DATABASE ALL OVER THE PLACE

Have you heard about Universal Database yet? It’s going to provide a seamless distributed database management system that will be absolutely transparent to everything. No one but the system itself -- not the user, not the applications -- need know where the data is, where the resources are, where the processing is taking place. Universal will take care of all that. Of course, to reach this paradise you must put all your data into Universal’s data dictionary and write all your applications in Universal’s language, which is a superset of SQL and fully supports the ANSI standard. Long-term, it will support IBM’s DB2 and yes, even its competitors’ products.

In fact, the Universal dbms is going to be so transparent you’ll never see it. It suffers from a common but fundamental fallacy. By the time Universal gets around to supporting its competitors’ products, they’ll have changed. In other words, global distributed database won’t ever proliferate as long as we have a healthy, growing market of competitors obsoleting their own and other vendors’ products, and making internal changes that cannot be easily accommodated by UDB in its seamless distributed database.

Distributed database assumes a homogeneous world where all data is available to all comers. The more realistic client-server architecture, by contrast, acknowledges that one side asks for data and the other side supplies it. The roles may be reversed at any time between transactions. Each time data crosses the interface between systems, there is a transaction, a clearly defined sequence of events that provides for error recovery, security, completeness, etc. Integrity is maintained within each sector by keeping everything perfect -- no redundancies, no inconsistencies, perfect knowledge. It’s maintained across systems by transactions, where one system treats another as an outsider. If you define the relevant territory as the entire db, you end up with massive problems in maintaining integrity, no provisions for local rules, etc. Each time a change is made anywhere the system must check everywhere for side-effects. At a time when we are adopting object-oriented approaches for programming, it’s ridiculous to move towards the opposite -- seamless integration -- in databases. Instead, we need well-defined, clear seams, which will isolate local errors and let us make continual improvements without wholesale commotion.

TRANSCRIPTS ARE IN BETA -- COMING SOON!
Anselm's Work Engine is an unusual system that makes observers talk about object-oriented programming, expert systems, hypertext and the like. It doesn't qualify precisely as any of these; instead, it takes the most appropriate techniques from each concept and melds them into a system replete with hybrid vigor. These techniques include forward chaining, rich data structures with inheritance, and modularization. Individual "objects," or case histories, are specialized from generic decision trees; they are manipulated by an expert-system-like linking tool that maps the specific content of those objects into various work products -- reports or documents or explanations of decisions. While some might argue the religious issues, we prefer to concentrate on the Work Engine's powerful capabilities.

If you wanted to be anti-visionary, you could describe Anselm's Work Engine as a glamorized mail-merge package (and it works fine in that capacity). However, it's also a tool for automating text-based tasks of substantial complexity, where the input is pre-configured work processes combined dynamically with real-time information and stored variables, and the output is documents, reports, and decision trees that describe a situation or represent a user's decisions. (Which would you rather sell?) Or you could call it a forward-chaining expert system or even a hypertext application.

What's it good for? To the naked eye, it's a document assembler. To the smart user or the clever marketer, it's a way to automate repetitive work processes and to record the decisions made in each instance of a task. Suppose you bring a new employee into your company. Here are the questions you ask, which vary according to her responses, and here's the print-out about benefits, working hours and other items that you give her to take home. And here, finally, is a record of the new hire's answers with appropriate supporting text. Those answers can also be used to help produce an appropriate letter when it's time for her six-month review, when she asks for a part-time job to go law school, or when she's fired.

With the usual caveats -- it needs someone to set it up -- the Work Engine can easily create a system to manage this entire process. With a little more work by the user, it can integrate with a corporate database, existing corporate text boilerplate and other electronic information. Areas it could manage include clients with legal or tax problems, users with support problems, travelers with holiday plans or even hypochondriacs with real and spurious medical complaints. It could also control a telemarketing dialogue and serve to generate follow-up letters and customer support files. Indeed, it could even assemble code modules in response to a specification elicited in a dialogue with a customer.

The Work Engine is currently in use at a number of law, CPA and architecture firms, where it helps manage client affairs and cases, generating the appropriate documents and records. In prospect are a communications software vendor who will use it for over-the-phone customer support and an instant print shop which will use it for order-taking.

---

1The first such program we saw was LegalWare's Document Modeler (Release 1.0, 86-10). Two years later, OM is still a tremendously functional program, but it lacks the flexibility, modularity and extensibility of the Work Engine (although, as a Mac product, it has a far nicer end-user interface).
Anselm is a publishing company, an extension of Kinetic Computing, a development company co-founded in 1986 by Jim Baker, formerly software development director at legal publisher Matthew Bender (a Times-Mirror subsidiary). Anselm will launch the shrink-wrapped development version of the Work Engine this fall. It will cost about as much as a pc database, says Baker, or $500 to $700. Later this year, Shepard's, the McGraw-Hill legal publishing unit, will launch a licensed run-time version of the Work Engine for wills and trusts. ("The template author and I both think run-time only is a mistake," says Baker, "but we'll let the consumers change Shepard's attitude.") Meanwhile, Kinetic will keep looking for specialists in other fields who could create flexible work templates and resell the package into other markets.

Job templates

The Work Engine comprises three kinds of objects which are combined variously with "instance" data (about clients, problems, cases or any other substance of a business unit) to produce new objects including client files, session histories, documents and records, and to manage interactive sessions with end-users. The builder-user, who need not be very technical but must understand the problem and the relevant procedures, sets up the three sets of task-specific information as follows below. A job template consists of a set of objects from Columns A, B and C that will accomplish a particular task, such as generating a will or a financial plan, when combined by the Work Engine's application logic with the appropriate case data from Column D (either entered in real-time or called from existing records).

Column A -- the screens

A set of data-entry screens, or questionnaires. They include response screens that provide additional information and follow through on a particular line of questioning. ("If you own a second house you may be able to take extra deductions. Do you rent this house out for extra income?") For example, the first six data-entry screens may be fairly straightforward, designed to elicit the basic biographical information. But on the seventh question, a yes to a question about previous experience with LISP will lead to a series of questions about the nature of that experience, while a no sends the user on to the next branch and set of questions. (This sequencing is managed by the lists in Column C.)

The questionnaire screens elicit the "instance" information (Column D), a database about all the individual cases, with information such as names and addresses, financial profile, etc. Each case also produces a decision-tree-like record of the choices made in response to the screen's yes/no or multiple-choice questions, which control the flow of screens and create the "shape" of the case decision tree.

