METAPHOR: WORTH WAITING PHOR

Take a vertical market business plan...please. Add to it the concept of segmentation by function, not just industry -- brand manager, branch operations manager, merchandise buyer -- with software specially tailored to do the job. Decide that the only way to do it right is to develop your own optimized hardware, rather than using an IBM PC with one of those highly touted micro-mainframe links. Next, target Fortune 10x-class companies and build up a direct salesforce so that you can effectively control the quality of the sales. Finally, hire a cadre of crack systems analysts so that you can customize applications for your users and establish firm account control. This is the plan of Metaphor Computer Systems.

Metaphor's strategy isn't that special. What distinguishes the firm is its no-compromise attitude. At this point in the plan most people -- and their venture capitalists -- would be exploring the use of value-added resellers to reduce the heavy startup costs of fielding a direct salesforce, incorporating Lotus 1-2-3 templates to reduce development time and borrow some of Lotus's sales sizzle, and certainly bowing to the ubiquity of the IBM PC. They might decide to reduce the specificity of their marketing in order to address a broader customer base...

But Metaphor is run by two firm-minded veterans of the Xerox Office Systems division (Don Massaro and Dave Liddle) who have made careers of resisting even reasonable compromises. The result is one of the classiest -- and costliest -- startups we've seen in years. Funding totals $15 million, with hefty contributions from H&Q, Kleiner Perkins, and Oak. Based in Mountain View, CA, Metaphor reminds us of another dare-to-be-great startup, Tandem, which risked everything on a non-standard solution it could prove superior. (Of course, there have been other such risky startups that didn't make it, but we've forgotten them.) The nub of Metaphor's business plan is that direct salesforce: If they make their numbers, the company will be a brilliant success. If not, well...

The system itself

Accordingly, Metaphor has done its best to create an attractive product. Where Apple is aiming for an appliance and IBM for a powerful tool, Metaphor offers an apprentice -- a machine that learns the user's job and then handles it. Systems are configured and customized to perform specific tasks on installation; thereafter, the user can easily generate his own applications (or "capsules") with a programming-by-example feature.
Metaphor's architecture foreshadows a direction many will take in the future. It eschews the now fashionable micro-mainframe link for a much closer connection between the data and the user. Instead of such a link -- which depends on little bits of information downloaded from a remote, busy mainframe -- Metaphor establishes an intermediate data base akin to Cullinet's Information Data Base. (That reminds us of the cartoon where two scientists are dubiously examining a blackboard covered with abstruse equations. At the bottom of the board is the notation: "And then a miracle occurs." Yes, that's easier said than done. But Metaphor offers software that makes the job of stripping the data from the mainframe much easier, translating it into the right format for Metaphor's applications rather than merely transferring it in raw form. The process still requires the involvement of a dp expert -- and of Metaphor's high-rent support people -- but once the procedures are established, the intermediate data base can be updated automatically at the dp manager's behest.

That data base appears local to the user, for he is tied to it with a local area network (Ethernet) that gives him access at the speed of a hard disk.

Why is all this so important? It's nothing that you couldn't do with a 1-2-3 model. True, but there's a point where a difference in speed becomes a difference in capability. With 1-2-3 you can have speed with limited amounts of data, or you can have all the data that you want (and that dp will allow you) with unpredictable delays due to competing demands for mainframe time. With Metaphor you can have virtually unlimited amounts of data, so that you can focus on different aspects of your business according to what's of interest. (The data base servers start at 72 megabytes and can grow up to 2 gigabytes each; the system handles consolidation of files located on different physical servers.) As a rule of thumb, says marketing vp K.C. Branscomb, she looks for customers among people with huge volumes of inches-thick weekly printouts marked up with highlighter -- the kind of stuff that is meaningless to anyone who doesn't know a business intimately, and hard to wade through even for those who do.

Metaphor's system is not for a sales manager tracking the performance of her seven salespeople with 18 products and a few thousand customers; rather it's for a person like the Delta Air Lines pricing manager profiled in a recent Wall Street Journal story (who is not a Metaphor user). Each morning he receives reports on hundreds of price changes on hundreds of routes throughout the country. This is the kind of task Metaphor would be suited to handle: the interaction of pricing, passenger load factors, hourly, weekly and seasonal variations in loads, with occasional peaks due to local events such as football games and Comdex, and so forth.) Now how can that kind of person keep all that in a PC spreadsheet model? He can't. The data may be available in the printout, but next week is too late... He has to be able to combine different parts of several substantial data bases on the fly. Metaphor can do that on a few seconds' notice.

