LOTUS AGENDA

Until now, there have been two different visions of personal productivity software tools: Those that let you do anything you want if you work at it, and those that do it all for you, their way. But now there's a tool for lazy people: It does the work your way. It's called Agenda, and it picks things up by itself. Agenda does not move along the trade-off curve; it moves the curve itself, giving the user flexibility with little extra work. It's the next step in software, and the next step for Lotus.

In Agenda ($395, shipping in January), Lotus recognizes a major capability of so-called artificial intelligence and applies it to everyday tasks: It doesn't just automate tasks, it organizes tasks. You type or load "items" into your computer -- memos, notes, references, to-do reminders, e-mail, bulletin board notices or articles -- and Agenda sorts and categorizes them for you (items can belong to many categories). Then when you want all references to Alice, or to the project she's working on but not anything else she's involved with -- Agenda can present them to you.

Agenda can see patterns and take action based on those patterns, not just when you remember to tell it to, but when you're distracted and your attention is elsewhere. Agenda's edge isn't in "intelligence," or deep reasoning capability, nor in exotic forms of graphical information representation, but in alertness and consistency.

Just like a secretary, Agenda can't guess what's on your mind, and just like a secretary, it needs training. But unlike a secretary, Agenda never forgets or has a bad day. It watches your every electronic move with attention and care.

So how does Agenda do its thing? Agenda is a confusing product to understand. It is complicated, so your input can be simple. It has elements of text-search packages, elements of Apple's HyperCard (Release 1.0, 10 August), elements of Persoft's IZE (Release 1.0, 12 May), elements of hypertext, elements ➔

Inside
LOTUS AGENDA 1
The next step.
SATISFACTION GUARANTEED 5
Why not?
THE UNIFICATION OF UNIX 6
United it stands?
RETROFIT TECHNOLOGY 7
Re-engineering old software.
Bachman Information Systems.
Language Technology.
Clear: Deciphering Dbase.
NITTY-GRITTY EXPERTS 16
IBM strategy in context.
IA: implementation assistance.
NITTY-GRITTY TEXT EXPERTS 20
Intelligent Markup System.
CorrecText.
U.K. SOFTWARE 23
People & players.
RELEASE 0.5 26
Defining groupware.
Hyper-hype.
Release $1.0: The market.
PHONE NUMBERS/CALENDAR 27/28

EDventure Holdings Inc., 375 Park Avenue, New York, NY 10152, (212) 758-3434
of the new category of information/agenda managers, all of which will be used for comparison. It's like asking what a secretary does: A little of this, a little of that, and mostly what you tell it to do. In essence, Agenda creates a rich, flexible index to all the information you give it, and then presents you the view of it that you ask for. Agenda won't dial the phone, but it's good at picking out a telephone number, totaling a set of numbers in a column, finding you related memos and an address when you need to write a letter, or reminding you of all your unfinished tasks. If you like things structured, it will pick up where you leave off and structure them more; if you leave things messy, it can compensate only so far.

**Agenda vs. text-search programs: Items and categories**

In many ways, Agenda sounds like a text-search package; it has those capabilities, and more. The data you enter is stored as "items," or chunks of text up to 350 characters long, plus associated notes up to 10,000 characters long. (The entire text base consists of two tightly coupled DOS files, plus the program files.)

**Views -- the user's perspective**

**Categories -- the data structure/index**

**Items -- the individual items**

Agenda can retrieve items based on single words or Boolean queries. But it also structures the entries more interestingly to offer you more information than you explicitly entered. Each item is automatically assigned to one or more "categories." As you enter new information, either by typing or by loading it from, say, an e-mail file or a news service, Agenda examines it using key words, synonyms, and some words it knows from the start -- not just "7/14/51," but also "next Tuesday" and "a week from Monday" and "the fourth Tuesday of each month." The system assigns each item to a number of categories, depending on those key words, on where the user types them in, and other criteria.

As in some of the newer free-form databases or "trackers," each item may have a number of fields (but not all items in a category necessarily have the same fields, since items can belong to several different categories). These fields could be phone numbers; reference numbers; article titles and authors; the date, to and from fields in a message; or any other specific information that typically appears in most records of a given type. Each time you create a field or put a new value in a field, Agenda automatically creates a new category: e.g., items with date = "today" or topic = "Software Carpentry Project." Thus the promise of almost no work. Agenda can work with structured data in a structured way, but it doesn't have to.

The system also makes inferences to assign items to categories: For example, Fred Smith, chairman of Federal Express, is listed in your Federal Express category. Suppose you refer to "Fred" in some other context. Agenda might not do anything, since you know several Freds. Now you say, "Check American Express bill with Fred." The two words "Fred" and "Express" together might be enough to assign this Fred to Federal Express -- in this case, erroneously. You can set Agenda up to be timid or aggressive in coming to conclusions, by determining the extent of matching required before it detects a pattern.
Discreetly, Agenda builds an underlying index of categories and their items, but it's not alphabetical. In fact, it's like a little piece of hypertext, rich with cross-references based on key words and patterns you've selected or the system has derived. It also has hierarchies, or trees of categories and subcategories (what concerns Juan concerns the people working with him on his project). Multiple items forming different categories in this index point to the actual data elements, which are stored only once for consistency and to save space.

From the user's perspective, Agenda offers what dbms folk would call views or reports, so that you can see the data in any structure you want, with the appropriate levels and fields displayed (for the to-do list, show people's phone numbers; for the letter list, show their addresses). When you request a particular view it is displayed quickly; only the categories, not the underlying data, need to be shuffled. Like an outliner, Agenda can show you an overview, or lots of details. But unlike an outliner, it can show you specific items based on content, rather than just on their position in a hierarchy. In essence, you can have "intersecting" hierarchies -- the same items structured in many different ways -- rather than the single hierarchy allowed by a typical outliner. What's the difference between views and categories? Categories are attached to, or comprise, items in their entirety; a view is the subset of several categories and their items and fields that the user elects to see.

...vs. HyperCard: Conditions and actions

Many people will no doubt correctly compare Agenda to HyperCard, but probably for the wrong reasons. Both products use hypertext, but are not encompassed by that facility of linking items to one another. And both products serve to empower the individual user, enabling him to organize and retrieve data as he wishes (although HyperCard requires the user to define his data structures more explicitly). Either product could be twisted to perform many tasks of the other, but each most naturally behaves very differently from the other. In short, HyperCard automates procedures, or repetitive tasks, while Agenda automates ad hoc, variable responses to nonrepetitive information. HyperCard acts; Agenda helps to guide your actions.

The real power of both products lies in their programmability. In the case of HyperCard, it's a language called HyperTalk; in Agenda, it's conditions and actions, a macro language that lets you set up conditions -- values of specified fields, for example, such as "priority = urgent and topic = budget" -- and the appropriate actions, which take the form of manipulating items and categories -- "then add the item to Monday meeting."

The conditions and actions macro language allows the user greater control over the assignments -- in other words, over the structuring and manipulation of his information. If you were pressed, you could consider Agenda an expert system tool: After all, a condition-action is nothing more than a way of expressing an if-then rule. The result, or conclusion, however, is the creation of a piece of information, but not necessarily the piece of information that triggered the action. For example, a fall in your stock prices (imported from the ticker) below certain targets could display a note to call your broker (with his number) in your to-do list and export a list of the offending stocks to an external DOS file.

Release 1.0 -- Comdex Issue 28 October 1987
...vs. Persoft's IZE text-structuring program

Like IZE, Agenda builds a table of contents to your data. IZE, however, users a single paradigm to do so, which makes it more predictable to the new user but ultimately less valuable to the experienced one. (IZE generally derives its own keywords based on their frequency within the target text base; Agenda could be programmed to do so.) Rather than a simple hierarchy such as IZE builds (when the selection of keywords changes, IZE simply restructures a hierarchy), Agenda builds a more complex structure which serves as an index to the underlying items. In short, IZE takes a particular metaphor and uses it deeply; Agenda is a richer but less consistent environment. It's hard to compare two beta-stage products, but we suspect that IZE works better with large textbases because of its simpler structure. Also, since IZE's first showings last spring, Persoft has much improved its ability to interact with other applications, especially word-processors, with hot links and application-specific conversion routines.

...vs. agenda managers

All the other personal information/agenda managers we've seen so far make an unsatisfying trade-off between flexibility and power -- from the file managers and outliners that let (or make) you build your own personal database, to the full-featured programs such as Conductor's ACT!, Chang Labs' Mac-based C.A.T., and Valor Software's newly announced Info-XL. Relational databases are powerful and flexible, but they are still far more difficult to set up than Agenda and just as rigid as the canned systems, if more extensible, once you've done so. Flexible toolboxes, such as Borland's SideKick and Lotus's own Metro, are in reality just a collection of extensible tools for record-keeping, memo-processing, phone-dialing, calculation and the like. The user can make a choice among the tools, but each of them individually has its own set of pre-defined structures and functions.

What does Agenda look like? Unlike other "personal managers," it has minimal "look," with no windows, fancy graphics, or other paraphernalia. It's just lots of information listed across your screen, in lines broken with occasional columns where you want them. Do you like hierarchies with subcategories, or lists, or tables? Or sentence fragments, like this? We like the sleekness of the default interface, but some of the housekeeping screens are still cryptic, such as the format for setting conditions and actions.

...vs. perfection: What's still missing

Agenda is still early in its life cycle. One long-term opportunity for improvement (to put it politely) is to turn it into groupware (see page 26). Why not allow multiple users multiple views of the same underlying Agenda text base? This will require more than just putting it up on a LAN, of course, with careful attention to version management, security, record-locking, and other issues. What if one person's condition/action conflicts with another's?