The data-entry screens vary from forms with many fields, each identified separately by the system, to mostly-text response or instruction screens ("You said your espresso machine doesn't work. Have you closed the lid? Here's how...."). The screen field IDs also serve to identify the instance data, by instance and by field, just as data in a relational database is identified by key value (row) and field (column). You could use a standard database as an information store, but this works for now, and a standard

2We're using the "columns" metaphor simply to make this easier to explain.
dbms couldn't easily handle the user choices that are also stored with the case record. As needed, the builder-user can create links to traditional database files, to avoid redundancy of information and the attendant errors.

Column B -- the boilerplate

A set of text blocks. Just as in traditional mail-merge, each hunk of text has its own unique ID, which typically has some meaning to the builder-user -- such as the systems that lawyers use for numbering paragraphs, or phrases such as "child3" or "mstatus." The texts are the organization's boilerplate: form letters, paragraphs and clauses from legal documents (usually already numbered for easy identification and ordering), detailed instructions or help text, diagnoses and prescribed activities, code modules. Existing boilerplate needs only a new label intelligible to the Work Engine to be used by it.

The text blocks, as in mail-merge, contain dynamic pointers to variable information from Column D such as names, addresses, amounts and other case particulars. Eventually, these text blocks, with appropriate information inserted, will be selected and strung together to create letters, reports, policy manuals, etc., customized for individual situations and users.

The same boilerplate text block can be referenced by a number of different work templates: Grieving widowers, corporate raiders, attractive divorcees and orphaned children all get the same expression of undying loyalty and concern at the end of the letters Haynes & Haws sends them.

This approach saves space as well as work, because the resulting templates and text are stored only once rather than with each set of client data. However, the merged files can be stored whole for further manual editing.

Column C -- the links

The linkers -- or what Baker calls the lists. They contain the specific task logic, the sets of links that point from one data-entry screen in Column A to another and that assemble the appropriate sets of text blocks from Column B to produce the work output according to the design of each particular task. The text blocks, in turn, can call information (data or decisions) from Column D or from a database as in regular mail-merge. Note that each list generally contains logic used both to run an interactive session and to sequence the text blocks for output, where it operates somewhat like a UNIX make file. In a totally interactive session with no external output (as opposed to internal data capture and decision-tree records), there are no links to text blocks; in a batch application, there are no links to data-entry screens.

The linkers take the form of a decision tree, which makes things relatively simple, but the builder must design the structure carefully. This part of the Work Engine, in fact, is a pre-compiled, limited, forward-chaining expert system. Unlike a full-scale expert system, however, it is rigid and executes in a defined sequence, rather than sitting in the background and processing new data whenever it appears. (You might also, if you wanted to stretch a point, call this a sort of hypertext application.) The parts of the tree that get instantiated in each case depend on the choices made during an interactive session and stored in a case record in Column D.

Release 1.0 18 July 1988
Note that these decisions can be stored for reuse later by another linker list. Depending on the task (or list) selected, you could generate a will or a financial plan from the same case information. The same client information from Column D, for example, could create both a report for the client and a second, more explicit report for the salesperson handling his account, with additional information on subsequent purchases appended. Yet a third version could be produced for another partner in the firm who has to deal with that same client on another matter, and doesn't want to see the text (all 132 pages!); he just wants to see the factors that went into it. Or a user could apply those same decisions against a file of French boilerplate. Or he could later apply a wholly different list, using data from a subset of several case records, to construct a totally different work effort, perhaps for a divorce or for client billing.

Column D -- the data

The interactive sessions with end-users conducted via the data-entry screens produce both specific data (names, etc.) and individual decision trees, or records of the choices made during a session (married or unmarried, interested in growth or income, etc.). As noted, the resulting instance data is not just a set of values in fields; the series of answers forms a unique, tree-like data structure, a subset of the parent linker decision tree, which varies in "shape" according to the specific facts or choices in each case or instance. Rich as this structure is (much like a traditional network or hierarchical database), it is uniquely suited to the particular applications it is built for; it's difficult to get the data out of it for use elsewhere.

Putting it all together

Columns A, B and C are built by the builder-user to make a work template, while end-users supply the case data for Column D. Then there's "Column E," the Work Engine code itself, which executes the logic in the Column C linkers that combines the task knowledge in Columns A and B with the instance data in Column D.

The work that the Work Engine produces can vary widely. On the one hand, the system can operate mostly in interactive mode, as a worker (for example) gathers information over the telephone and uses a sequence of data-entry screens to guide a conversation. Most of the text -- talk about product benefits and the like -- and the effort resides in the data-entry screens.

In other cases, the main purpose of the system may be to elicit a couple of pieces of information -- name, address, and order data -- to generate several different work products: a picking slip for the shipping department and a bill for the accounting department, with a brief sales report and a summary of customer preferences (and objections) for the sales department's files (perhaps with a follow-up date attached), and perhaps some data to populate a database.

By contrast, in a legal office, a very small amount of information may be used to produce huge volumes of paper from text blocks for which the client will pay dearly. In a doctor's office, the information could be used to produce a suggested diet and exercise program for the patient, and a medical history for the doctor. In another office, a new worker could generate a printout (or online text) describing common office procedures or the rules
for holiday leave. Cleveland-based WidgetWorks gives four days’ vacation after a year, while San Jose-based World of Widgets gives six days, plus a membership in the Decathlon Club; each has its own policies (different linkers), but needs similar information about employees to carry out those policies (same questionnaire screens, but different policy-explanation text).

Task automation

The Work Engine is a relatively comprehensible tool for building task-automation templates -- in short, the kind of system we've been awaiting for years, one that uses a variety of new programming approaches to good effect. The Work Engine provides the executing code; the builder-user provides the task logic and the text content.

Of course, it was hard to recognize at first, and hard to describe since there aren't yet specific words for many of the things it contains or does. There are two reasons for this. One, we don't have enough implementations yet to abstract out a good specification of the concept, or better yet, to build a generic task-automation tool with all these attributes. Moreover, the Work Engine is still more rigid and less abstract than we'd like.

Second, perhaps there is no generic task-automation tool. As system-building tools get closer to reality (from concern with computer system issues), they will become inherently less generic, as they have to model discrete real-world objects and processes -- clients, interviews, decision-making, referrals, documents -- that are specific to certain kinds of tasks.

So how does the Work Engine square with the new programming paradigms of objects, expert systems and the like? The system itself is written in C for DOS in its initial implementation, but it should be easily portable to other environments given its emphasis on function as opposed to form, including a text-only interface for maximum portability. Among the generic approaches the Work Engine uses are:

- objects and inheritance. The Work Engine uses data objects that are combinations of data and structure -- the structured decision records in Column D. Each instance inherits or specializes its structure from the generic decision-tree linker in Column C that created it. However, while rich and structured, the data is still inert, with no assigned methods as in the object-oriented paradigm.