What can Metaphor do with the data? It can perform the usual spreadsheet functions, perform statistical regressions within the spreadsheet, combine different series of data: For each route where we compete with Eastern airlines, retrieve load factors and average yields per mile grouped by time of day and fare classes. Now do a regression analysis... Results can be graphed, formatted into reports, and spliced into documents and memos. The user can also set up a variety of queries or other routines he likes to use frequently (capsules), and edit them as necessary. (Suppose we look only at the routes where our pricing is 15 percent or more below Eastern's?)
The Metaphor window above, labeled "Query Control," shows the query of a user who wants to see account descriptions, center numbers, and actual and budgeted amounts for each account from numbers 4 to 25 in centers 1200 and 1400 only, for fiscal periods 2 and 3 separately. (Account descriptions are kept in a separate table from the rest of the data.) That's hard to describe elegantly in English, but easy enough to display on the screen.

With a huge, Xerox Star-like screen, the 68000-based system can easily display several full 25x80 windows. Within those windows are all the usual productivity functions -- graphs, word-processing, electronic mail, etc. However, Metaphor retains an open-minded approach and provides special routines for easy communication with IBM PCs, among other foreign bodies. The user may provide his own electronic mail to communicate easily with non-Metaphor systems. Despite the non-standard operating system, at least the system does run MS-BASIC for application creation or porting, and Xenix will be available next year.

Thou shalt not recreate data

Metaphor's raison d'etre is to derive meaning from data. Aside from producing reports and graphs, it also displays diagrams of the various relationships between data that the system is using, posing the implied question: Here's what you're asking for; is that really what you want? (See illustration above.) The system works on the principle that data, once entered, should not have to be retyped. Thus the user can point to almost anything he needs with the mouse, and uses a function keypad (for those who eschew the keyboard) to move, copy or delete items, open or resize windows, and ask for an "options" menu. Using existing data, a user can actually go through an entire session without once touching the keyboard.
Worth paying for

So, is this a simple system? Not exactly. Most of the people who use it perform complicated analyses, and they still have to know what they're doing. Indeed, it's precisely because they're such smart, knowledgeable -- and expensive -- people that improving their productivity is worth the cost of the Metaphor system, typically about $8000 per workstation, including support and data bases.

Once a contract is signed, Metaphor sends in its troops for about 15 person-days all told of network configuration, data definition, and application installation. Customized applications are extra, at $1000 per application per site (not per user). Typical pricing of an initial installation is above $100,000, but Metaphor will make its real money (if any) in selling additional systems into a base of existing users (again, much like Tandem).

The company's greatest strength -- and its greatest exposure -- is its huge 40-strong corps of sales and support people, currently covering New York, San Francisco and Chicago, and scheduled to increase to 100 within a year. They will face a tough market, for despite Metaphor's data-import and other capabilities, the company's workstations appear as invaders to a corporation standardized on IBM PCs or IBM mainframes and terminals or any other vendor's system. Yes, the system works, the initial customers like it, Arthur Andersen will support it. But will the salespeople make their numbers?

Well, we're hopeful. In the long run, Metaphor doesn't replace a pc; it's what the pc is going to be. It addresses the naive user who is nonetheless a sophisticated analyst. He knows all about his data, and would prefer not to know about his computer.

MARVIN MINSKY TO SPEAK AT 1985 PERSONAL COMPUTER FORUM

In furtherance of our recently instituted campaign to enlighten the pc community about AI, we have been lucky to secure the services of Marvin Minsky as the dinner speaker at the 1985 Personal Computer Forum in Phoenix. Marvin Minsky is the well-known AI pioneer who currently works out of MIT's Artificial Intelligence Laboratory.

And no, you haven't missed your invitation in the mail; it will be sent out later this month.

RELEASE 1.0, September 11, 1984
INTRODUCTION TO EXPERT SYSTEMS: A NON-EXPERT'S DIARY

Until recently, artificial intelligence has been a hardware-intensive, mostly impractical technology of greatest interest to academicians, while personal computers have been limited-hardware machines with cryptic software usable only by fans. But both disciplines are converging on a single marketplace -- the naive user -- and will ultimately join into one discipline. The Xerox 1132, after all, is simply one of the world's most expensive personal computers -- except for the Cray machine operated single-user-mode by Gerry Demos of Digital Productions in Los Angeles, who uses it for film animation.