Agenda won't fulfill its promise (if ever) until it is better able to deal with the outside world. Its target market of users -- managers, information workers, all those people who seem to do nothing and everything -- don't have as well-defined need as do spreadsheet users. And they probably don't want to be tied to using a single application -- assuming that they use a
computer at all. Agenda is ideal -- as are most tools -- for OS/2, an environment in which it will be able to co-exist more easily with other applications. Using OS/2's communications facilities, Agenda should be able to call other applications, for example, loading up a word-processor and firing off an appropriate letter to the right name and address in Agenda when a certain condition occurs. For the moment, its communication with other applications is through .STF files, much as 1-2-3 communicates through .WKS files. Agenda includes a facility to convert external files into .STF files, but that's still a little clumsy. In particular, Agenda should coordinate with other database managers as well as word-processors.

Like Lotus 1-2-3, Agenda has enough power and extensibility that users may well use it as an environment even though the system is not yet built to be used that way. Let's just hope that Lotus's next step with Agenda is to make not a symphony of discordant elements, but rather a framework that can seamlessly manage other data structures with the same flexibility that it handles Agenda items.

Where it came from

Agenda has no single creator. In early 1985, Mitch Kapor had a problem: A busy mind and a profusion of notes that he couldn't keep in order. Jerry Kaplan, then recently departed from Teknowledge, the company he co-founded, had a concept: A rich, flexible data structure with elements of hierarchy, inheritance and hypertext. And Ed Belove, then newly hired and now Lotus vp of corporate R&D, had an front-end: The simple, friendly list manager that became part of Lotus's Metro. Kaplan joined Lotus in May 1985 to work on the project with Kapor, ringing up voluminous phone bills from his home in Portola Valley and later San Francisco. Late in the summer of 1985, says Kaplan, he met with Belove and "knitted this thing together. It wasn't a shotgun marriage." The notion of the conditions and actions language, to extend the product, came a year later -- much as 1-2-3's macro language was almost an afterthought. But like 1-2-3's macro language, it is the facility that should ensure Agenda's success.

With the product substantially completed except for polishing by last summer, both Kaplan and Kapor have now left the company, Kapor to work on a series of unspecified projects including a deli and to enjoy his family, and Kaplan to found GO Corporation (Release 1.0, 21 September).

Lotus's next step

Agenda has arrived at an opportune time for Lotus. 1-2-3 has done its share for Lotus, but now it's time for the company to move on. If Agenda is as successful as it should be, doing for the nascent category of personal information managers what 1-2-3 did for spreadsheets, it will prove that Lotus is not a company built around 1-2-3, but a company built upon 1-2-3 -- an independent, viable organization that can successfully find, adopt and nurture exciting new products and even markets. Unlike a founder-managed company, Lotus under Jim Manzi has the humility -- and lack of emotional ties to its first product -- to move on happily to new fields. The irony is that Lotus's own founder is part creator of its second big product.
SOAPBOX: SATISFACTION GUARANTEED

Now that Lotus has finally dropped copy protection, it's time for the pc software industry as a whole to move on to the next stage of mature business practices in a consumer market, and offer a money-back guarantee of satisfaction to each customer. Sure it's gutsy, and it may mean toning down advertising hype, explaining clearly what a product does and does not do, and fixing bugs before shipment. But if you already do all these things, and bear their cost, why not take advantage of it -- and pressure your competitors to behave as responsibly as you do?

Check the back of this newsletter: We're doing it too. Although it has always been our policy to refund money to an unhappy subscriber, we never bothered to say so. Now we do.

The best companies in other markets make the same offer, and who wants to be anything but the best? Why not?

People will rip us off. They'll buy our software, copy it, and return it. True, some will, but those are probably the same people who are copying it already. You'll just get one less unit of revenue from them, plus some additional headache. Any person who tries this more than a couple of times will be easy to identify if you're keeping records as carefully as you should. (Such a person is unlikely to haul you into court for breaking your guarantee, but if your lawyer insists, put in some fine print. Or pay up -- the fight isn't worth it.)

People will try our software, not like it, and return it. Either you have been ripping them off, and it's time you stopped... Or you're pitching your product wrong, selling it for benefits it doesn't have or to people who don't need it. Let them tell you, not their friends. If you refund the money cheerfully, these former customers are unlikely to badmouth you, and maybe you'll learn how to position your product better. The rule of thumb is that one unhappy customer offsets ten happy ones.

We'd do it if our competitors would. Why not force them to? If you've got a product that you're willing to stand behind, use that as a competitive advantage. If you don't, well... (You can ask us for your money back. We'll return it cheerfully.)

The logistics will kill us. Companies who have tried this approach, most notably Microsoft, find they have little trouble with returns. Microsoft's returns amounted to 0.4 percent of sales on Microsoft Word, where the money-back offer was highlighted as a marketing promotion (and it still holds). Egghead's Victor Alhadeff, who is promoting the practice, says he prefers that dissatisfied customers bring packages back to the store, so that Egghead can satisfy their needs (and win their affection).

But will it really help sales? Definitely it does, says Microsoft director of applications marketing Jeff Raikes. Maybe it won't in the short run. But in the long run widespread guarantees will slow sales of unsatisfactory products (not yours, of course!), and will drive customers your way. It will also serve to raise consumer confidence in general. "It changes the dynamics when the manufacturer is willing to stand behind the product," says Alhadeff. He has increased the shelf space devoted to those vendors, including Alpha Software, Intuit, and Individual Software.

Release 1.0 -- Comdex Issue 28 October 1987
UNIX: UNITED WE HOPE TO STAND...

Sun Microsystems is on a roll, and it's trying to carry UNIX with it. From a small company meekly accepting a perceived standard, Sun has come to realize that UNIX is not a standard, but that it may have the power to fix it.

In late October, Sun and AT&T announced together that AT&T will adopt Sun's SPARC (Scalable Processor [RISC] Architecture). The two companies will also join in defining an extended UNIX standard that will be as complete as Apple's Macintosh specs or IBM's Systems Application Architecture. But, Sun president Scott McNealy stresses, unlike SAA the new UNIX standard is not several years off (although AT&T's SPARC machines are). All the components already exist -- C, SQL, UNIX V with Berkeley and Sun OS extensions, X.11 and NeWS, TCP/IP, RFS/NFS, RPC, a graphics toolbox, etc. They need only to be glued together.

That's great news for the UNIX forces. For those outside the industry, AT&T's and Xerox's endorsements of the new UNIX ABI standard lend it great credibility. And for those inside the industry, Sun's sponsorship does the same. Yet, as we pointed out last month (Release 1.0, 21 September), time has almost passed UNIX by. The unification of the different "standards" has been promised for years, and never delivered. When a questioner at the press conference pointed this out, AT&T Data Systems group president Vittorio Cassoni responded in essence, That was not my job [at the time]. He joined AT&T in late 1986 (Release 1.0, 17 February 1987), and now the promises are more concrete.

Market dynamics

The two companies presented an intriguing slide displaying the four industry standards: two identified by their owners, IBM and DEC; one semi-open, DOS & OS/2; and one open, UNIX. We might have added the Macintosh. IBM's offering was pictured as divided, DEC's as lacking high and low ends, while OS/2 tops out on the 386 (currently). Only UNIX fits all sizes...

The UNIX community's strategy will be to try to paint OS/2 into a corner -- too big for pcs and too puny for servers or workstations. DOS is great for little old pcs, they will argue, but we're the leaders on servers and workstations. Moreover, UNIX and SPARC together have the advantage of openness. The perceived taint of monopoly has passed from IBM (a former "monopolist" under attack) to Intel and Microsoft. They may sell to anyone, but they are the only suppliers of OS/2 and the related Intel architecture.

Yet this works both ways: IBM, Intel and Microsoft (and Apple) have the advantage of owning their standard, and having an incentive to invest in it. Sun has to share the fruits of its investment with its competitors. Meanwhile, UNIX is still a system favored mostly by engineers and scientists and scattered VARs. DOS developers and users will find the migration to OS/2 more natural, and most corporate applications run under some form of IBM's (future) SAA. We see no easy victories for any side, but rather a world where companies have not allies but only common interests.

But let's not forget hardware. UNIX and SPARC are not inextricably linked. How about a SPARC-specific version of OS/2 to extend its potential base? Sun and its team will surely provide 386 emulation within the next few years. As we said, a world not of allies, but of enemies' enemies...

Release 1.0 -- Comdex Issue 28 October 1987
NITTY-GRITTY EXPERTS: RETROFIT TECHNOLOGY

What’s the single biggest obstacle to the adoption of CASE technology for application development? Not cost. Not technological barriers. Not lack of integration among tools. Not even human inertia and resistance. No, the biggest obstacle is...old applications. Retrofit technology -- the new buzzword is re-engineering -- can help us find the value in those old applications rather than throw them out.

Computer-aided software engineering (CASE) tools currently assist only the programmers, designers and analysts doing 20 percent of the work -- designing new systems (80 percent is fixing or enhancing old stuff). Most CASE tools work only down the so-called life cycle waterfall -- that is, from requirements to specs to design to implementation, but not backwards, from an existing implementation to a better one. (Moreover, most of the tools automate only one or two cascades of the total waterfall.) The assumption is that software is created, used, and finally dies; new software requires a fresh start. But in fact, as Eric Bush (below) told The Wall Street Journal: "Old software never dies. It just goes into production every night."