Moreover, while the specific structure of each column D decision record is unknown to the system's linker applications, its general type and parent decision tree are. So there's not much of what the scientists consider "information-hiding" in the Work Engine. In fact, it's a carefully crafted system whose parts fit together tightly with strong interdependencies.

- modularization of knowledge and data-driven execution. Think of the Work Engine as layered horizontally instead of vertically. Although the system does not have object-oriented encapsulation, so that logic is linked to data, Work Engine's knowledge is modularized into a variety of locations -- some in the decision trees that manage the flow of data-entry screens and text blocks, some in the screens and text blocks themselves, and some in the case-specific structured records.

Release 1.0	 18 July 1988
It has moved one step beyond the database paradigm -- separation of data and applications -- to the separation of applications into data-driven control flow and the code that executes it.

- **inspectability.** Anselm’s decision-tree approach makes the logic behind the information easier to understand -- and to modify. In practice, it doesn’t quite look like a tree, but like an outline -- considerably easier to understand than the nested structured-programming, IF-THEN-ELSE systems typically used for this purpose.

- **text orientation.** Like many interesting new applications, much of the Work Engine deals with text and richly structured data objects rather than with numbers and short strings. Much of its content -- texts and screens -- is meaningless to the computer, but with IDs and logic it can assemble them in ways meaningful and valuable to people. However, Work Engine could be far more powerful with a better way of managing and classifying its data -- an object-oriented database, for example -- that would allow it to define the information it needs as object classes with specified characteristics rather than specify the exact text chunk or data-entry screen that contains it.

**Not just reusable, but resellable**

The value of the Work Engine will lie in the knowledge builder-users will put into it -- stored in questionnaire screens, text blocks and decision-tree lists. Some of the information will be company-specific, but much will be generic (or at least customizable) across a broad array of potential users. Within companies, the Work Engine will also contain valuable, reusable information about individual cases. The Work Engine is a medium to store the wisdom that underlies many standard business procedures and policies and deliver it back in accessible, reusable form. The screens, texts and decision trees can be examined directly by a knowledgeable user, and can be enriched with comments to make them more understandable. The different components, as noted, can be reassembled for a variety of work products.

The value of all this is obvious: It’s a way to gather rich, mostly textual information, and use it in a variety of ways. It extends the basic ability of databases to store data permanently for access by multiple applications and people. Yet the rich structure of the Work Engine’s data is somewhat constricting, and could be better handled by object-oriented database.

**Setting the precedent**

In its first forays into the marketplace, Anselm/Kinetic is sensibly taking the path of least resistance. This is certainly the right initial approach, but the real potential lies in the transfer of the technology into other fields. The separation of knowledge about particular cases from the routine, procedural knowledge for handling myriads of combinations of generic situations will enable us to transform huge amounts of ad hoc, case-by-case work -- the majority of what goes on in offices today -- into routine, automated processes, freeing people to use their minds on the really hard stuff.

(Or to go home early?)

*Release 1.0* 18 July 1988
IBM’s Systems Application Architecture is widely perceived as a massive tactic -- a defensive response to Digital Equipment’s across-the-board VAX strategy and the misled perception of UNIX as a single standard. In fact, SAA represents a fundamental strategy and a vision of a CASE-based environment -- an environment within which other software vendors must operate.

Two basic tenets of SAA are (1) a set of programming languages and communications interfaces that will operate across multiple environments and (2) DB2/SQL, a standard database structure and query language. While many vendors are trying to solve the problems of heterogeneity by eliminating the heterogeneity with a tightly defined architecture, IBM acknowledges the world as it is and proposes a more radical approach that will depend on the use of powerful computer-aided software-engineering (CASE) tools (such as that of Transform Logic, with which it has just signed an agreement).

CASE will provide application vendors and users with hardware independence. The old argument -- we don’t use tools so that we can optimize our products -- no longer holds water in a world where continued development requires more expensive resources than the extra hardware needed to overcome minor execution overhead. If you can define your application at a high level and rely on automated processes to optimize and implement it for a specific environment, heterogeneity is no longer a problem. Indeed, this approach is wide open to future hardware developments -- insulating the application designers and the CASE tools (via compilers) from the changes while letting them benefit from the performance of new architectures. That same argument will work as we move on to parallel platforms, RISC architectures and object-oriented systems such as Intel-Siemens’ BiiN. This does not happen by magic, of course. It simply shifts the burden to the hardware/language vendor, who must provide compilers for his system to make it marketable.

Moreover, if we can rely on CASE tools and SQL-based query tools, there’s no need to write for specific environments or use fourth-generation languages (4GLs) to give users an easy way to write their own applications and queries. (4GLs suffer from both poor performance and database and hardware dependence, a combination of disadvantages usually found as trade-offs.)

The arguments are slightly different in the case of database: While CASE can also insulate you from the database with compiled applications, only a relational dbms such as DB2 can easily support ad hoc interactive queries or easily manage its own reconfiguration. The use of SQL adds generic support of front-end tools and applications.

The trick is to insulate people not just from static differences among environments, but also from differences across time, which are inevitable as vendors’ lines evolve. In times past, customers typically relied on a single vendor, and so they were less interested in portability and interoperability. In the current environment of turmoil, the layered approach has benefits not just for vendors, who can offer a product line for multiple environments, but also for customers, since it insulates them from their own multi-vendor, changing environments. In a world of good CASE tools and compilers and a standard relational dbms, the current fuss over hardware and OS standards may come to seem irrelevant to commercial customers. The focus will shift to user-interface issues and application content.
Whither the value?

Indeed, the next locus of commerical software value-added is going to be function- and market-specific applications, built with CASE tools for maximum customizability. For example, customers no longer regard accounting as a separate back-office function where the only possible benefit of accounting packages is cost reduction. They're unresponsive to the blandishments of vendors offering off-the-shelf applications. Instead, they want more comprehensible billing, designed their way; integration of order information with sales efforts; proactive inventory management, rather than mere after-the-fact recording of materials flows; etc. In other words, customers want custom applications, but they don’t want to build them from scratch.

How can we build applications in a way that their logic is reusable within a CASE environment? Easily, by building them with CASE tools from the start, and then selling the tools along with the applications so that customers can maintain, enhance and re-engineer them themselves. This is the approach of Walker Interactive, a traditional accounting application vendor, and Cullinet Software, which began in the database and report business.