We have spent the last month (with time out for a trip to Dallas) collecting facts about some AI companies and their products. What we found was intriguing enough that our exploration will continue. But in the meantime, here's an initial trip report -- with perspective.

TWO PC-BASED EXPERT SYSTEM BUILDERS

We recently spent a week at Teknowledge's four-day course for purchasers of its $12,500 M.1 expert system generator (RELease 1.0, June 18), which runs on the IBM PC. We also spent some time both in Austin and New York playing with TI's Personal Consultant ($3000, with training for an extra $1500), which runs on the TI Professional Computer. The Personal Consultant is similar to M.1 but oddly enough is more closely based on the Emecin system -- an early expert system created partly by several of Teknowledge's founders. Both M.1 and TI/PC allow a semi-sophisticated, non-programming user to create an expert system consisting of several hundred rules expressed mostly as "If X, then Y."

Unfortunately neither of the two systems has any automatic way of checking for rules' consistency or validity; rules can be miswritten so that they're never used (or "fired" in AI parlance). Suppose you specify that phone calls from California can be returned until 8 pm and you specify that calls from California must be returned by 7 pm. It's a toss-up as to which rule is fired (depending on its location in the program). Moreover, a phone call from Palo Alto will not be so handled unless you remember to specify that Palo Alto is in California -- or to have the system ask the user if it is. However, they do allow you to "trace" a consultation, watching as rules are fired and either applied or rejected.

TI's system, as befits a pc vendor that has also done some other user-friendly pc software (notably the NaturalLink English menu system; RELease 1.0, February 22), pays more attention to the user as opposed to the programmer interface; the end-user can select items from a menu, where Teknowledge makes him type them in. M.1 will at least show him what to select from, although if he displays that list he also has to watch the rules firing and succeeding or failing in succession; there's no way to see part of M.1's innards without seeing them all. Moreover, if you ask M.1 why it concluded something, it will probably answer with a cryptic set of rules. It takes a fairly complicated set of procedures to change that into something comprehensible in M.1, whereas TI has a semi-automatic translation function for humanizing the user's interface, although the builder still has to supply the translation -- e.g.

"because the caller specified no deadline and the caller is not calling about an important matter."
(One could easily change the definition of "an important matter" as important matters come and go.)

On the other hand, Teknowledge's people attempted to improve on the fundamentals of their work with Emycin, and added enough bells and whistles that their system can handle fewer but more complex rules. For example, TI's Personal Consultant lacks variables; although the parallel is not exact, imagine a VisiCalc or Lotus 1-2-3 without the replicate function. Indeed, variables are even more valuable, since a spreadsheet replicates and stores the formula -- relative or absolute -- each time it appears, while the variable function enables M.1 to store a rule or expression applying in a multitude of situations only once, saving valuable space in what is typically a tight hardware environment. (TI does have a poor man's version of variables called context hierarchy, but it's not as powerful.)

TI's system can address the entire 768K maximum of the TI Professional Computer, whereas Teknowledge made the (nonbinding) choice to limit M.1 to 128K so as to address a larger market. However, M.1 does have overlay capabilities so that chunks of rules can be shifted in and out of memory.

In the end, the capacity of the two systems is probably close enough that each system's character is probably a bigger issue. M.1 is for experimenters, sold partly in hopes of upgrading users to Teknowledge's high-end S.1 system (see below), although the company has been surprised at the practical uses to which M.1-generated systems are being put. Meanwhile, the TI/PC was intended for builders to create user-friendly systems for real end-users (and sell TI Professional Computers into the bargain). Indeed, TI offers a runtime license for $25,000 upfront plus $25 per copy, while Teknowledge (for now) has only per-unit pricing. But TI also has some interests in the high end, including an investment of under 25 percent in Los Angeles-based Lisp Machines and a rumored AI computer of its own.

If you took this course, then you would conclude...

Although this is hardly news to your typical expert system expert, what impressed us most during the course was the arbitrariness of an expert system's design. Like any other program, an expert system can be written well or poorly (quite apart from the validity and completeness of the rules it embodies). There are millions of ways to write one wrong, but also thousands of ways to write one right. A moderately well-written system will execute slowly; a really poor one simply won't work; a brilliantly written one will run quite fast. But circumstances change and a well-written system may perform poorly on certain problems. Optimization matters!