Performance in progress: Do not enter

Long-run, if CASE systems are fully adopted, they will also help solve the maintenance and enhancement problem, because you can edit the original specs rather than the automatically generated code. (See Release 1.0, 29 May and 2 and 32 December 1986). But in the short run, which is likely to last about as long as COBOL has (since 1960 and well into the future), people won’t adopt these CASE systems wholesale, and programmers will continue to have to maintain and enhance patched spaghetti code. Of course, as every writer or coder faced with blank paper or empty screen knows, it’s easier to be an editor than an author -- but only as long as you can figure out what you’re editing. Try to grasp thousands of lines of source code, let alone tens or hundreds of thousands. The promise of re-engineering is to resurrect old code by clarifying its meaning, to breathe new life into it and let programmers and analysts reuse their predecessors’ work.

Why the problem? You’d think that if a machine can read code and database schema well enough to execute them, someone ought to be able to build a system that could explain what they mean. It’s got to be easier than understanding natural language, for example.

The cat’s cradle

The problem is that by abstracting to a higher level, you may lose implicit intentions that are not explicit in the physical implementation. In implementing, the coder (machine or human) made many choices, only some of which are relevant to the original requirement. (See box.) Consider this scenario: We have an animal builder that knows how to build a number of generic animals -- cats, dogs, ostriches. You pick an animal, and you specify a few details: coloring, weight, food preferences, behavior, and perhaps a few unique attributes that matter only to you (answers to the name Critter, perhaps, or likes to chew The New York Times). With those specs, the system can build you a serviceable cat that fits your requirements.
Now suppose you present it with a ready-made cat, and ask it to figure out the specs that led to its creation. The system can recognize cats, and so it says this is a cat. But now someone says, I want a dog with the same behavior. So it tries to figure out the requirements, not just the specs: This cat likes newspapers; does the dog have to like all newspapers, or is liking just The Times sufficient? It's slightly pudgy: Is that because the creator wanted a pudgy animal, or because he specified 18 pounds and this particular animal, what's the essential information, and what's the noise due to its implementation as a cat?

SPECS VS. REQS: A DIGRESSION

The transformation between requirements and specifications will remain the last preserve of human effort in CASE. (If you were to look at it from an intellectual property/copyright point of view, you might say that a requirement is an unprotectable idea, while the spec -- screen designs, report formats, data definitions, and the like -- is its copyrightable expression. The human creativity that went into the transformation -- the decisions about esthetics and details of how the requirement or intention should be met -- constitute the intellectual effort that the law protects. With the advent of code generators and other "automatic programming" tools, the transformation of the specs into design details and code becomes a mechanical translation into an alternative medium (not an alternative expression). That is, just as the copyright of a novel protects it even in a French translation, so is the essence of program contained in the specs, not in the code that implements them.

Precisely because that transformation is arbitrary, it cannot be performed by a machine. (If it is, it's not a transformation; it's simply a response on the order of, "Here, take this; it seems to meet your requirements.") Likewise, although a machine can read code and determine the specs it meets, it cannot go further up the cycle without some human input -- some guess as to the reasoning or preferences behind the arbitrary decisions. (If you let a machine make those decisions automatically, then the creativity resides in the determination beforehand of those arbitrary decisions.)

Of course, all this will change. Systems will get smarter at parsing natural language, and the notions of specs and requirements may merge. But there will still be human creativity needed to decide upon the requirements at least until the day this newsletter becomes Release 2.0.

In the same way, consider a programmer/analyst confronted with an existing program. The programmer can quickly determine that program is written in COBOL for IDMS; that's the equivalent of defining the animal as a cat. He can also figure out what parts of the program are doing: These bits are painting screens, those bits are describing report formats, this piece is making calculations, these lines handle records called "orders." These are the skin, hair, and bone structure, say, of our cat in the example above. At this point, the programmer has a general idea of what's going on, and he can also pick out some details. If he's been asked to fix a bug (the wrong eye color in a cat, say), he can pinpoint the relevant piece of code and make the change. That's a lot easier to do with a code-inspection tool.
such as ViaSoft's ViaInsight (Release 1.0, 2 December 1986) or with cleanly structured code such as produced by LTI's Recoder (below). But what is the essence of the program? What does it do to those order records and what are the business procedures that it handles?

How do you discern the decisions that govern handling inventory, placing orders, etc.? That behavior was built into the system in bits and pieces over years, and is not clear from the code. (That's why people like expert systems for expressing business procedures: They are inspectable, although even rules get tricky when they number in the hundreds or thousands.)

The trick is to have a system that can both parse the code and data structures to figure out what they're doing, and also recognize patterns and combinations of functions that constitute higher-level programming "cliches." Boilerplate would perhaps be a better term, but the notion of "cliches" has entered industry terminology as a result of work at MIT's AI Lab that deals with precisely these issues. Led by principal scientist Charles Rich and Richard Waters, the Programmer's Apprentice project began in the Seventies and has had a strong influence on two commercial efforts in this area.

Reincarnation routines

One is Bachman Information Systems (BACHMAN), a Venrock, Kleiner Perkins and Harvard Management venture founded four years ago by Rich, Palladian founder Phil Cooper, and Charles Bachman, creator of the precursor to Cullinet's IDMS database management system when he was at General Electric in the early Sixties. (Waters is a consultant to BACHMAN.) The other is Language Technology Inc. (Release 1.0, 21 April 1986), vendor of Recoder, a mathematically grounded tool that takes messy COBOL code, determines the flow of control, and regenerates the system in perfect, structured COBOL (and segregates the unconnected, "dead" code).

BACHMAN is just about to ship its re-engineering tools, its first products, but LTI is more established as a business and has been shipping a related product for two years now. BACHMAN is approaching the issue from a database design point of view, with plans to expand into code analysis, while LTI is approaching it from the code-analysis end, with plans to attack database design later. Speaking together at a recent CASE conference, the two firms' founders could have traded many of their slides with no disruption.

1Of course, as database structures get richer -- courtesy of the object-oriented database management systems discussed in our August issue and Bachman's own partnership model, a logical view that subsumes relational, hierarchical and network models -- more and more application logic will end up in the database, while much of the rest (screen designs, communications protocols, etc.) will be handled automatically by high-level programming tools and languages. Consider the difference between these two databases: "Rick, a casino owner; Victor, a patriot; Ilsa, his wife;" and "Rick, a cynical but ultimately honorable man deeply wounded by a past relationship with a woman he mistakenly believes to be faithless; Victor, a well-meaning patriot who loves but does not understand his wife; and Ilsa, the lady in question, torn between love and duty." The first requires a plot/application program; the second merely needs to be set in motion by putting all three elements together in Casablanca; the plot is an inevitable consequence of the database. But remember, this is the future...

Release 1.0 -- Comdex Issue 28 October 1987
The two companies will eventually compete with each other (and there's probably too much ego and sheer firepower on either side for them to merge comfortably, as some customers hope), but for the moment they have a common interest in changing the world's perception of software engineering. The software life cycle, with software falling down the waterfall, is obsolete. Meet the new life cycle of birth, aging and resurrection. The body is the implementation, but the soul -- the requirements -- lives on. Both companies are working to mine the moribund mountain of stagnant COBOL, some 80 billion lines, and give it new life in more efficient implementations.

BACHMAN: DISCERNING DATABASE DESIGNS

"The theoretical goal," says Charlie Bachman, "is to be 100 percent automatic. But that means that you could only do a part of the job. Our goal is to be useful, and so we're attempting to do far more. We're not too proud to ask the user for a little help once in a while." That is, the BACHMAN products are designed to be tools in human hands, or co-pilots rather than automatic pilots. By letting the user contribute, say, 30 percent, they can do 120 percent of what would be possible automatically.

Early next year, BACHMAN will ship the first two modules of its suite of CASE products. The first is Data Analyst, a tool to design or decipher logical data structures, independent of any particular database management system. The second is IDMS Database Administrator (DBA), an IDMS-specific tool that can create or parse IDMS Data Definition Language. Within a few months, the company will ship DBAs for DB2 and for flat file (VSAM) database design, with IMS to follow in 1989. (It will also sell the Information Model Library, a set of generic database models for specific business areas that could provide some of the "cliches" for deriving requirements.)

Basically, any design tricks explicit enough to teach to a database designer you should be able to represent as rules. These rules can then work automatically to construct a database design. For example, a designer would decide where to store the data depending on the relative number of elements or the way it would most frequently be used in the application. A good CASE tool, such as BACHMAN's, will ask about these things and make the appropriate suggestions/decisions, optionally explaining to the user what it's doing. (Those capabilities, even in prototype, have already saved BACHMAN partner McDermott a week out of a two-month database design project, estimates manager of data administration Mike Barefield.) These same rules work "backwards" when the BACHMAN Data Analyst deciphers a database design. In trying to derive the reasoning behind the design, it might ask the user, "Did you do this because...?" At the level of requirements and intentions, however, the rules and the reasons get fuzzier, and most will turn up in the application logic rather than in the database design.

Data Analyst in action

It's probably simplest to explain by relating a demo: Despite the 8 megabytes of code underneath, the system looks alluringly simple in operation. Except for some code that turns the mainframe into a file server for program source files and database definitions, everything runs on a PC AT or 386 machine on top of MS-DOS, Gold Hill Common LISP and a DOS extender, with a stunning million-pixel screen custom-built for BACHMAN. The appropriate DDL
(Data Definition Language extract) is read into the system's memory, typically from a mainframe database. The Database Administrator (DBA) inspects the low-level design and produces a Bachman diagram (as developed years ago by Bachman when he was at GE) illustrating the data records and their relationships to one another. The diagrams describe the low-level attributes of the records -- how they are stored and accessed, whether they are unique, and all their parameters as shown by IDMS. The user can select any of them with a mouse to inspect it further, seeing descriptions of the fields each record holds, its "owners" and "members" and so forth.