Indeed, the difference between CASE-customizable applications and application-specific CASE is only a matter of perspective. From the other end of the spectrum, vendors with significant industry-specific experience will distill that experience into application-specific CASE tools. One likely candidate for leadership here is Arthur Andersen, which has just fielded its own CASE line, Foundation, a productization of the tools it has been using internally (Release 1.0, 88-4). Over the years AA has used Foundation and its forebears to build thousands of systems for customers, and learned a good deal about certain markets. We expect to see more and more such specific knowledge wind up in application modules or templates that AA will sell to the appropriate customer sets as part of the Foundation suite.

The land that CASE forgot

On the other hand, there's a whole world that CASE as currently constituted misses. Software engineering as an intellectual pursuit and the CASE marketplace have very little in common. In the common view, software engineering is about specification languages, transformations, provability, artificial intelligence, object-oriented programs and databases and other esoterica, while CASE is about building database-based applications in COBOL. It's a pity the two don't meet; it's a waste to spend so much time and effort building trite database applications when the real action is in designing new systems in ways that haven't been used before. In fact, as Cadre Technologies demonstrates, there's no reason you can't use CASE tools to model object-oriented designs; as EDS has shown with Apache and FCMC's Tony Kobine wants to show with Calypso (Release 1.0, 88-6), you can also use CASE tools to model work flows. As Cullinet is about to show, expert systems can be used both as part of the CASE environment and embedded in the systems CASE produces, while Aion already has an expert system development tool that is half CASE, half ES tool, and runs on both mainframes and PCs.

Yes, it's true that we're currently using CASE just to build customizable database-based applications that will soon become commodities. But let's not abandon CASE; let's abandon our limited vision of what it can do.
STRATEGIC CASES: IN THE CASE OF ACCOUNTING...

While Computer Associates is eager to take over Management Science America, the vanilla mainframe accounting market is losing its appeal. Mainframe accounting applications are so standard now that users don't write them anymore; they buy them (and they buy them infrequently, as replacements). The only benefit a vendor can bring to the party is price or customization. Price is a painful game, as shown by MSA's results. Customizability, on the other hand, is driving customers to IBM's DB2 and other SQL dbmses, and to CASE. Two vendors in particular -- Cullinet and Walker Interactive -- are addressing this demand, through a mix of CASE tools and appropriate design decisions, although both have troubles of their own. Cullinet comes off a couple of disastrous years in the database market and several management upheavals; Walker paid off the last of its pre-1985 debts last spring.

WALKER INTERACTIVE...REACTIVATED

"Call us a restart, not a turnaround," says Walker Interactive president Dave Brownlee. The company's strategy was years ahead of its time back in 1981, when it entered the financial-software package business after 12 years in the consulting business. Unfortunately Walker's implementation was so inadequate and its optimism so excessive that it almost lost the chance to prove the worth of its strategy. Now, the timing is just right for the revived firm to be one of the first SAA-compliant vendors, and to assist customers in the move from IMS or flat files to DB2 and other SQL databases.

The story thus far...

Walker moved into the package business with what it calls Strategic Software, incorporating the Software Bridge, a software layer that allows the company to build applications independent of underlying hardware and database systems. On the strength of its people and its ideas, the company raised $21 million from the likes of Adler, Accel Partners, Kleiner Perkins and TVI. But instead of building a couple of products that worked, Walker overexpanded, underdelivered and spent all the money the investors had put up. In 1985 it arranged to pay off its debts at 36 cents on the dollar, which it has now done. The investors brought in turnaround artist Bruce Coleman from Informatics. He got the company to focus on basics and turned it over to Brownlee, who had run European operations. In the first half of this year, the company made $1.3 million (it will pay no taxes for a while!) on revenues of $7 million, serving 210 active customers.

Now read on....

Accounting is fairly boring stuff. You can't go out and find new customers as you can for, say, PCs or personal information managers. Anyone who wants a beyond-PC-size accounting system probably already has one. So how do you find customers? The time to catch them is when they're changing anyway -- either from one set of hardware to another, or to a new database. They do this for a few reasons -- growth, inability to implement the systems they want, or acquisitions of incompatible systems along with the companies that own them. In the market's coming changeover to DB2 and other relational dbmses, Walker has found a bird's nest on the ground. Walker estimates that IBM will get 400 new DB2 customers in the U.S. this year -- not a bad target market for Walker at $400,000 to $600,000 per initial sale.

Release 1.0

18 July 1988
Walker can help these people over the hump. If they buy Walker's accounting packages, they can later convert their financial records from IMS to DB2 with a minimum of disruption. How can you take advantage of DB2 if you're not ready yet? Buy Walker now, and use DB2 later. Many of its accounts have in fact moved to DB2 sooner than they planned, since Walker made it so easy, evp Craig Carlson says. Likewise, if they're going to consolidate systems, they're probably well aware of the pain of doing so, and Walker's pitch sounds especially attractive: You're going to have to go through this now, but by selecting Walker you'll make it easier the next time it happens.

The next chapter....

In addition, by building its systems with the aid of its Strategic Software tools, Walker is now able to sell those same tools to customers as customization aids. Walker is just the first of many companies whose strategies will complement IBM's Systems Application Architecture. No longer intent on eking out the last millisecond of performance by code tuning, they are instead concentrating on building tools to build software that will be malleable enough to adjust in years ahead.

However, Walker has neglected to garner all of those benefits for itself. The Software Bridge generates a fair amount of machine-specific code rather than standard-language source code that can be compiled automatically on the target machine. Over the next few years, says Carlson, the company will be rewriting the Software Bridge itself to be less environment-dependent. With its layered architecture that keeps interface-, OS- and database-specific modules separate from the application-generation logic, he adds, the Software Bridge should also easily support client-server implementations, with applications on the server and interface modules on the client.

CULLINET: BACK TO THE FUTURE

Cullinet has adopted a more radical approach than Walker. You could call it all the buzzwords rolled into one, or you could call it a visionary synthesis of the best the next decade has to offer. (See also page 19.)

Cullinet flew high as a database company by selling a product better than IBM's IMS, but it failed to respond to the advent of DB2, a better product yet. Along the way, Cullinet acquired Distribution Management Systems (Release 1.0, 86-8) and its chairman John Landry, who is now setting Cullinet's technical agenda. Landry displays unusual flexibility for someone so passionate about technology -- from a belief in 4GLs (he was behind McCormack & Dodge's development of Millennium), he has moved on to espouse expert systems (the basis of DMS's rejuvenation), client-server architectures and now CASE, which will underlie whatever progress Cullinet makes henceforth in the applications market.