The fundamental lesson of the week is how painful it is to create an expert system. It may be fun to be an expert talking about one's job; it's less exciting to transform all that expertise into a complete, rigorous set of consistent rules; the intellectual challenge soon wears off. (Of course, it's certainly no worse than coding, say, a payroll system.) The M.1 system has no pretensions to being an end-user-oriented system, and has few cosmetics. (The systems it generates, however, are intended for use by a naive user.) Although it provides the tracer and a number of other debugging tools in a selection of brightly colored panels, the basic work is still deadly.

The problem with generating expert systems in the pc environment is that until a problem is fairly complex it's easier to do it yourself -- just as many poor
managers do rather than delegate work to eager but unschooled juniors. At the moment expert system generators are of about as much interest to end-users as the C language; it's nice to know it's around so that programmers can write applications for us. But the ES generators have the advantage of being able to create a certain kind of extremely flexible, application that gives answers in terms of advice, diagnoses, etc., rather than just data.

Yet both these pc-based systems are indeed like programming languages; what we want is a development environment, with preprogrammed functions, debugging facilities, etc. They make one wish for a host of add-ons.

Hoping for a complete development environment on a pc right now is overreaching. The important point is that the runtime systems generated in such environments be able to run on pcs. Unlike human assistants, a pc expert system never gets bored or sick, or wants to be promoted. Moreover, these machine juniors can be cloned. One great value of expert systems is their consistency. Calls get handled the same each day -- regardless of whether we're cheerful or gloomy, present or not present -- unless of course we specify that on busy days certain calls are to be delegated that otherwise might be answered directly. There are other ways to get consistency, of course, such as company-wide expense account forms or 1-2-3 models; expert systems extend that consistency into decision making.

text continues on page 8...

| Rule: If a company has four or more founders, then it is likely to be an AI company. |
| Company     | Number of founders | Comment                                      |
| Intellicorp | 4                  |                                              |
| Quintus     | 5                  |                                              |
| Silologic   | 2                  | Some companies with fewer also qualify.      |
| Symantec    | 2                  | See page 13.                                 |
| Teknowledge | 20                 | Eighteen of them are from Stanford.          |
| Xerox       | 1                  | Xerox is only partly an AI company.          |

Release 1.0, September 11, 1984
STRUCTURE OF A KNOWLEDGE SYSTEM

As little children will no doubt be learning by rote in school decades hence, a knowledge system consists of an inference engine (the data- and rule-processor) and a knowledge base. The knowledge bases described here mostly have three parts: data, rules, and control rules (or procedures). The data are static items -- objects, attributes and values of attributes, which have relationships to each other. Large hierarchies of data and their relationships can be built; the structure defining each item and its relationship to others is called a frame. (Data can also be represented as rules; e.g., If a call comes from Palo Alto, then it is from California. But it's simpler just to define Palo Alto as one of the locations within California -- especially if you want to list a large number of cities.) The rules themselves are typically encoded as if-then statements. Facts and rules can both be modified by certainty factors -- as can conclusions.

Control rules govern the operation of the system, managing the order in which questions are asked or rules tested. They also handle things like translations, telling the system, for example, that to determine the location of Palo Alto it should display this message to the screen: "And what state is Palo Alto in?" (To get it to display the message for any city, the system builder can use a variable for the city's name.)

Finally, there's the user's data. The system proceeds methodically, applying its rules and factual knowledge to the user's data to determine new facts and applications of its rules. Note that the system doesn't "understand" the rules; it merely applies them. Take the statement, "If red is white, then green is blue." A goal-seeking system will attempt to test the hypothesis that green is blue, and will do so by back-chaining (as governed by the control rules). It will seek values of red to find out if red is indeed white (and therefore green is blue) -- perhaps by trying other rules of the form "If ..., then red is white," and so forth. A data-directed system, on the other hand, will start with the fact that red is white, and forward-chain to the fact that green is blue. Then it will look for rules of the form, "If green is blue, then ..." and so forth, until all the true rules are used in order to reach a final conclusion.

Frequently the system will solicit data to test its rules -- the famous question-and-answer sessions typical of expert system consultations. Those answers, in conjunction with the data within the system, enable the system to come to final conclusions, such as "Answer this call right away!" -- translated from an internal conclusion such as "time_to_answer = immediate and not(delegation)."

