Comdex, so don't worry yet. (If you get your registration back to us by year-end, you should have no problem getting in.) The Forum will be managed by Daphne Kis, who will be joining us (thank goodness!) in mid-October from Home Office Computing. We have just started inviting speakers; we will publish a preliminary list next month.

Note the dates: March 19 to 22

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, groupware, text management and connectivity markets, and artificial intelligence. Editor & publisher: Esther Dyson; associate publisher: Daphne Kis; circulation & fulfillment manager: Lori Mariani; executive secretary: Maria Soler; copy chief & transcripts editor: William M. Kutik. Copyright 1988, EDventure Holdings Inc. All rights reserved. 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 22 September 1988
RESOURCES & PHONE NUMBERS

Gordon Bell, Ardent, (408) 732-0400
Bjarne Stroustrup, Paul Fillinich, AT&T, (201) 522-6664
Bob Dunn, Cadware, (203) 397-2908 or (800) CAD-WARE
Diane Smith, Umesh Dayal, CCA (Probe), (617) 492-8860
Gordie Campbell, Chips & Technologies, (408) 434-0600
Vadim Yasinovsky, Clear Software, (617) 232-9788
Seth Goldstein, Complete Computer, (212) 582-2635
Sherman Drusin, Computer Systems Advisors, (201) 391-6500
Larry Meador, Decision Support Technology, (617) 354-6400
Walt Samuelson, Roger McKee, Envos, (415) 966-6200
Bill Hoffman, Joe Hancock, Graphael, (617) 890-7055
Roger Sipl, Informix, (415) 322-4100
Valdis Krebs, Lexus (Toyota), (213) 328-2075
Davy Thomas, Maxem Corporation, (602) 827-8181
Tom Malone, MIT/Sloan School, (617) 253-6843
Tom Atwood, Dan Weinreb, Object Design, (617) 270-9797
Drew Wade, Bob Field, Objectivity, (415) 854-8900
Jeff Walker, Pete Tierney, Oracle Corporation, (415) 598-8000
Bob Martin, Ontologic, (508) 667-2383
Adele Goldberg, Ted Goldstein, ParcPlace Systems, (415) 859-1000
Mike Schmit, Quantum Software, (408) 244-6826
Mike Connell, Jacob Stein, Servio Logic, (503) 644-4242
Dennis Sisco, Stepstone, (203) 426-1875
Deborah Huisken, George Brandt, V.I. Corporation, (413) 253-3482
Jeffery Bonar, Bill Weil, Virtual Machine, (412) 361-5028
Chuck Duff, Mark Achler, The Whitewater Group, (312) 491-2370
Phil White, Wyse Technology, (408) 433-1000

COMING SOON...

- The NeXT announcement.
- Graphics standards.
- Document and file management.
- Constraint-based modeling.
- And much more...

Release 1.0 22 September 1988
RELEASE 1.0 CALENDAR

October 3-4

October 3-5
Adam Green's dBASE² symposium - Burlington, MA. Three-day conference of dBASE third parties, including clones, compilers and complements. Contact: Marny Peabody at Digital Consulting, (508) 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 5-6

October 7

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 9-12

October 10-11
3Com network systems forum - San Francisco. With Bill Gates, Rod Canion, and a panel on groupware moderated by Esther Dyson and featuring Tony Kobine, FCMC; Camilo Wilson, Lifetree; Eric Sall, Lotus; Andrei Jezirski, Coordination Technology. Contact: Cheryl Smith, (408) 562-6400 or (800) 638-3266.

October 10-12

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.

Release 1.0 22 September 1988
<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
<th>Location</th>
<th>Contact Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>October 11-14</td>
<td>Info Show - New York City. Contact: Frank Fazio, Cahners Exposition Group, (203) 964-0000.</td>
<td>New York City</td>
<td></td>
</tr>
<tr>
<td>October 12</td>
<td>The NeXT announcement - San Francisco. By invitation only. (Test your social standing...)</td>
<td>San Francisco</td>
<td></td>
</tr>
<tr>
<td>October 12-13</td>
<td>Bioanalysis and computing conference and exposition - Burlingame, CA (SFO airport). &quot;Computing: Its role in bioanalysis.&quot; Data analysis, imaging, instrumentation control, etc. Contact: Norm de Nardi, (415) 941-8440.</td>
<td>Burlingame, CA</td>
<td></td>
</tr>
<tr>
<td>October 16-19</td>
<td>ADAPSO MANAGEMENT CONFERENCE - Dallas. The software and services industry's premier gathering. Contact: Sheila Wakefield, (703) 522-5055. Followed directly by...</td>
<td>Dallas</td>
<td></td>
</tr>
<tr>
<td>October 19-20</td>
<td>Vardex - Dallas. The value-added reseller &amp; dealer exposition. Co-sponsored by Adapso and NAVAR, the National Association of VARS. Vertical markets and tailored software galore. Contact: Dan Dembicki, (313) 274-8590.</td>
<td>Dallas</td>
<td></td>
</tr>
<tr>
<td>October 17-20</td>
<td>C++ conference - Denver. The first full C++ conference, sponsored by UseNIX, with topics and speakers inherited from last year's C++ meetings and workshops, as well as gurus Bjarne Stroustrup and Bill Joy. Chaired by Andrew C++ Koenig of AT&amp;T Bell Labs. Contact: Judy Desharnais, (213) 592-1381 or 592-3242.</td>
<td>Denver</td>
<td></td>
</tr>
<tr>
<td>October 17-20</td>
<td>Electronic publishing conferences - San Jose. Sponsored by CAP International. Keynote by Arthur Laffer. In three parts: electronic printers, color hard copy and electronic publishing. Contact: Jean O'Toole or Kristin Fischer, (617) 982-9500.</td>
<td>San Jose</td>
<td></td>
</tr>
<tr>
<td>October 18-20</td>
<td>OPEN SOFTWARE CONFERENCE '88 - London. With speakers from Sun, IBM, AT&amp;T, etc. Sponsored by Sphinx Software. Contact: Amanda Stuart, (44 62) 882-2266.</td>
<td>London</td>
<td></td>
</tr>
</tbody>
</table>

Release 1.0 22 September 1988
October 23-28 Monterey Classic - Monterey, CA. Where investors and companies discuss common goals... Contact: John Baumeister, (408) 987-4200.