Over the years, much of Cullinet's success came not from its IDMS database per se, but from the suite of tools the company developed or acquired to work around it. Those tools allow customers to build their own database-based applications; they include ADS, an IDMS-specific 4GL. Beginning in the early Eighties, Cullinet also ventured into the applications business, with packages built around IDMS and modifiable by the Cullinet tool set. At this point, 50 percent of Cullinet's new-product revenues (aside from service and add-ons) come from applications for functions including manufacturing, accounting, banking, distribution and human resources.

Release 1.0 18 July 1988
With the acquisition of DMS and Landry, Cullinet has begun (slowly) to enrich its applications with expert system modules for installation support and, in distribution, to manage stock-picking sequences. The company has also moved beyond the ADS 4GL to KnowledgeBuild, a CASE tool that generates third-generation languages such as COBOL, BASIC, FORTRAN and (in October) C, and embedded SQL.

So far as we know, KnowledgeBuild will also initially be unique aside from Aion in its ability to generate applications for cooperative architectures at a time when most CASE vendors are just starting to take note of client-server architectures in their development platforms. In a release due within 12 months, KnowledgeBuild will define certain processes as either server or client functions, and build the proper environment-specific code (you could call it object-oriented messages) to manage the communications. Presentation always occurs on the pc or terminal; application logic on the host. KnowledgeBuild currently runs on and builds systems for VAXes and pcs; an IBM MVS version will be out by year-end.

STRATEGIC CASES: CADRE TECHNOLOGIES EXTENDS THE TECHNOLOGY

Cadre Technologies, a longtime vendor to the scientific/engineering community, is now carefully entering the commercial market through a tie-in with Pansophic. Cadre too is using modern development techniques to ensure the enhanceability and extensibility of its products.

Cadre Technologies was co-founded by former Yourdon employee Lou Mazzucchelli to build front-end CASE tools, on Apollo and later Sun systems, for the engineering market. After five years in business, the company dominates the high-end upper-CASE (analysis and design, as opposed to coding) market with 3000 installations at $5000 to $15000 a seat. (Index Technology, selling a PC-based product to a more commercial marketplace, is the unit leader with an installed base of 10,000 at $8400 list). Index recently went public; Cadre, which is profitable, has funding from Advanced Research & Development, Fairfield, Burr Eagan Deleage and T. Rowe Price among others, and also picked up an undisclosed sum from Pansophic Systems last year.

To Cadre, Pansophic represents not just money, but also a channel into the commercial marketplace it has neither the inclination nor the resources to exploit. The two have a marketing/development deal that should prove fruitful for both sides, especially with the new OS/2 version of its Teamwork line (single-user this fall for $4995, with a server version early next year). Several vendors including Nastec, Index and KnowledgeWare currently offer bridges to Telon, Pansophic's code generator; Cadre will also offer such a bridge by year-end. But the real promise lies in Teamwork's future ability to create program elements to load into Pan/RD, a depository that will serve as the foundation of Pansophic's development/execution environment. (See Release 1.0, 88-4.) That will enable customers to design (and ultimately maintain or enhance) systems using the Teamwork workstation, and then generate or regenerate the actual applications with Telon.

Aside from this excursion into the commercial market, which seems to promise a maximum of payback from a minimum of investment, Cadre is focusing on the functionality of the Teamwork, which has grown from a diagramming tool to an extensive suite of tools that does almost everything -- real-time system de-
sign, documentation management and even a little code generation and simulation -- except for COBOL generation (Telon's forte). Cadre is now rewriting the application logic of Teamwork in C++ (from C) and is considering doing the same for the data management portion. (The user interface was already developed to be hardware-independent and tightly coded, so that it couldn't benefit much from rewriting at this point.) The first result, a system tailored for Ada developers with Ada structure graphs and semantics, will ship in September. The next release of the whole system next winter will be based on the new object-oriented implementation.

So far Mazzucchelli is delighted with the effects of using C++. The system takes up less space since many of its functions are inherited rather than implemented several times over in slightly different form in different parts of the program. In virtual-memory environments, that also ends up improving performance because there's less need for swapping code in and out of memory. The object-oriented approach also gives the product line hardware independence, flexibility and potential for reuse of code. Says Mazzucchelli, "It's easier to maintain and enhance and think about."

Objective CASE

The new implementation was designed with Teamwork. How does a CASE tool mix with object-oriented paradigms? It requires some adjustment of thinking, but it works fine. Traditional notions of structured analysis and design -- data flows, state machines and entity-relationship diagrams (ERDs) -- all apply to object-oriented systems. You can use state machine concepts to model the behavior of objects, ERDs to model the object or class hierarchies, and data-flow diagrams to model the interactions of objects, with the difference that messages rather than data are passed from node to node. It's simply the implementation that ends up differently -- and you can also use traditional programming languages to carry out the coding, except that objects call subroutines instead of sending messages.

Long run, Cadre hopes to build tools not just for CASE but also for applications, so that customers can reuse and tailor them for their own applications when (if) they too adopt object-oriented implementations. Customers who have expressed interest include those with applications in database design, computer-integrated manufacturing and telecommunications. Potential object classes might thus include transactions or tables, data switches, machines, work cells and other "real-world" objects.

When in Rome...be yourself

For better or worse, Cadre is taking an aggressive approach to its market. Where arch-competitor Interactive Development Environments adapts to its local hardware -- following the relevant interface conventions, for example, and relying on standard tools such as UNIX's source code control system (SCCS) -- Cadre has constructed an entire environment that looks identical across platforms. That makes it easy for customers to switch from one platform to another, but a little tougher for them to get used to Cadre's Teamwork in the first place.

---------

3...which may only be a matter of how you think about them...
HORIZONTAL COORDINATION

Last month we wrote at length about groupware and the benefits of computer-supported coordination in workgroups. But especially if you're talking about horizontal (not hierarchical) coordination, there's no reason it can't happen across corporate boundaries, mediated by the market or by one of those ubiquitous "strategic alliances." In fact, coordination technology, in conjunction with the emergence of standards, is going to shift the competitive landscape dramatically by changing the relative advantages of vertical (hierarchical) coordination vs. horizontal or market coordination.

In an increasingly complex and mass-market world, there's increasing need for antidotes to complexity. One is standards, which reduces the need for coordination. Standards make for fungible products except where differentiation confers distinct competitive advantage, and allow people to buy products without concern for details, since theoretically those details are the same across the board.