How does a knowledge system differ from the generally-used term expert system? Basically, a knowledge system consists of easily verifiable facts, whereas an expert system is more subject to argument (as is the opinion of any expert; one wag suggests we should call expert systems "belief systems"). A knowledge system can be as productive as an expert system, since either can be applied to real problems; moreover, most real-life problems probably require just knowledge, not expertise. (Expertise by definition is rare, poorly-defined knowledge.) A system to assess structural soundness would be a knowledge system, using facts about physical properties of materials, structural engineering, local weather, etc., whereas house design would also need an expert system, using someone's strongly held opinions about roofing materials, the proper number of people per bathroom, preferred cost trade-offs, etc. Ultimately, the two might interact to determine such issues as window size and placement. Some things you can find in books, and some things everyone just knows... So how many bathrooms do you have?
TWO HIGH-END ENVIRONMENTS

Several high-end expert system development environments were on display at the overcrowded fourth annual conference of the American Association of Artificial Intelligence (Menlo Park, CA), held in Austin, TX. What used to be a gathering of the clan had suddenly become a public event; with attendance of 3100 rather than the 2500 expected. Yet the academic flavor still reigned. What any other group would have called the show floor of booths was demurely described as the research and development exhibit area. But the exhibitors were mostly dressed to kill and primed to sell.

Walking around the booths, surrounded by mice, windows and bit-mapped monochrome screens, was like entering a foreign country where the normal rule (that IBM rules) doesn't apply. But perhaps it will. A number of IBM PC-based systems were in evidence, and notable among the exhibitors was IBM itself. Known for jumping in only after others have tested the waters, IBM was showing LISP/VM for its mainframes, along with five other R&D exhibits.

Compare...

But the most elegant systems by far were Teknowledge's S.1 and IntelliCorp's KEE. Xerox was showing LOOPS, a collection of development tools for the 1100 series available at cost and without support (much like UNIX a few years ago) from Xerox's AI systems business unit in Pasadena, CA, as opposed to its AI research operations at the Palo Alto Research Center.

KEE, for Knowledge Engineering Environment, is IntelliCorp's first product, announced a year ago and now in its second release. KEE costs $60,000, with prices dropping to $20,000 and below for additional units -- demonstrating the high proportion of the price attributable to support. KEE's system is implemented in two different brands of LISP (we'll get into that another time), running on the Xerox 1100 line, the Symbolics 3600 series, and the LMI Lambda. S.1 is Teknowledge's high-end system, available for $50,000 with follow-on units for $15,000. It also runs on the Xerox 1100 series in LISP, and will soon be available on the DEC VAX. Both companies' prices include several weeks of training.

Both S.1 and KEE provide the rule-checking, data-import and other facilities -- in short, the intelligence -- so lacking in the PC-based systems. In addition to their rules-generating capability, both let the user enter data in frames, with "inheritance" (although S.1's frame capabilities are limited). Inheritance lets one describe the general properties of members of a class so that each member of that class will "inherit" them. If cars have engines, and a Porsche is a car, then a Porsche has an engine. But the user doesn't need to go through this rule. He simply describes cars, and indicates that Porsche is a car. Such a set of relationships is a "frame." The use of frames can save lots of time over the use of rules.

Although they are machines and recognize only logical inconsistencies, not factual ones, both systems use the superb graphics of their LISP hardware to diagram data relationships, which makes inconsistencies (often betrayed in crossed lines) or irrelevancies (missing connections) easier to spot. Both systems use several windows, so that, for example, one can trace (in text) the rules fired during a question-and-answer session. Either system lets the user examine all rules concerning a given item, edit frames, trace and replay the system's actions (to see why an unexpected conclusion was reached, for example).

Release 1.0, September 11, 1984
...and contrast

One of the nicest features of KEE alone is its ability to make a similar display of the relationships among rules (see illustration). This display will also enable the user to pick up fallacies that the machine may not be programmed to catch and that he might not notice in the usual one-dimensional listing, which has all the impact of "...and then I tried this rule, and then I tried that rule, and then ..." Unfortunately there's no infallible expert system yet that can properly check out an expert system not just for inconsistencies, which can be detected, but also for unclear thinking. Did the user build the system he intended? If not, he may never know, as long as it's logically consistent. Visual displays help to reduce this kind of problem. All in all, KEE is distinguished by its heavy -- and enlightening -- use of graphics. The new release allows the user to develop his own visual images of data -- using diagrams that alter shape or size as appropriate, perfect for a process-control application, for example.