Justified junctions

The user can also ask for a logical view. The Data Analyst takes over and analyzes the Bachman diagram, drawing a logical view of it on half the screen after redrawing the low-level view on the other half. The DA asks questions as it works. For example, it wanted to know whether to remove a junction record. We asked the demonstrator to explain the term. As if on cue, he told us to select a help box that explained it for us: Since an IDMS database can't support many-to-many relationships, IDMS designers typically create "junction records," whose sole purpose is to handle many-to-many relationships -- for example, students in many different classes, each with many other students in it; their grades in those classes; and most complex of all, courses that are prerequisites to other courses. In each case, the system asks if it should replace the one-to-many junction record with a logical many-to-many link.

In this example, the schedule junction record was removed, since it was merely a proxy for a set of relationships between students and classes, and held no unique data. However, we kept the grades records since the grades, "owned" by both students and classes, are in fact independent entities. That is, the grades records contained discrete, additional data in a way that the schedule records did not. Finally, the prerequisites record was the most interesting; it ended up pointing to itself.

We added a couple of new data entities (one of them suggested by us, not the demonstrators) and some new relationships, and set the system to redraw the Bachman diagram. From that, we regenerated an IDMS database description, 1000-plus lines of perfect but (to us) incomprehensible syntax. What amazed us was not just its power, but its intelligibility. We have seen other CASE tools that looked very powerful themselves, but none that made us feel competent. We could figure out what it was doing, and why. (Unfortunately, BACHMAN will not yet release screen shots.)

Product plans

As soon as the DB2 and VSAM versions of the BACHMAN data tools are delivered in early 1988, the system will also be able to automate much of the database conversion process, enabling customers to transfer data from traditional databases to the more fashionable DB2 (or vice versa). By late 1988, Bachman Information Systems plans to apply its talents to doing the same thing for applications, not just data, parsing COBOL code and analyzing its semantics. Those products will be called the Programmer Assistants (PA), for IDMS- and DB2-specific versions of COBOL, and the Systems Analyst (SA), for higher-level, code/dbms-independent analysis.
Given the broader range of capabilities and meaning in an application, this is a more ambitious undertaking than the data tools. SA will have to understand at least some of the meaning of the code -- enough to say, for example, Here is all the code relating to screen design. At first, the system will simply present the appropriate chunks for redesign by hand. Long-run, that code could create the screen design itself and load it back into a screen painter where the user could alter it and send it back for reconversion into code (perhaps by a CASE tool that could rebuild it as a PC-based interface that works cooperatively with the rest of the program on a mainframe). For the moment, the SA will simply identify and describe the code, and perhaps offer syntax-sensitive editors such as ViaSoft's ViaInsight.

In general, only 10 to 20 percent of a COBOL program is anything other than such routine things as screen design, error-checking and other tasks that can easily be identified by PA. The rest is program logic that SA should be able to classify into "cliches" or scenarios by the code's use of relevant data elements and certain standard procedures: This part deals with inventory, that with order entry, credit-checking, discount calculations, etc. Beyond that comes the creative part -- understanding the reasoning behind the unique program logic that will show up as black boxes, deciding the requirement behind the spec that reflects the customer's intention. Just as there's human creativity or arbitrariness in going from requirements to specs, so must there be an understanding of that extra factor in going back.

Assisting the experts

Despite its technical prowess, Bachman Information Systems is unusually market-savvy. It uses the advice as well as the money (six figures each) of its eight development partners, including McDermott, American Express, AT&T, Aetna and Prudential, and has had them up to its Cambridge, MA, headquarters on a monthly basis since last February. Last year, with the encouragement of marketing and sales vp Steve Lipsey, himself a former MIS guy from AT&T, Bachman converted its delivery vehicle from LISP machines to 286/386 machines running Gold Hill GC LISP (quite a feat considering that Symbolics chairman Russell Noftsker is an investor). But BACHMAN didn't tell the world, says Lipsey: "As long as we were talking about LISP and AI, our potential competitors were kept off guard."

Now the company is letting the word out, with plans for a formal announcement in January. Pricing should be in the neighborhood of $20,000 (plus or minus $5000) per seat (including both analyst and administrator), little enough so a single data-oriented analyst/programmer can try to persuade his manager he needs one, or a company can acquire a large number at a discount for widespread use. Production shipments of the first, data-oriented modules early in 1988 will be followed later in the year by ones for programmers and systems analysts, who outnumber the data people by 10 to one.

On one level, BACHMAN's goal is to redefine CASE from a one-way life cycle to the re-engineering paradigm, where people and tools can enter the systems at any point and move in any direction. Based in Cambridge, the company is close to the Programmer's Apprentice project both physically and through many of its people. Yet it is not missionary-driven. To the contrary, says one of his lieutenants, "Charlie already has patents and awards coming out of his ears [and his work has made millions for Cullinet]. Now he'd like to have his own commercial success."

Release 1.0 -- Comdex Issue 28 October 1987
LANGUAGE TECHNOLOGY: CULLING CONTENT FROM CODE

Bachman, 62, comes from the dp world. Language Technology founder Eric Bush, 35, is a mathematician. That says a lot. But to amplify:

Although LTI’s Recoder itself is only a restructuring tool, the company leads the field in understanding the intricacies of COBOL, just as Bachman Information Systems has assembled a unique cadre of database experts. From restructuring -- understanding the syntax of COBOL and using it to reorder the code according to the actual flow of activity in the program -- to understanding the activity itself is a long step. However, the COBOL experts at LTI are likely better-equipped than anyone to take that step.

Bush started thinking about the problem some years ago, he says, when Richard Waters of the Programmer’s Apprentice project cited LTI’s Recoder as an example of the sort of transformation necessary for the understanding of code. Recoder, Waters pointed out, does not simply translate messy COBOL into tidy COBOL: It abstracts it into a higher-level representation (a graph) and then reimplements it in structured form. Re-engineering requires that same sort of abstraction, but taken to an even higher level.

With a few years of thought and work behind them, it should take the LTI team another year or two to come up with products, perhaps with some early modules in the meantime. Why is LTI talking about it so early? In part, says Bush, because he feels the concept needs to be sold, and he’s not afraid of tipping off competitors: “We’ve been talking about Recoder for two years now, and nobody new has come out of the woodwork. We’re more worried about having our ideas accepted than about having them stolen.” Call it arrogance, or call it healthy lack of paranoia.

One problem solves another

BACHMAN and LTI aren’t unique in appreciating the value of understanding existing code and the re-engineering it makes possible, but they appear to have the best prospects for capitalizing on it. A number of other vendors have products, including Reasoning Systems (Release 1.0, 12 May) and Clear Software (next page). However, Reasoning is still positioned firmly in the scientific/engineering market, while Clear offers a less powerful but still tremendously useful tool which turns Dbase code into flow charts.

The promise of re-engineering is huge. Maintenance and enhancement are tremendous barriers to the sale of new hardware, to say nothing of the country’s economic health. Dp managers routinely dismiss five times as many potentially high-return projects as they undertake, simply for lack of programming resources. Hardware and software vendors find their customers unable to buy more equipment because they haven’t yet installed and implemented what they already own. At the same time, those of us on the sidelines wonder why our bank statements are incomprehensible, our accounting departments can’t tell us when our expenses will be reimbursed, and the companies we hold stock in don’t know how much they’ve earned for months after the books are supposedly closed. BACHMAN and LTI aren’t about to produce systems anytime soon that can parse an application and explain in plain English what a company’s business strategy is. But they move us tangibly forward in the battle to turn the growing mountain of old code from a problem into a solution to the growing applications backlog.

Release 1.0 -- Comdex Issue 28 October 1987
CLEAR: DECIPHERING Dbase

There's no mountain yet of Dbase code, but what there is tends to be even less documented than COBOL. While CASE is invading COBOL shops, dBASE applications remain a last bastion of one-person implementation, poor methodology and the like -- even though the systems being built are becoming larger and more complex. In fact, most dBASE applications are half-coded before they're designed. dBASE authority Adam Green notes that few of his clients/students evince any interest in database design tools.

Meanwhile, there's Clear from Clear Software, a documentation tool that can be helpful without getting in the way of users' standard operating procedures. Just as re-engineering tools can derive the structure of code and some add-ons can derive spreadsheet structure ex post facto (and as Agenda or IZE can do for text -- page 1), so can Clear derive the structure of a dBASE application from the Dbase source code (not the DBASE data structures). That is, Clear can draw a tree diagram, in a printout or on-screen, showing the procedures within an application or a flow chart showing the subroutines that accomplish each procedure. Each box on the chart shows the first 30 characters of the Dbase source code it represents.

With Clear's trace function, the cursor moves through the procedure step by step, prompting the user at each decision point and taking the logical direction he indicates. Users can invoke an editor or dBASE itself from within Clear, define comments invisible to dBASE or line numbers to appear on the charts, date-stamp the charts and collect them for tracking revisions. Clear also acts as a syntax checker, and generates formatted listings.

While Clear cannot understand the deep semantics of the applications, and can't regenerate dBASE code from the user's changes in the diagrams, it can make the structure of a program visible (VisiCode?), either to the builder himself or to those he's building it for -- boss, user, or customer. Adam Green, a disinterested proponent, reports that good programs produce elegant flowcharts, and vice versa.