The other antidote is technology, which improves our ability to coordinate. Better coordination technology will make it possible for smaller companies to work together and compete successfully with bigger ones. Many small companies will be not systems but business integrators -- performing roughly the same function as upper management in a more traditional situation. But instead of owning the coordinating parties, they will act as intermediaries among them. There will be benefits to doing it this way: Each unit presumably will do a better job by itself than if directed by a rigid owner, there will be no corporate overhead (although there will be local overhead), and competition will keep the individual units lean and mean.

There are two basic ways of coordinating large-scale productive/economic activity -- hierarchies (management or a single powerful customer tells you what to do) and markets (you negotiate with a number of peers). As pointed out by professor Tom "Information Lens" Malone of MIT's Sloan School, information technology will have a huge impact on the relative merits of these two approaches to coordination, so much so that he's forming a center at MIT for the study of coordination technology. Hierarchies are a natural, effective approach when the costs of communication are high; one central authority knows what's going on and controls everything, while the players may have little or no information. There is inherently less local flexibility this way, because planning is traded off for real-time coordination. (On the other hand, a centrally controlled structure may respond faster once someone at the top makes a decision, because information flows down instead of trickling across.)

But now, with better communications, both in speed and bandwidth, it's possible to have effective horizontal coordination, not just within companies but also across them: If you know exactly what you want to buy and who has it (standards and good information) and you can keep track of your vendors with trustworthy records for royalties and other cross-payments, you can safely buy from outside.

---

4For more on this issue and its implications, read some of Malone's excellent papers; see resources, page 20.
Moreover, you can farm out functions such as distribution or order consolidation. Electronic data interchange (EDI) systems, such as GM’s links to its dealers and suppliers or Sterling Software’s Ordernet unit which transmits orders between health & beauty companies and drugstores, will become commonplace. Credit-card systems and travel agents are just the beginning of a proliferation of third parties who are getting into the act between customers and suppliers without any physical involvement.

Coordination technology

In short, coordination technology consists not of technology per se, but of its application to tasks such as transaction management (see Release 1.0, 88-6, for discussion of work efforts or tasks as transactions), inventory management, record-keeping, pricing, terms & conditions negotiations, and simple communications so that people know what’s going on. All these are normal activities, but they cost so much that frequently it was cheaper to avoid them by integrating vertically. But now many large companies have cost centers, transfer prices and other coordination mechanisms that either are redundant, or implicitly argue for increased decentralization or even divestiture. And small firms can afford to stay small and cooperate instead of rashly consolidating.

We’re not talking about a group of similar companies getting together in restraint of trade to take on a big competitor, but rather a group of companies parceling out different roles among themselves, working together to form a single virtual company. Take, for example, the recent three-way deal among Sybase, Microsoft and Ashton-Tate. Sybase built it; Microsoft owns the OS/2 OEM rights; and Ashton-Tate distributes it at retail. Very little physically changes hands; what’s going on is the transfer of intellectual property, with appropriate records kept. (In the old days coordination of such a deal would take place by the transfer of physical assets: Juan grew the corn, Alice milled it, Fred packaged it, and Greg’s Grocery sold it. But now Juan is AgriJuan, a farming conglomerate; Alice is Global Milling, an Alco unit; Fred is Federal Fred, a wholesale food supplier; and Greg got bought out by Safeway years ago.) With luck, and with coordination technology, we can keep the world safe for smaller companies.

Overnight slides

To pick a small-scale, real-life example to make the point, take MAGICorp. MAGICorp is a small company that you might describe as a service integrator. It runs a shop of slide-making equipment -- and relies on other companies for the rest of its operations.

Customers who use graphics software frequently want to make slides, a cumbersome, expensive, time-consuming process -- unless they call MAGICorp. Using toll-free lines (billed to MAGICorp by AT&T), they can transmit their computer data to MAGICorp’s office in Wilmington, OH. Why Wilmington? Because of the Airborne Express hub next door. Once the slides are made -- within hours -- MAGICorp ships them back overnight to the customer via Airborne Express. Magic! Juan can wait till the day before the presentation -- not because he’s lazy, of course, but in order to incorporate last-minute data from Alice -- to prepare his slides.

Release 1.0
18 July 1988
MAGICorp markets through companies such as Business & Professional Software, Ashton-Tate, Digital Research and Symantec. For example, it supports the new version of MORE announced last week, via an arrangement for graphics drivers with BPS, playing yet another role in this horizontal net. Cricket and Symantec stuff their packages with inserts with instructions and a trial offer of MAGICorp's service, and get royalties (based on revenues minus rush charges and Airborne fees) in return. (MAGICorp also handles other vendors' software, such as Software Publishing's Harvard Graphics, but they don't promote the service and MAGICorp doesn't pay them royalties.)

MAGICorp knows exactly how much of each vendors' product it handles, since it needs the information to create the slides from object files generated by each brand of software. (Typically, it works with the software vendor to create the appropriate drivers.) It uses eight $250,000 cameras from Celco (the Cray Research of the image business, for whom MAGICorp is the largest customer outside the U.S. government). Each order is carefully recorded for billing and later royalty accounting. All this information, of course, makes it easy for MAGICorp to work with the outsiders whose services it integrates into a virtual company. MAGICorp performs just one basic service -- the slide-making -- and coordinates the others. Its fees can reach twice what it would cost to get the slide done at your neighborhood slide shop. But MAGICorp lets you push a button at your desk, and get the slides back by noon the next day. That's the value of coordination, and there's sizeable profit in it for MAGICorp, as well as for Airborne and for the software vendors.

United we coordinate -- and independent we incorporate

Vertical integration makes sense in a world where it's the only way to ensure a steady, reliable source of the exact quality and quantity of products that you need; in a communication-, coordination-rich world, it's no longer necessary -- and may in fact be a disadvantage. There's a lot of redundant information generated, managed and wasted when someone at the top needs to manage things; there's an extra layer (or several) of fundamentally unproductive people whose only job is traffic control, not locomotion; and there's a tortuous communication path up and down the hierarchy that simply slows things down.

But then who makes the important decisions? The same people who always made them. The guy who determines what the market wants still does that; the guy who designs the product to suit the market still does that; the guy who figures out the best way to build the widget still does that. The only difference is there's no one at the top trying to make all those decisions. (Look at the Soviet Union as it was and still, painfully, is, to get a sense of what that's like!) Instead, there's someone on the other side of the (electronic) table providing information and negotiating for that decision. The reason the Soviet Union is having such trouble now is because it has no markets to transmit information, let alone computerized markets to transmit that information quickly and effectively.