But the biggest difference between the two companies is one of philosophy. Palo Alto-based Teknowledge, under chairman and ceo Lee Hecht, schooled at the University of Chicago business school and founder of another software company, is most interested in getting the job done. Up the road at Menlo Park-based IntelliCorp, newly installed president Gene Kromer, formerly with TI, has turned the place into a profit-oriented, more focused outfit, but he hasn't taken away its fascination with technology. Technically, the KEE system is more elegant and broader, ready to tackle almost any problem with a variety of reasoning and knowledge representation methods (frames, rules, graphics, etc.), while S.1 is designed specifically to handle efficiently a more narrow class of rule-based, structured-selection problems. While KEE lets you appreciate the elegance of what you're doing, Teknowledge attempts to do more of it for you.

Thou shalt not recreate data

Expert systems, just like VisiCalc or word-processing, won't be of much use until they can easily accept data from existing data bases, whether IMS on an IBM mainframe, the specially structured files of a bank accounting system, or a user's carefully constructed VisiCalc file (which may itself use data from a mainframe). This avoids errors in data, provides consistency, and above all avoids an amount of data entry that would otherwise render many systems unusable. Both systems have limited capabilities of importing data, but that's more along the lines of an ability to write code that will do a query to a remote data base, rather than true integration with existing data bases. KEE is also fairly well designed to use what it calls "active values," power plant water levels, for example.

Another item on the wishlist is runtime versions of these systems. Neither KEE nor S.1 is likely to be ported to a PC anytime soon, but both companies could well benefit from Xerox's rumored announcement of an AI workstation that would sell for less than $10,000. Their expensive existing hardware is necessary to support the high-level capabilities of KEE and S.1, but such a low-end workstation would presumably offer an ideal vehicle for runtime versions to run the systems built in those high-end environments.
SOME DISPLAYS FROM KEE

How? This shows the logic that was used to reach the conclusion.

What? This shows some values displayed as graphics.

RELEASE 1.0, September 11, 1984
TWO PROLOG COMPANIES

What's the difference between LISP and Prolog? Well, LISP is older, more like assembly language, slow to write and fast to execute, powerful and complex. Also, simply because it's older, it happens to have associated with it a host of powerful development tools. Prolog, by contrast, is newer, less flexible but more powerful, with a single statement taking the place of several in LISP. LISP is procedural; Prolog is declarative in nature. (Prolog's name comes from logic; in brief it is a language to describe logical relations, such as "X is a member of the set Y," and so forth.)

Both of these languages should be invisible to the expert system builder, who wants to write his system as rules and data descriptions, not as code. As it happens, because of its complexity and long existence, LISP has been used as the foundation of a number of expert system generators, notably Xerox's LOOPS, IntelliCorp's KEE, and Teknowledge's S.1 (M.1 is written in Prolog). Now a similar body of tools is beginning to grow up around Prolog. Also, just like LISP, Prolog is starting to creep out of academia and AI machines into business and onto familiar workhorses from DEC and IBM, among others.

Spearheading this move is Quintus Computer Systems of Palo Alto, CA, selling Prolog as a standalone language with add-on features like graphics and database links to come later. Quintus comprises many of the top names in Prolog development, which started in Europe. Among the founders is senior vice president of engineering David Warren, who developed the first successful Prolog compiler at the University of Edinburgh in the mid-Seventies. His co-authors Lawrence Byrd and Fernando Pereira are also Quintus co-founders, along with IBM veteran Cuthbert Hurd and Bill Kornfeld, an expert in both Prolog and LISP who has optimized Quintus's Prolog code for speed. Marketing vp is Carolyn Morris, formerly of HP and of her own startup. In addition to the DEC 10/20-based Prolog it is already selling, Quintus hopes to announce Prolog for some more mass-market-oriented machines this fall.

Meanwhile, Silogic of Los Angeles is downplaying the Prolog connection (although the name is derived from Silog, its Prolog dialect). Silogic is designing a Prolog-based development system to run on top of UNIX on any 68000-based micro. The so-called Knowledge Workbench includes its own knowledge base generator and inference engine, an interface to a variety of commercial databases (both relational and standard IMS types), and a natural-language interface. This all sounds very nice -- note that it's not yet delivered.