Clear Software was founded on a shoestring by Vadim Yasinovsky, a Russian emigre who has worked at MIT and several Boston-area software companies. Based in Brookline, MA, Clear is about to launch its first version of the product, limited to 128 symbols in a flowchart and 256 in a tree chart, at an introductory price of $99. Next year, a version with virtual memory and without those restrictions will follow at about twice the price.

Does this seem very cheap in light of the prices BACHMAN will be charging? Perhaps, but except for the underlying concept of abstracting from code everything is different. Clear should garner a larger user base at a lower unit price. The language addressed, the support level, the targeted customers, the level of abstraction and the sheer depth of the products are all at opposite ends of the spectrum. CASE is growing up and refining itself into segments, but you can have elegance and utility even at the low end.

Juan: What's an SAA-compatible 3270 interface to an old COBOL program?
Alice: Spaghetti sauce.

(For more on CASE tools, come to our Comdex session Thursday; see calendar.)
NITTY-GRITTY EXPERTS: HOW TO SELL 'EM

Complex computer systems in general and AI in particular don't need sales efforts as much as implementation support. IBM is a case in point.

In the old days, when computers commanded their own rooms and their own staffs, IBM used to work closely with its customers to build systems and applications. Not only did that win IBM happy customers, it led to the creation and ultimately the independent marketing of such stalwart products as PROFS (created with Amoco), TPF (Transaction Processing Facility, initially developed as Airline Control Program with American Airlines), CADAM (developed and sold jointly with Lockheed), and IMS (Rockwell).

But more recently, with the proliferation of third-party seminars and support and easier-to-use systems, and with the antitrust threat that made IBM loath to provide extensive free service, relations with customers became less cozy. Companies such as EDS, Computer Sciences, American Management Systems, Computer Task Group, AGS, as well as accounting firms such as the Arthurs Andersen and Young, jumped in to help. Software vendors themselves got into the support business, and VARs showed up to offer support and collect the extra margins.

Lately, IBM has been fighting back to reassert account control. While acknowledging that it needs third parties (and it does!), the company is going more aggressively after systems integration business, sometimes in joint efforts with third parties. Its Systems Application Architecture is an attempt to offer hope to users without the facilities to coordinate roomfuls of incompatible equipment, but will have its effect only as they buy or build new software and equipment. IBM is also eager to help its larger customers with direct support. Hence the widely publicized deals with Ford (office automation), United Airlines (reservations systems), General Motors (Manufacturing Automation Protocol), USAA (image-processing), and K mart (warehouse systems), among others. Hence IBM's new focus on its Systems Integration business, an offshoot of Federal Systems Division, and its cooperative efforts with such applications vendors as Lab Force (medical labs) and Hogan (banking). Hence the new Application Systems division. Hence the transfer of 6000 staffers into the field, where they will market, support, install and integrate products as well as sell them.

IBM: Internal evidence

IBM is especially active in the area of artificial intelligence, an area where its customers especially need help. If you were to generalize, you might assume that customers who were true-blue might also be traditional in other ways, and loath to experiment with arcane, risky notions such as AI. In general, you'd be right. Thus IBM's active use and marketing of AI is particularly significant, because of its potential both for IBM's own business and for broadening the market of users.

Although he's unwilling to quantify, Dr. Herb Schorr, IBM's group director of advanced systems and the man in charge of spreading the AI gospel, points to the company's seven AI support centers (Cambridge, MA; Menlo Park, CA; and five overseas, including Tokyo) as evidence of IBM's commitment. The company has also just extended its line of expert system building tools (see box). The overall aim is to establish IBM firmly in the technical vanguard,
as well as to sell more machines. In essence, the high end of IBM's market is saturated unless the company can help its customers significantly enhance old applications (see Retrofit Technology, above) and build extensive new ones. To paraphrase this thinking: We've automated the routine stuff. Now let's A) do it better and B) automate the non-routine stuff.

**IBM’s knowledge-processing line-up**

With the recent announcement of KnowledgeTool and a co-marketing agreement with IntelliCorp, IBM now offers three distinct expert system tool sets:

**Expert System Environment**, IBM’s traditional expert system shell, is now a fully supported program product instead of a program offering. ESE includes both a development ($42,500) and a delivery ($7500) component. It runs on mainframes using Pascal, and is targeted to building stand-alone advice-type and diagnostic applications. Systems built with it include IBM’s DASD Advisor (Release 1.0, 2 December). ESE is a complete, discrete environment targeted to non-dp professionals, designed for end-users who want to build and use stand-alone rule-based expert systems. Integration of sorts may be accomplished by examining other applications' files, but ESE does not (without substantial work) communicate easily with other applications. ESE forms the base of IBM’s expert systems SolutionPac, a $57,500 support-included package.

**KnowledgeTool** ($950), by contrast, is a no-frills language designed for dp professionals who can use it to build forward-chaining inferencing capabilities into traditional applications. Based on PL/1 and OPS5, it operates happily within traditional mainframe environments such as CICS, MVS/XA, and so forth, and can be used with standard IBM databases for real-time or transaction-oriented applications. ESL/1, a precursor language to KnowledgeTool built at IBM’s Yorktown Research Center, was used to build YES/MVS, an expert system that monitors the performance of MVS operating system. Many of its features are now an integral part of MVS.

For rich object-oriented data representation and modeling, IBM has joined forces with IntelliCorp to build and sell a mainframe version of KEE, IntelliCorp’s **Knowledge Engineering Environment** (Release 1.0, 26 January). Pricing and dates are unannounced, but IBM KEE will run on 370-style mainframes on top of IBM (Lucid) LISP. Built to follow SAA guidelines, it may well include a PC-based interface along the lines of IntelliCorp’s PC-Host offering.

As well, IBM is looking for applications vendors to work with. So far it has picked and announced one, Syntelligence, vendor of the Loan and Underwriting Advisors (Release 1.0, 4 August 1986).

Clearly, one way to do this is to use AI internally. IBM is currently running an expert system to configure the 9370s it is newly shipping -- some 6000 examples by the end of this year. The system is akin to DEC’s XCON, which saves DEC $40 million per year, and deals with such issues as meeting...
customer requirements, matching power supply to power load as determined by
the hardware options used, collecting the right parts, managing versions and
upgrades, and so forth. This system is built on top of ESE. Altogether,
over the last year IBM has doubled the number of expert systems in daily use
from a dozen to about two dozen, says Schorr. (See Release 1.0, 4 August
1986.)

Nor is IBM alone in using AI internally. Cullinet, for example, not only
sells its Application Expert as a standalone tool, but also uses it in guid-
ing its salespeople, with rules for how to sell against each of its competi-
tors and how to approach a variety of customer needs by function and by
industry. It has also incorporated expert configuration and set-up modules
into its general ledger and fixed asset packages to make them easier for
customers to install (and with less support from Cullinet!). MSA is using
an Aion-based system to aid in configuring software for its manufacturing
system customers. And a large number of hardware companies including DEC,
TI, Honeywell and others are using expert systems for problem diagnosis.

The new-technology marketing game

The "AI market" looked good for several years because "AI" vendors were
selling to the committed. However, the curve is bumpy, not bell-shaped, as
shown below. In the AI market stage 3 (competition) may arrive before stage
2 (market acceptance) is complete.

1 Research types and early adopters beg for your product. No need to
support it; they'll figure out how to use it for themselves. Hire
a lot of scientists or guru programmers and sell blue sky.

2 Commercial customers are interested but dubious. Show them how to
use your product, in painstaking detail. Make a missionary sell to
the uncommitted. Companies who can do this effectively, and who
can delay the arrival of stage 3 by scaring off or overwhelming
competition, can do very well. Lotus, for example, has just moved
from stage 2 to stage 3 in the spreadsheet market.

3 You've opened the market. Prepare to sell against your competi-
tion. Sales may be high, but profits and share will be low.

The early adopters are easy to sell to, because they know almost as much as
the vendors. They don't need convincing; nor do they need much support.

The next batch -- the market being addressed now in stage 2 -- is intrigued,
but nervous. This group of customers is larger, but much harder to sell to.
Interestingly, a survey of about 50 Dallas-area companies by Mind Path Tech-
nologies (which probably is not statistically defensible) shows that the

Release 1.0 28 October 1987
barriers to expert systems have shifted from the esoteric -- lack of knowledge engineers (61 percent, down to 45 percent currently) -- to the mundane -- 42 percent (against 24 percent last year) are worried about "integration with conventional/existing applications." This change in concerns reflects not just changing concerns but also changing customers -- from 54 percent in MIS and operating groups last year, to 63 percent this year.

IA: Implementation assistance for artificial intelligence

Meanwhile, expert system shell companies are taking the same tack as IBM in focusing on support and implementation consulting. IntelliCorp and Inference are finding that product sales efforts aren't returning enough on investment and are beefing up their support and consulting efforts. (Teknowledge has always had a high proportion of consulting revenues.) Artificial Intelligence Corp., meanwhile, has rounded up four corporate sponsors for its forthcoming Knowledge Base Management System, due out in 1988. "It's like selling dbms in 1975," says AIC president (and Cullinet veteran) Bob Goldman; "you've got to educate them. The reason they paid $300,000 each for a product that's going to cost $150,000 is to get a full-time consultant [and lots of implementation support] for six months."