In sum, the innovations in computer technology that small firms are creating will make the world safe for small firms. The benefits include faster communications, less overhead and waste, and ultimately a better match between supply and demand.
ADAPSO LEADS THE WAY

We write this with a certain vested interest, since we sit on the board of Adapso's Micro Software unit and helped bring about the events described.

Acknowledging a technological manifest destiny before it has occurred, the micro and "mainframe" software sections of Adapso have agreed to merge. Adapso began in the Sixties as a dp-services-oriented trade association, and merged in the mid-Seventies with the Software [products] Industry Association. (Its other sections include processing services, VARs, and systems integrators.) In the Eighties, micro software companies joined Adapso in force. Dan Fylstra of VisiCorp, David Cole of Ashton-Tate and Mitchell Kapor of Lotus participated actively.

Since then lots has changed, and the micro companies are less visible -- but more integrated. The software groups are calling on each other's customers, buying each other and otherwise mingling. Ed Esber sits on Pansophic's board. Members of the micro board such as Rich Carpenter of Index Technology and president Paula Brooks of UniTech Software sell their software through traditional mainframe (direct) channels; it just happens to run on pcs. In fact, the machine software runs on no longer defines its character.

Meanwhile, the Software Publishers Association has attracted a large following of its own, including a number of Adapso members, and is becoming increasingly business-oriented (although a large part of its membership still sells games and educational software). Thus, it makes sense for Adapso to distinguish itself by offering its micro members a chance to join other companies who worry about MIS issues, while allowing the SPA to focus more on retail-distribution issues. (Of course, both organizations would like to think they can offer everything to their members...)

There were other reasons for the merger too. Both boards needed rejuvenation, and the combined entity should have more clout within Adapso. Perhaps the precipitating factor for the merger was a proliferation of committees devoted to the subject of intellectual property. (What is it? How strictly should it be protected? What is openness, and how much is good?) While it's futile to hope that one committee will agree on most of these issues, it's far better to have the members disagree across a single table -- and at least coming up with some common definitions, if not some common advice for the courts -- than to hash over the same ground in separate, smaller groups. (For our part, we couldn't spend the time going to all these meetings!)

Approval was not unanimous: Some companies think they'll get more notice as a large part of a smaller section than as a smaller part of a larger section. On the other hand, companies such as Microsoft, Ashton-Tate and Lotus hardly need to worry about being heard....

We believe fiercely that both Adapso and SPA are useful, productive organizations, and that companies can only benefit from getting involved, trying to define intellectual property and the laws that will affect it and mingling with their peers (competitors, potential partners, employees or employers all). Please call either group for further information (see Resources, page 20).

Release 1.0
18 July 1988
ORACLE: DATABASE COUP

Buzzwords have a way of obscuring meaning rather than illuminating it, and so it is with SQL. There's a lot more meaning in the phrase "SQL support" than is revealed in most check-off lists. Consider Oracle's recent clever deal with WordTech. Without much ado, it enables dBASE users to move to Oracle without having to throw out all their old, painstakingly developed dBASE applications. This fits the natural order of things, since it's much easier to move one's data to a new database than to rewrite applications, and Oracle's strategy supports people in following this natural inclination. And it's perfectly legal, given WordTech's deal with Ashton-Tate.

In one fell swoop, Oracle has now made itself an indirect beneficiary of the large third-party community that supports dBASE. Any dBASE source-code program, whether a third-party accounting package or an application built with tools such as WallSoft's UI, can be compiled by the Oracle "dBASE liberator" products (Oracle dEXL and Oracle Quicksilver) into an application that runs off the Oracle server. While WordTech will probably produce versions of Quicksilver and dEXL that support IBM's OS/2 Extended Edition and the Sybase SQL Server, we understand it is unlikely to do so for Oracle's arch-competitors Ingres and Informix.

Meanwhile, we hear Borland is about to do the same thing for Paradox, so that dBASE users can run their old applications against new Paradox database servers. That should be beneficial for Paradox, but of less moment to Ashton-Tate. Moreover, Paradox's greatest strength is its user interface, so this kind of move is less strategically compelling for it than for Oracle.

LIFEBOAT, RELEASE 4 (MAGELLAN)

Remember Lifeboat Associates? After a couple of management changeovers, it was acquired in 1986 ago by Scarborough Software, which it joined under the umbrella of Hudson Technologies. That effort didn't really take, and now the company has merged again, with Corsoft of New Jersey, to form Magellan Software. This time things look a little more promising. The two companies serve the same market -- languages and tools for programmers and technically savvy users -- whereas Scarborough sold Master Type. Moreover, the nerd market has blossomed substantially in the last few years. Borland, Microsoft and Think Technology are flourishing in the language business, but there are hundreds of other smaller companies whose products are selling well -- just not at Egghead volumes. Programmers have a tough time getting hold of them, and rely mostly on mail-order companies such as Lifeboat's own Programmer's Paradise telemarketing unit or specialty houses such as Corsoft, which sells direct to engineering-oriented companies such as Bellcore, AT&T and Bell Labs (worth $2 million in annual business each to Corsoft).

Lifeboat also brings a dowry -- rights to the DOS and (coming this fall) OS/2 versions of one of the hottest products around, G++, through a deal with Dublin-based Glockenspiel. Courtesy of the VCs who brought the units together (J.H. Whitney, Alex. Brown, Oak and NEA), the new company also has new management in president Roger Paradis, a veteran high-tech entrepreneur who once worked for AT&T. Now Magellan may finally come around the world to where it started -- a premier software house for technically oriented folks, with revenues estimated at $30 million over the next 12 months.

Release 1.0

18 July 1988
RELEASE 1.1: THE OPEN SOFTWARE FOUNDATION

The Open Software Foundation is just about to issue its first RFT -- Request For Technology. It is soliciting offerings for a user-interface standard -- one of the most hotly debated issues around and a good one to test the OSF's mettle. The decision date, when OSF plans to announce its selection, is December 15. We put it at Thanksgiving in our last issue, but the Foundation says the date hasn't slipped; it was simply mistaken in giving us that date. Make of this what you will.

One interesting question is what Microsoft will do about this. As we understand it, the company is free to license Presentation Manager on whatever terms it pleases (although we've got to believe it will listen to IBM's point of view with intense interest). Of course, Microsoft tried to dissuade IBM from joining the group in the first place. And of course, no one has to make any decisions just yet. The RFT asks only that the licensing terms be "reasonable [and] stable," and that's clearly a negotiable point that could stall action at any time. We suspect that Microsoft will send something in. What happens next is anyone's guess.

Meanwhile, Steve Jobs at NeXT, who has reportedly licensed his UI to IBM, may also offer it to the OSF.