Silogic's people, the US stars of the Prolog community, have been at work for about two years and hope to have a product to show this winter. Rather than selling its system as a mass-market development tool, Silogic is working on joint marketing agreements with a number of large resellers who will use the workbench as the foundation of a working expert application to be resold (at a premium) to end-users. Silogic's new marketing vp Al Barkovsky (from Computer Sciences' end-user-oriented Infonet division) right now is concentrating on a number of unidentified financial institutions, although he's not turning anyone away. (Perhaps it could use Metaphor?) This approach will enable Silogic to remain a small company and husband its scarce support resources, and will allow the system to be more powerful and less "finished" than would otherwise be necessary.

Silogic realizes there's a lot more leverage in improving the productivity of tens of thousands of, say, bank loan officers, than that of hundreds of expert system builders.

RELease 1.0, September 11, 1984
builders. Last week we had lunch with salesman Barkovsky and Silogic's new president, both brought in last spring by the company's investors, mostly local Hollywood types dabbling in high tech. President Fred Braddock, 46, comes from Informatics General, where he lasted 15 years — almost as long as Mark IV, the product he designed for Informatics back in 1967, when IBM was still giving software away. Mark IV over the years has improved the productivity of thousands of IBM mainframe programmers; total sales by now have exceeded $125 million. In all those years, Informatics' customers experienced huge gains in productivity that they never shared with Informatics. Now, selling applications in conjunction with partners who know particular markets, Braddock hopes to make the kind of vertical-market margins that systems software-oriented vendors rarely achieve.

**BUT WHAT IF YOU DON'T KNOW THE RULES?**

For those who read our January 23 issue and wonder where Expert-Ease fits in:

There is a fundamental difference between the Expert System generators we've described and Expert-Ease, the much-heralded expert system generator from Scotland now selling over here. The systems described above take the user's rules and then apply them; Expert-Ease takes data and deduces rules. The problem with Expert-Ease is that the rules it infers may be nonsense if it has too few or even just fortuitously misleading data points. On the other hand, in many cases the rules may be unknown; the express purpose for which E-E is used is first to derive and then to apply those rules. Expert-Ease, with new support from company president Sandy Blackie who has immigrated here to run US operations, seems to be making headway in the US market. More on that later.

(For another, more financially oriented description of the brave new world of AI, we recommend the excellent report by Ken Sonenclar of the Gartner Group in Stamford, CT.)
TALES OF SOFTWARE

Early one Saturday morning we were walking through Washington Square in New York City sporting a canvas PFS bag. Working nearby was one of New York's nearly-indigents, pushing a shopping cart which he was filling with beer cans and soda bottles, detritus of the night before worth five cents apiece in deposit refunds. Espying our bag, he called out, "What's PFS?" "A computer program; you know -- software," we answered. "Software!" he shouted in glee. "You can make a lot of money in software. Forget hardware!!"

Safety-Pin Software

Our first reaction to mention of PFS:plan was: Who needs another spreadsheet? But our reaction to the product itself was different: If 1-2-3 is a sewing kit for people who love to sew, then PFS:PLAN is a safety pin for the rest of us.

Although one could legitimately argue that the product is about two years late, so was PFS:write, announced in the spring of 1983 and now running neck and neck with the high-end, $495 MultiMate for first place in word-processor sales. PFS:plan does bring something new to the spreadsheet business: simplicity. Indeed, it fulfills the promise of the cryptic, somewhat misnamed VisiCalc by truly making calculations visible: PFS:plan's columns and rows are governed by the formulas that appear optionally at the tops and sides. A PFS:plan spreadsheet will show, for example, that entries in the top row "grow by 15%" per period, and the fourth row is the "Total" of the first three rows. In a prerelease version, row and column calculations couldn't be disabled: i.e., a row or column had to be consistent all the way through. That was a little too simple and inflexible (like the single-margin setting which prevails through a PFS:write document), and now the user can designate a cell to have its own formula. Relative and absolute replications are handled automatically in the way that makes most sense; users can adjust the models as they see fit. Clearly, PFS:plan is not for the spreadsheet whiz. But for the user who handles numbers gingerly and would like to see what his model means, PFS:plan is ideal.