Aion is selling to other software companies such as MSA and Boole & Babbage (Release 1.0, 26 January) who are better equipped to use the technology and to leverage Aion's expertise by using it within their own products. Aion is also working closely with Arthur Andersen, Computer Task Group and American Management Systems, who are using it to build and deliver systems for their customers. In all, about 15 of Aion's 60 employees work directly on customer/reseller support; they in turn have trained almost 100, estimates marketing manager K.C. Branscomb. However, Branscomb argues that Aion's product is easy enough to use (and built to look familiar to dp types) that Aion's greatest need now is to continue to grow its roster of referenceable success stories. (Recent articles in BusinessWeek and Fortune have helped.)

Third parties share in the spoils

Just as third parties rushed in to help IBM, so are they helping the AI vendors in what is probably a more profitable business right now than product sales. Decision Support Technology of Cambridge, MA, is running Expert Systems Implementation Group, in which a small group of companies (limited to 20) meet together with the DST staff and affiliated outside experts for two days at a time. Over the course of a year and five meetings, plus homework assignments between-times, the customers should move from commitment (to the tune of $30,000 and the time of a senior manager and a technical coordinator), to working through the building of a prototype of their own application on their home turf. With two exceptions, the participants are all names you'd know -- Kodak, Miller Brewing, Amoco, et al. Two of the best-known, with visible AI successes, decided to join the group anyway because "they felt maybe there was some luck involved and there was still a lot they could learn," says DST president Larry Meador. The first session was last month; response has been strong enough that the firm may start another yearly cycle next spring. AGS Management Systems in New York City teaches expert systems courses in conjunction with a seven-person AI consulting team that works in tandem with AGS' application and dbms experts. New Science Associates, the most visible firm in AI market research/consulting, is holding its second annual conference near Phoenix next month, and expects two to three hundred attendees.
The following, as well as the Agenda article on page 1, continues last month's analysis of text tools. The manipulation and understanding of text, combined with AI, is one of the most exciting areas of software development.

Just as it's handy to be able to look at old code and figure out what's going on (or what the developer intended), so is it handy to look at rich text (including tables and graphics, headlines and footnotes) and recognize the underlying structure, or "text objects." Humans do it easily: Any eight-year-old could pick out a table from text, point to a picture caption, find a running page header (although he might not know the formal term).

But for a machine to perceive those same patterns is difficult. If the originator of the text uses appropriate coding, such as Standard General Markup Language (SGML), this is no problem. But a word-processor, typesetter, or other text processor will not recognize, for example, a headline that is centered by means of a user hitting the space bar.

There's an answer: Avalanche Development Co.'s Intelligent Markup SYStem, or IMSYS. Its ability to discern text formats allows for the transfer of text from one text processor to another, or global changes to text elements within a document. It will also ultimately enable a machine to derive as much meaning from the structure of text as a smart human could. For example, last month we discussed how vendors preparing text databases look to cues such as headlines, citations, and cross-references to parse text for creation of text bases. This can be done either by making people look at the text, or by letting a machine read SGML codes entered by a person. Now, IMSYS can detect text objects either from scanned text or from files that use spaces, not just tabs, to delineate columns. IMSYS recognizes up to ten kinds of paragraphs (inward, hanging, or no indent, block or ragged edges, etc.); lists with up to five levels of nesting, including bullets or numbering; and so on. Avalanche is currently working with Data Retrieval Inc. to build a system that can identify headlines, authors, and other text objects as text is imported into a text base, and assign the items to the appropriate fields for easy retrieval.

IMSYS can also detect the presence and structure of tables. This capability is the basis of its first public outing in a mass-market product from Datacopy this week. It will be demonstrated at Comdex as part of Datacopy's OCR 4mat, a $195 product sold by Datacopy along with its scanner and software ($1695). OCR 4mat takes ASCII or scanned text, identifies embedded tables, and stores them separately as .WKS (1-2-3) files for manipulation by a spreadsheet. In a future release, the product may also infer mathematical relationships, so that the derived tables will contain formulas when totals, subtotals, row-by-row or column-by-column operations are detected.

The underlying technology for IMSYS is an expert system, written in C for portability, that analyzes the text by testing hypotheses (Is this a header? Or simply some standalone text?) and coming to the most likely conclusions. The company, founded in 1985 with $300,000 of private money, consists of Haviland Wright, formerly with the Coopers & Lybrand AI group; Chris Severson, a systems wizard from Arthur Andersen and NBI, Inc.; and Bill Zoellick, a key figure in the CD ROM world and co-founder of the High Sierra group. They hope to keep the Boulder, CO, company small, and to grow on contract funding from people who will integrate and resell their products.

Release 1.0 -- Comdex Issue 28 October 1987
While IMSYS looks at the location of blocks of text to discern the text objects they comprise, Houghton Mifflin Company's pc-based CorrecText looks at text word by word to find and correct spelling, usage and grammar mistakes. Like IMSYS, it pays no attention to meaning, but it does know all about words and grammatical structures within a sentence. (Remember those grammar-school diagrams showing subjects and predicates, noun and verb phrases?)

Unlike the other grammar checkers we've seen, CorrecText doesn't simply look for a list of common mistakes and do rudimentary matching of subjects and verbs. After assigning each word to its appropriate part of speech and parsing each sentence, it checks for basic subject-verb agreement and proper verb forms and then follows a number of rules to check for a wide variety of grammatical errors. The parsing techniques alone are a huge achievement; after that came years of painstaking work encoding the myriad rules of grammar and their myriad exceptions. In fact, The Insider's Guide to CorrecText, a manual that explains most of the system's reasoning (but not how it does its work so fast), is as good a grounding in grammar as we've seen in a long time. Now if everyone would just read that manual and understand it!

While it doesn't catch everything, it's far more powerful than any other grammar checker we've seen (except perhaps for IBM's slower, mainframe-based Epistle/Critique project) -- and quicker than most teachers we know. Houghton Mifflin is peddling the product to OEMs, as it does its other writing tools such as hyphenation products, spelling correctors, and thesauruses. HMCo software customers already include everyone from DEC and Honeywell to MicroPro and Microsoft (the dictionary and thesaurus in Bookshelf) and Aldus and Interleaf (hyphenation tools). CorrecText itself will be unveiled this January in a product from a large OEM customer that HMCo won't name.

The trick to avoid annoying your user, says director of product development Jonathan Press, is to make sure you don't overflag mistakes. It's better to miss a few than constantly to correct things that are okay, or at least defensible. And where there's argument, or two possible ways to do something, the system leaves the text alone, showing respect for the user. It's geared towards memos, letters and documents, not advertising copy, poetry, or LISP code. In case you're worried, it will not reduce everyone to a boring sameness of syntax. For starters, it corrects selectively, and with suggestions, not arbitrary changes. The English language is rich enough to withstand correct grammar without losing its color.

Inside CorrecText

Conceived and prototyped by Henry Kucera -- a professor at Brown University, president of Language Systems Inc. and holder of the patent on HMCo's spelling corrector -- CorrecText was implemented in C by HMCo's Business Software division. An impressive, practical expert system, it takes each sentence through several stages of parsing. (Cross-sentence dependencies such as antecedents are beyond its ken.) It begins with spelling, to give itself a sentence of real words to parse. Then it classifies each word by part of speech and attempts to resolve the function of ambiguous words, which could be used as, say, verbs, nouns or adjectives. For example, any word following "the" tends to be a noun, adjective or adverb -- except in this sentence (but the punctuation provides a clue).
With a slightly reduced realm of possibilities, CorrecText must next construct a rich tree of hypotheses for the sentence structure -- all the while knowing that one or more words may misidentified -- and then pick the parse that best satisfies its criteria. For example, take the immortal, apocryphal headline "Nut bolts and screws," about a crazy escaped rapist, with a number of ambiguous words. (See, there are jokes about grammar!) If CorrecText gets no successful parse, it will warn the user that it is looking at a sentence fragment. In some cases, it will be able to suggest the reason (no verb, for example, or a run-on sentence).

Once that's done, and it's relatively sure of what are subjects and predicates, nouns and verbs, CorrecText checks for inflections, noun-verb agreement, verb forms, pronoun forms, irregular plurals, and other syntactical items. Next, it looks at specific phrase and word problems such as:

- singular phrases that may take plural verbs (e.g., a number of);
- commonly confused words as long as they're different parts of speech so it can distinguish them (their, there, they're);
- common mistakes (different than, impact used as a verb);
- informalities (mighty used as an adverb);
- wordiness (this point in time).

Finally, it searches for cliches -- the area that comprises the bulk of other products' expertise.

In the midst of all this, at different levels, CorrecText is also checking for punctuation, capitalization, etc. What makes the problem so complex -- and requires CorrecText to be so powerful -- is that all these issues are interrelated. If this is a verb, than that's a noun; but if this is a noun, then that verb is wrong. Which is it? The soup can be art. Or, the soup can is art, if it's designed by Andy Warhol. And so ad infinitum. Pardon the sentence fragment!

Phooling around

For grammar fanatics, playing with CorrecText is like finding a fellow pedant with unlimited time to fool around. For the rest of the world, practical business folks, CorrecText is a little slower than a spelling checker, typically less than a second per sentence, but it can save them from egregious errors. Within a few years, we expect CorrecText or a pale facsimile thereof to be part of every writer's toolkit -- and every wp vendor's product line. Of course, there is still no commercial product that can find silly errors that are grammatically correct -- such as this bone.