RELEASE 1.1: CHAPMAN SURFACES AT COMPUTER POWER

Computer Power is Australia's largest dp services company, akin to EDS and Computer Sciences rolled into one. The company has had U.S. operations for some years, mostly devoted to selling its Status text database, initially out of offices shared with the New York Post. (Post owner Rupert Murdoch is a substantial investor in the company.) But recently CP has reorganized at home and has started buying up professional services organizations in the U.S. with a view to replicating its Australian business here. The U.S. operation, which now includes Applied Information Development, Cornell Computer and CP Information Systems, will be run by Dave Chapman, who left the chairmanship of Cullinet last spring shortly after John Cullinane returned. Estimated revenues for the U.S. operation this year are $70 million.

RELEASE 1.1: ERRATA

In our last issue, we mistakenly referred to Cullinet's new VAX dbms as IDMS/R. In fact, that's the name of the relational version of IDMS that runs on IBM mainframes. The VAX product, an entirely new product built on a foundation purchased from Esvel Technology, is called IDMS/SQL.

We also made mistakes with the names of Mike Szabados of Apple and Bill Crow of Hewlett-Packard. Sorry!

Release 1.0 18 July 1988
COMING SOON...

• Active objects. (It is expected that passive objects will also be discussed.)
• Database styles: Oracle, Sybase, Interbase, Cullinet, IBM et al.
• Graphics standards.
• Network strategies.
• Implementation support.
• Truth maintenance.
• And much more...

George DeBakey, Ron Palenski, Adapso, (703) 522-5055
John Schneider, Airborne, (206) 281-1044
Jim Baker, Anselm Software, (415) 893-2973
Jim Baker, Kinetic Computing Inc., (415) 454-9707
Glover Ferguson, Mel Bergstein, Arthur Andersen, (312) 507-6478
Lou Massucchelli, Cullinet Technology, (401) 351-5950
Dave Chapman, John Moriarty, Computer Power, (508) 872-7551
John Landry, Cullinet, (617) 329-7700
Tony Kobine, FCME, (415) 549-7100
Roger Paradis, Magellan Software, (201) 389-8950, (212) 496-8200
Karl Lautman, MAGICorp, (914) 592-1244
Tom Malone, MIT/Sloan School, (617) 253-6843
Alex Morrow, Open Software Foundation, (508) 683-6803
Pete Tierney, Oracle Corporation, (415) 598-8157
Ken Wasch, Software Publishers Association, (202) 452-1600
Juan deNaranja, Alice Haynes, Universal Database, (212) 758-3434
Craig Carlson, Dave Brownlee, Walker Interactive, (415) 495-8811
Mike Gardner, WordTech, (415) 254-0900

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

Release 1.0 18 July 1988
RELEASE 1.0 CALENDAR

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

July 28-29
EDI Intertrends - San Francisco. Find out what's going on with electronic data interchange. Speakers from DEC & IBM; real-life stories from Chevron, AMD, others. Contact: Jan McDaniel at Input, (415) 961-3300.

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

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

August 5-7
LANDEX '88 - Chicago. Sponsored by LANDA, the Local Area Network Dealers Association. Contact: Pauline Nishi at (312) 279-2255.

August 10-13

August 15-19

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

August 22-26

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

August 28-September 1

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


September 7-10  Optical information systems - New York City. Sponsored by Meckler Corp. Contact: Hannah Preston or Marilyn Reed, (800) 635-5537 or (203) 226-6967.

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


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

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

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

Release 1.0  18 July 1988
Release 1.0

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

September 26-30


September 27


October 3-5

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

October 3-6

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

September 26-28


October 10-14

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

Release 1.0

18 July 1988
### October 11-14
**Info Show - New York City.** Contact: Frank Fazio, Cahners Exposition Group, (203) 964-0000.

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

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

### October 19-20
**Vardex - Dallas.** The value-added reseller & dealer exposition. Co-sponsored by Adapso and NAVAR, the National Association of VARS. Vertical markets and tailored software galore. Contact: Dan Dembicki, (313) 274-8590.

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

### October 17-21
**Electronic document management: Approaching the 21st century - San Jose.** Sponsored by CAP International. Contact: Kristin Fisher or Jean O'Toole, (617) 982-9500.

### October 23-27
**American Society for Information Science - Atlanta.** The 51st annual. Contact: Bob Hurd, (202) 462-1000. With speeches by Charles Marshall, AT&T; Max Hopper, American Airlines; Anthony Craig, GEISCO.

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

### October 24-25

### October 24-27
**Conference on software maintenance - Phoenix.** Sponsored by IEEE and other organizations. Talks by Robert Arnold, Software Productivity Consortium, among others. Contact: David Card, (301) 589-1545.

### October 24-28
**Federal computer conference and Defense & government computer graphics conference - Washington, DC.** Sponsored by National Council for Education on Information Strategies. Contact: Luanne Bell, (800) 343-6944 or (301) 961-6575.

### October 25-27
**Teamwork, technology, & organizational performance - Cambridge, MA.** The sixth annual Seybold Executive Forum. Contact: Deborah Hay, (617) 742-5200.

---

**Release 1.0** 18 July 1988

October 26-28  Venture forum '88 - Boston. Sponsored by Venture Economics, the authority in the field. More for the benefactors than the beneficiaries, with focus on trends and mechanics of the business. Talks by many VCs and by perennial supplicant Bill Poduska (Prime, Apollo, Stellar, ...?). Contact: Diane MacArthur, (617) 237-1121.

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


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

October 31-November 2  UNIX Expo - New York City. Managed by National Expositions. Contact Don Berey, (212) 391-9111.

November 2-4  Adam Green's dBASE R symposium - Woodland Hills, CA. See October 3. Contact: Marny Peabody, DCI, (617) 470-3870.


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

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

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


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

Release 1.0  18 July 1988
December 5-8  CASEXpo - Anaheim. Sponsored by Arthur Young; chaired by Howard Yudkin, president of the Software Productivity Consortium. Contact: Rhoda Canter, (202) 956-6041.

1989


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

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

March 14-16  Interface and World Congress on Computing - New York City. Moved from Chicago, in search of more enthusiasm. Sponsored by the Interface Group. Contact: Walt Heithaus, (617) 449-6600 or 800) 325-3330.


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

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

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

Release 1.0

18 July 1988
SUBSCRIPTION FORM

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

Name
Title
Company
Address
City State Zip
Telephone

How did you hear about Release 1.0?

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

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

Lisa Braden
Associate Publisher

Release 1.0
18 July 1988