C&E/Symantec: And then there was one

Wealth encourages waste. Nowhere is this clearer than in the pc software business, where tremendous resources, funded mostly by venture capital, not profits, have been applied to the development and marketing of fundamentally indistinguishable products. Most of these resources -- development skills, money, management, marketing -- have been spread dangerously thin. Few companies have the necessary critical mass in all these areas, yet new companies are continually springing up, bleeding existing companies of the few resources they have.

Countering this distressing tendency is the recent merger between Symantec (RELease 1.0, April 14, 1983) and C&E Software, joining two incomplete companies into a single, well-endowed one, with an infusion of about $1 million in new venture capital. Symantec disappointed many (us among others) by failing ever to deliver its dbms front-ended with a powerful natural language interface, ceding the laurels in this market to Microrim's R:base 4000 and its Clout NL interface. Founder Gary Hendrix, a leader in natural language, basically took on too much in attempting to design a dbms as well as a front-end to it. However, the company did pick up a new president and ceo along the way -- Symantec's own senior vice president Vern Raburn, who had left Microsoft in 1982 to manage the successful
launch of Lotus 1-2-3, and then left Lotus for the usual "personal reasons" in the spring of 1983. Right now Symantec has a great team but no product, and it has used up most of the $3.5 million it raised from Kleiner Perkins and others back in 1983. Sales vp Don Farrow, in an arrangement displaying commendable flexibility, does consulting four days a week to pass the time (and pay the bills).

But Farrow may have to go back to work pretty soon. In a merger sponsored by Kleiner Perkins, Symantec will be acquired by C&E Software, the creation of Gordon Eubanks and Denis Coleman. Eubanks wrote CBASIC and sold his company, Compiler Systems, to Digital Research. He lasted at DRI as vp and general manager for two years before leaving last fall to found C&E with Coleman, a longtime friend who had written Spellguard and sold it to Sorcim. At C&E, Coleman was the architect of a set of application development tools designed to build a suite of low-end personal productivity applications; those tools will come in handy in merging those applications with Symantec's natural-language interface. As noted, Raburn is president; Eubanks is chairman, Coleman is vp of development, and former chairman Hendrix remains vp of advanced technology. The new company, based in Cupertino, will keep the name Symantec.

Easing the pain of home finance

Suppose someone handed you a fancy new vacuum cleaner with several new cleaning tools that enabled you to reach nooks and crannies previously inaccessible. Yet after using it for a while you found it took you longer than before to get your vacuuming done. How delighted would you be?

According to president Scott Cook of Intuit of Menlo Park, CA, that's about how delighted most people are with their home finance software. It lets them do more things that they don't really want to do, and slows them down in handling a chore they'd really rather not handle at all. True, there are a few finance hobbyists, people who enjoy balancing their books, constructing budgets, and testing whether their insurance coverage is appropriate, but most of us would rather take out the disk and replace it with Alluring Aliens or even turn the computer off altogether.

Cook found all this out when doing consumer research for his new product Quicken. A veteran of Procter & Gamble where he handled Crisco shortening, Cook has applied the P&G formula to home finance. The P&G approach, he stresses, is not just advertising and brand management, but research and unique consumer benefits. (For example, contrary to popular belief, the chief benefit of disposable diapers is not convenience for the parent, but rather dryness for the baby. P&G successfully sold this benefit in Pampers, and you know the rest.) Cook has now applied the research approach to home finance, and found that the chief benefit is speed rather than features. In other words, people want to do as little as they can get away with, as quickly as possible.

Hence, Quicken. Running on the PC, it's extremely simple and fast, reports our publisher Rona Levine. (It can also be used by small businesses who don't want to do much more than keep a check register, maintain their balances, and set up a few payment and receipts accounts.) Six-person Intuit, which Cook founded, is privately funded and can't afford huge marketing outlays. Instead, Cook has teamed up with a large retail-oriented bank that will help sell the Quicken software and the special checks that go with it; they're especially designed to be easy to align in the printer. The aim is not to improve home finance software; the aim is to lessen the burden of home finance.

RELEASE 1.0, September 11, 1984
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. Multiple-copy rates on request. Subscriptions are available on an advance payment basis only.

Name

Title

Company

Address

City    State    Zip

Telephone

Please send payment and this form to:    EDventure Holdings Inc.
                                      375 Park Avenue
                                      New York, NY 10152