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

2Principle is always a noun, but principal is sometimes an adjective and sometimes a noun. Even "a matter of principal" is correct if you're talking about losing your shirt on Wall Street.
OUR COLLECTED OVERSEAS

If you're a student of business strategy, you can treat this as a story of business responding to its environment. If you're a software vendor, you can treat it as a discussion of your options in moving overseas. And if you're a typical Englishman, you can treat it as a diatribe, ignore it and go home for tea.

What does a U.S. software company do once it gains critical mass? It goes overseas, of course. In the U.K., it's different: To gain critical mass, you have to go overseas first. This fundamental dilemma, combined with a general lack of avarice and greed, has served to keep the indigenous U.K. software business small and underfinanced, and to provide an inviting market for U.S. companies. Software support, by contrast, is a relatively flourishing field, in part because some of the better managements have turned their efforts in this direction, and in part because users can't easily go to headquarters for support. Quips Chris Robert, president of Corporate Software: "The value of support is proportional to the square of the distance from [vendor] headquarters."

Successful and growing as these support businesses are, they represent a distressing accommodation to a depressing lack of venture capital and venture spirit. Overall business conditions in the U.K. are encouraging: Construction cranes crowd the skylines, unemployment and inflation are down while growth is up. The computer business is also booming, as Margaret Thatcher's economy and Alan Sugar's Amstrad clones combine to broaden the market within large companies. With a year or two's time lag but good communications, U.K. companies are adopting pcs in a more deliberate manner than happened in the U.S. Fewer pcs sneak in as office furniture, and more are hooked up to networks right from the start, than in the U.S. Figures are difficult to come by, but most estimates put the U.K. pc software market at about $150 million last year, and it's growing by double-digits.

Yet managers and venture capitalists alike are hard pressed to find the sort of eager and ambitious people that crowd the highways of Silicon Valley, Boston, Houston, Seattle. ISL's biggest supplier is Softsel Ltd., the U.K. arm of Softsel Computer Products. Businessland has just bought Granada Business Systems, a high-end dealer with annual turnover around $8 million.

Recently, we spent time with three of these U.K. sales and support firms: Management Software Ltd., an Edinburgh-financed startup in London; International Software Ltd., a subsidiary of Corporate Software of Westwood, MA; and Sphinx Ltd. of Maidenhead outside London, an independent company funded with 1.5 million pounds by venture capitalists Abingworth in London, and units of Alan Patricof Associates, NEA, and Olivetti. All resell mostly U.S. software. MSL and ISL acquire most of it directly from those vendors' U.K. operations, which is how they prefer to work; with local distributors including Softsel a strong second source. Sphinx itself operates as a distributor of UNIX software within the U.K. None of them carries Mac software, and Softsel's John Vivian reports it is not a big seller in the U.K.

MANAGEMENT SOFTWARE LTD.

Management Software Ltd. is a U.K.-owned and -operated support firm founded in 1983 as a software distributor and funded by Edinburgh Financial Trust
The wholesale distribution market rapidly became overcrowded, says MSL managing director Henry Marriott, partly because of the advent of overseas competition, notably Softsel and First Software (now independent of its bankrupt parent). MSL quickly turned to supporting end-user customers directly, at a time when, Marriott recalls, Lotus offered support only from 11 am to 3 pm daily. MSL is now a customer of Softsel and other distributors, although it still gets more than 25 percent of its volume from a small number of reseller accounts.

MSL, run by former programmer (and Wharton business student) Marriott, takes a somewhat active role in trying to influence its customers' choice of products. A small company (with turnover of $3 million this year vs. $2 million in 1986), MSL stays close to its customers and eschews the use of direct mail. MSL sells quite broadly within a corporation, addressing departments and even end-users (under a corporate purchasing plan, to be sure), although in general corporate customers in the U.K. are more centralized than in the U.S. MSL has three salespeople out of a total of nine.

INTERNATIONAL SOFTWARE LTD.

By contrast, interloper International Software Ltd. is a typical U.S. creation, conceived by Corporate Software, funded in part by Hambro (an early backer of Corporate Software), and run by David Skok, a South African entrepreneur who brought his company (Skok Systems) to the U.S. and left it over disagreements with his investors over corporate strategy. Yet one of those investors, Hambro's Rich D'Amore, introduced him to Corporate Software chairman and founder Mort Rosenthal. The two hit it off, cased the U.K. market together, and founded ISL in April 1985. The financial machinations are complex, but Corporate Software now owns all of ISL, and Skok owns 3.4 percent of Corporate Software.

ISL is devoted to the PC marketplace and sells to information centers (real ones that go by that name or virtual ones that consist of centralized buying channels for groups of PC users within large companies). Like its U.S. parent, London-based ISL has grown rapidly, selling into a market where support is tough to come by. The truth is, Skok acknowledges, "there wasn't much reason for us to exist last year. There was one version of Lotus, there was DOS, there was the odd question about word-processing file conversion. But now! Now there are three versions of 1-2-3, there's DOS 2 and DOS 3 and Windows and UNIX and OS/2, there's networking! Now they really need us!" And so they do. ISL's volume is booming, amounting to about a fifth of Corporate Software's $15 million total in the September quarter.

Skok hires his people young and eager, at salaries a U.S. employer would kill for ($20,000 or so), and trains them vigorously in the products they will support. Right now he's putting his 12 technical people through some 20 different courses, with special attention to OS/2, networking, and desktop publishing (PCs only; the Mac has yet to catch on in the U.K. as it has in the U.S.). Many of these courses are taught by the vendors themselves. The company supports about 400 different products altogether.

ISL buys as much of its merchandise as it can, about 40 percent, direct from vendors' U.K. operations. Another 40 percent comes through distributors, mostly Softsel, and the rest through Corporate Software in the U.S., with a
small amount directly from the U.S. The company prefers to buy direct, says Skok, because he thinks distributors don't add much value (although they take points). In other words, software vendors' lack of U.K. operations has helped create a need for ISL's services, but a U.K. presence definitely helps vendors get software into ISL and out to its customers.

Just last week, ISL held a seminar on "Information Weapons," outlining the potential of pcs and software as competitive advantage, attended by 200 information center managers and their bosses. In fact, says Skok, business is so good that he expects to see fiercer competition soon.

SPHINX LTD.

Sphinx is a more traditional start-up -- less traditional for the U.K. market. "All my English friends warned me how tough it would be, how likely I was to fail, what a silly idea it was," recalls founder Pamela Gray, who left Zilog in 1983 to found Sphinx. "All my U.S. friends congratulated me, gave me advice, offered to make contacts." It was those U.S. friends who helped her find the products to sell, which include office automation systems from Quadratron and Uniplex, dbmses from Informix, OS and productivity packages from Santa Cruz Operation, and PC Interface from Locus Computing.

Sphinx serves the UNIX market -- customers who buy machines and applications that use UNIX, many of which are likely also to use OS/2. For the moment, these two don't compete, although that is likely to change...as in the U.S. The business operates in three parts: reselling software packages, mostly from the U.S.; software consulting for users; and software consulting for vendors. The company does a fair amount of porting and localization business for vendors, many of whom don't understand hardware or country environments outside their own. In a recent triumph of marketing meeting demand, Sphinx has founded a consortium of local software distribution companies in 10 to 12 European countries plus Abu Dhabi who will refer customers to each other. Sphinx charges a membership fee, provides product training, and picks up a reasonable amount of porting and localization work in this arrangement, which started last January. So far, 18 companies have signed up, selling UNIX applications in such languages as French, German and Danish. Sphinx's total revenues should reach $9 million this year, up from $5 million last year. Sphinx has 69 people.

Gray visits the U.S. frequently to find products, consult with her suppliers, and attend conferences. As president two years running of /usr/ group, the association of UNIX users, she is about to shock her well-meaning friends again: This trip, she's hunting for OS/2 applications to support and resell. "We're open-minded," she says. "We're not wedded to anything but what our customers want to buy." The demand isn't there yet, but the yearning is. As in the U.S., people are looking for more power on their pcs and workstations, and with a pc flavor.

RELEASE 1.1

Dan McCracken of Knowledge Systems is in fact Don McCracken. We apologize for our error.
GROUPWARE DEFINED

Workgroup computing, or groupware, will be on everyone's lips at Comdex and after. But what's the difference between old-fashioned multi-user applications running on a mini or a mainframe, and groupware? After all, those traditional applications coordinate several people working together, assign and monitor the completion of tasks, and so forth. That distinction is fine, but it's the difference between workers on an assembly line, and artisans cooperating in a workshop. Or it's the difference between a mass transit system, and a traffic system that allows individuals to travel freely but efficiently in their own cars. To be sure, there's nothing to prevent some authority from imposing rules about driving behavior that severely limit individuals' freedom of movement, or on the other hand supplying each individual with his own powerful but clumsy bus or subway car. In the same way, groupware does not automatically imply decentralization, and mainframe software can, but is rarely designed to, encourage individual autonomy.

HYPER-HYPE

It had to happen. If it's not groupware, then it's Hyper-something. With the prospective success of Apple's HyperCard, a nifty product which has some hypertext features but which is also distinguished by its programmability and object-oriented structure, a lot of old products are being spuriously repositioned. Take SearchExpress, a perfectly nice text-search package ($179-plus) for the PC released last spring by Executive Technologies. Now, release 2.0 (which has some new features with nothing to do with hypertext) is being touted as a "hypertext software package," based on its ability "to link related documents for recall with a single keystroke." That's a nice feature, and one that it has had since release 1.0, but it hardly turns the whole package into "hypertext software." True, all hypertext is is the ability to represent links -- but it's a pretty small subset of all that SearchExpress does. The new positioning is about as meaningful as describing Excel for the PC as a windowing package: Sure, it has that capability, a fashionable one right now, but that hardly defines what the package does.