October 24-25 Intertainment '88 - New York City. A conference on interactive entertainment for and by vendors in software, broadcasting, movies, videotex, CD-ROM, theatre, advertising, etc. Keynote: Nolan Bushnell. Also: Gary Arlen, Trip Hawkins, Robert Gehorsan (Prodigy), "Tamara," Ed Schlossberg. If you don’t know these names, forget it; if you know them all, you must be a dilettante. Sponsored by Alexander & Associates. Call Sally Chin, (212) 382-3929.


October 26-28 Venture forum '88 - Boston. Sponsored by Venture Economics, the authority in the field. Talks by many VCs and by perennial supplicant Bill Poduska (Prime, Apollo, Stellar,...?). Contact: Diane MacArthur, (617) 237-1121.

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.


Release 1.0 22 September 1988
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. Speeches by Henry Crouse, OSF; Jim Cannavino, IBM; Lawrence Dooling, AT&T. Call Don Berey, (212) 391-9111.

November 2-3  TISC SEMINAR ON EMERGING SOFTWARE TECHNOLOGIES - Tarpon Springs (Tampa), FL. Targeted at chief technical officers of vendor companies, with a focus on implementation of new technologies such as cooperative processing (Paul Rampel, Orion/Apple), object-oriented programming (Dennis Sisco, Stepstone), expert systems (Harry Reinstein, Aion), software repositories and development workstations (Lou Mazzucchelli, Cadre), communications (Lew Shepherdson, Simware), and groupware (Esther Dyson), both for internal use and for products and services to be resold. Presentations will be brief and assume some knowledge of the topics; audiences will be small enough to allow for extensive interaction. Sponsored by Adapso Technology Information Services Committee. Contact: John Gracza, ADAPSO, (703) 522-5055.


November 2-4  CASE user conference - Monterey. With Vaughan Merlyn and user war stories. Sponsored by CASE Research Co. Contact: Elizabeth Skowronnek, (206) 453-9900.

November 3  MICROPROCESSORS '89 - San Jose. Architects with vision from each of the major microprocessor vendors gather to discuss design, architecture, and strategy. Speeches by SPARC champion Dave Ditzel; Nexgen's Nick Tredennick, RISC critic and 68000 designer; Chris Rowen of MIPS; 88000 guru Mitch Alsup of Motorola. Targeted at people using, not building, chips. Sponsored by Microprocessor Report, an excellent newsletter. Contact editor Mike Slater, (415) 494-2677.

November 9-10  UNIX: Mainstream or myth - New York City. With Yankee's Howard Anderson and Nina Lytton; the OSF president, whoever it may be; Larry Dooling, AT&T; Jim Bell of X/Open; Max Hopper, American Airlines. Sponsored by the Yankee Group. Call Melinda Stoddard, (617) 367-1000.


Release 1.0  

22 September 1988
November 14-18  Comdex - Las Vegas. The one and only, sponsored by the Interface Group. Contact: Jane Wemyss, (617) 449-6600.

November 15-17  Neural networks and their applications - Nimes, France. A practical-minded workshop. Sponsored by a variety of mostly French, mostly academic organizations. Contact: Norbert Giambasi, (011) (33.1) 47.80.70.00.


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.


February 14-17  Software development '89 - San Francisco Airport. Sponsored by Miller Freeman, with speakers including Larry Tesler, Dick Gabriel, Terry Winograd, Bjarne Stroustrup, Ed Yourdon. Contact: KoAnn Tingley, (415) 995-2471.


March 6-10  Fifth IEEE conference on artificial intelligence applications - Miami. Contact: IEEE, (202) 371-1013, or Mark Fox, (412) 268-3832.

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

March 14-16  Interface and World Congress on Computing - New York City. Moved from Chicago, in search of more enthusiasm. Sponsored by Interface. Call Walt Heithaus, (617) 449-6600.


Release 1.0  22 September 1988
March 28-30  Fourth CD ROM conference - Anaheim. Sponsored by Micro-
soft. Contact: Min Yee, (206) 882-8131.

April 10-12  The software re-engineering symposium - San Francisco. 
Sponsored by Digital Consulting Inc. With Rich Currier, 
Panoramic; others. Contact: Dan Horgan, (508) 475-6990.

April 30-May 4  CHI '89: Conference on human factors in computing systems 
- Austin. Sponsored by ACM/SIGCHI and a host of other 
groups. Contact: Claudia Raun, MCC, (512) 338-3798.

June 11-14  Expert Communications '89 - San Francisco. Sponsored by 
Graphics Communications Association. Contact: Norman 
Scharpf, (703) 841-8160.

July 31-August 4  SIGGRAPH '89 - Boston. Sponsored by the Association For 
Computing Machinery. The annual festival for visual, 
graphical thinkers. Contact Cindy Stark, (312) 644-6610.

August 22-26  IJCAI-89 - Detroit. The international version of AAAI. 
Sponsored by the American Association for Artificial Intel-
ligence. Contact: Claudia Mazzetti, (415) 328-3123.

Please let us know of any other events we should include. -- Maria Soler

Release 1.0

22 September 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? ________________________________

____________________________________________________

CSCW 88-9

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.

Release 1.0

22 September 1988