RELEASE $1.0

For all those who remember when Release 1.0 covered the stock market: In all the analysis of the recent upheaval in the stock market, we missed the fundamental question -- not why the market crashed, but why it got so high in the first place. Of course it had to come down. It's just devastating that it came down so quickly and in such a way as to destroy people's confidence. Unfortunately a crazy market that did not reflect reality may nonetheless change reality by scaring people into economic paralysis.
PHONE NUMBERS

Adam Green, himself, (617) 861-6641
DeJean Melancon, AGS Systems, (212) 962-4646
Harry Reinstein, K.C. Branscomb, Aion Corp., (415) 328-9595
Vittorio Cassoni, AT&T, (201) 898-6681 (Booth 2530)
Bob Goldman, Artificial Intelligence Corp., (617) 890-8400
Haviland Wright, Avalanche Development Corp., (303) 449-5032 (c/o Booth 440)
Charles Bachman, Steve Lipsey, Bachman Information Systems, (617) 354-1414
Ken Schuman, Cascade Software, (617) 862-6246 (panel Thursday)
Vadim Yasinovsky, Clear Software, (617) 232-9788 (panel Thursday)
John Landry, Cullinet, (617) 329-7700
Bert Sheingate, Data Retrieval, (414) 355-5900
Jim McNaul, Datacopy, (415) 965-7900 (Booth 440)
Larry Meador, Decision Support Technology, (617) 354-6400
Jim Geer, Executive Technologies, (205) 933-5494
Jerry Kaplan, GO Corp., (415) 543-3200 (panel Tuesday)
Jonathan Press, Mary Sutherland, Houghton Mifflin, (617) 725-5285
Herb Schorr, Janet Warton, Bob Berland, IBM, (914) 749-3064
Alex Jacobson, Inference Corp., (213) 417-7997
Tom Kehler, InetTechCorp, (415) 965-5500
David Skok, International Software Ltd., U.K. (01) 847-3761
Eric Bush, Bill Engel, Language Technology Inc., (617) 741-1507
Conall Ryan, Ed Belove, Lotus Development, (617) 577-8500 (Booth 690)
Henry Marriott, Management Software Ltd., U.K., (01) 731-7200
Alex Tsakiris, Mind Path, (214) 233-9296
Mark Mackaman, Bill Gates, Jeff Raikes, Microsoft, (206) 882-8080 (Booth 1816)
Ed Harris, Persoft, (608) 273-6000 (Booth H8706)
John Vivian, Softsel U.K. Ltd., U.K., (01) 568-8866 (Booth 1730)
Pamela Gray, Sphinx Ltd., U.K., (06) 287-5343
Scott McNealy, Bill Joy, Sun Microsystems, (415) 960-7400
Lee Hecht, Pete Weber, Teknowledge, (415) 424-0500
Steve Sando, Valor Software, (408) 978-3044

COMING SOON...

- Connectivity: Promises, promises.
- Groupware -- software from start-ups.
- Parallel processing.
- Channels -- Micro and otherwise.
- Nitty-gritty experts: Are they intrinsically friendly?
- Retrofit technology: Gluing applications together.
- And much more...

Release 1.0 -- Comdex Issue 28 October 1987
RELEASE 1.0 CALENDAR

November 2-6

November 8-12
Data Training Conference - Anaheim, CA. Sponsored by Data Training Magazine (Weingarten Publications). Why Johnny can't compute... How to make documentation readable and products usable... For dp trainers and people whose products need them. Contact: Catherine Sununu, (617) 542-0146.

November 9-11

November 9-10
AICPA Computer Conference - Las Vegas. The 23rd annual... Serious enough to be worth 17 hours of CPE credit. General speakers: Mark Tebbe, Lante Corporation; Raymond Weadock, Plus Development; Peter Simon, Lotus, on Where is Lotus going? Contact Phil Neagle at AICPA, (800) AICPANY or (212) 575-5696.

November 9-11
Corporate strategy made visible by design - Boston. You know about intellectual property; now consider visual equities. Sponsored by Design Management Institute. Speakers include James Uehling on "The optic nerve: Expressway to corporate positioning." Contact Besty Speer at (617) 236-4165.

November 9-12
Autofact - Detroit. The event in factory automation. Sponsored by the Society of Manufacturing Engineers and other groups. With Richard Goatman (Rolls-Royce) Bill Poduska (Apollo, Stellar) and possibly Secretary of Defense Cap Weinberger. Contact Judy Evola at (313) 271-1080.

November 10-12

November 13-15
Hypertext '87 - Chapel Hill, NC. Sponsored by IEEE, ACM, and other worthy groups. Contact: Frank Halasz, co-author of NoteCards, (512) 338-3648, or John Smith, (919) 962-1792.

November 15-17
The Culpepper Forum - Atlanta. How to get the most out of your sales force. Practical sessions on techniques and trends.

Release 1.0 -- Comdex Issue
28 October 1987

November 15-19  IIA annual convention - Chicago. With Jim Treybig (Tandem), Phil Lemmons (Byte), Joseph Dionne (McGraw-Hill), and others. Sponsored by Information Industries Association. Contact: Mike Atkin at (202) 639-8262.


December 7-11  DECUS Fall '87 - Anaheim, CA. The other side of DECWORLD. If DECWORLD persuaded you to buy or you already use DEC gear, here's your chance to meet fellow users, join special interest groups, and hear technical presentations. Sponsored by Digital Equipment Computer Users Society. Contact: Mary Oskirko, (617) 480-3283.

December 9-11  Database management systems for engineering and knowledge-based applications - Santa Clara. Seminar course on object management systems led by Mohammad Ketabchi, with specific assessments of Vbase, GemStone and Iris. Contact Dr. Ketabchi at Santa Clara University, (408) 554-2731.

December 14-16  Expert systems and artificial intelligence symposium - Atlanta. With James Martin, Larry Harris (AI Corp.), Herb Schorr (IBM), and others. Sponsored by Digital Consulting, Inc. Contact: Lisa Mosca, (617) 470-3870.

January 11-12  Neural networks - Los Angeles. A commercial assessment of NN applications, with most of the major players, followed by a separate three-day course on the technology. Sponsored by the Institute for International Research. Contact: Russell Webb, (213) 883-1770.


February 8-10  IFIP conference on computers and law - Santa Monica, CA. Issues that just won't go away: Copyright, contracts, taxa-

Release 1.0 -- Comdex Issue 28 October 1987
tion, computer crime, legislative actions. Sponsored by IFIP and Los Angeles County Bar Law and Technology section. Contact: Michael Krieger, (213) 208-2461.


February 21-24 ELEVENTH ANNUAL PERSONAL COMPUTER FORUM - Naples, FL. We moved it in search of variety and better weather. Registration forms will be mailed to subscribers shortly. For further information, please call Sylvia Franklin, (212) 758-3434.


February 25-27 Workshop on technology and cooperative work - Tucson, AZ. Sponsored by Bell Communications Research and the University of Arizona. Contact: Robert Kraut, (201) 829-4513 or Jolene Galegher, (602) 621-7477.

March 1-3 Third international CD ROM conference - Seattle. Sponsored by Microsoft. Contact: Sherrie Eastman, (206) 867-3305.

March 7-10 IEEE conference on computer workstations - Santa Clara. Sponsored by IEEE. With Sun's Bill Joy, and sessions on distributed systems, computer-supported cooperative work, and OS/2. Contact: Pat Mantey (408) 429-2158 or Robin Williams, (408) 927-1842.

March 8-10 Connect '88 - New York City. Sponsored by Cahners, with Datamation and the Gartner Group. A trade show on connectivity and integration directed at corporate end-users. Contact: Richard Malden, (203) 964-0000.

March 14-18 Artificial intelligence applications - San Diego. Sponsored by IEEE. Contact: Richard Greene, (301) 468-3210 (exhibits) or IEEE, 371-0101 (program) or Paul Harmon (415) 861-1660.

March 16-23 Hannover Fair CeBIT - Hanover, West Germany. Contact: Donna Peterson Hyland, Hannover Fairs USA, (609) 987-1202.

March 20-23 ADAPSO SPRING CONFERENCE - Palm Desert, CA. Software and services vendors at the oasis. Contact: Sheila Wakefield, (703) 522-5055.


March 27-30  Software Publishers Association spring conference - Berkeley, CA.  Contact: Jackie McDonald, (202) 452-1600.


April 7-10  13th West Coast Computer Faire - San Francisco.  Contact: Jason Chudnofsky at Interface Group, (627) 449-6600.


April 11-14  AIIM show - Chicago.  Information and image management.  Sponsored by Association for Information and Image Management.  Contact: Sue Wolk or Betty Garrett, (301) 587-8202.

May 9-12  Comdex Spring - Atlanta.  Peaches and PCs.  Contact: Jane Wemyss at the Interface Group, (617) 449-6600.


August 22-26  AAAI-88 - St. Paul, MN.  The seventh annual.  Sponsored by the American Association for Artificial Intelligence.  Contact: Claudia Mazzetti, (415) 328-3123.

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

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

October 23-28  Monterey Classic - Monterey, CA.  Contact: John Baumeister, (408) 987-4200.

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

Release 1.0 -- Comdex Issue 28 October 1987
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 refunded.

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

EDventure Holdings Inc.
375 Park Avenue, Suite 2503
New York, NY 10152

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

Sylvia Franklin
Associate Publisher

Release 1.0 -- Comdex Issue
28 October 1987