CASE-BASED REASONING: A FAMILIAR STORY

Remember rule-based expert systems? A perfectly nice technology, overhyped to the point that dozens of companies staked their lives on it -- and lost. Now expert systems are widely used all over the place, but more as a programming technique or modules integrated into broader systems than as standalone black boxes based on a single vendor's product.

Now get set for case-based reasoning. No programming required! So simple tech-support people can use it! Totally intuitive! And, of course, object-oriented!!! CBR was the hottest thing around at last summer's AAAI conference, the AI community's biggest annual event. And three vendors -- Esteem Software, Inference Corporation and Cognitive Systems -- are now offering CBR systems (see page 10). All the hype notwithstanding, CBR is a worthwhile field, and is likely to be as widespread (and frequently as invisible) as AI within a few years.

Case-based reasoning is a fairly simple idea: Instead of solving a problem with rules, solve it by comparing it to similar problems -- much as a real expert often does. (Case in this sense stands for a case as an example, and is not an acronym.) Case-based reasoning is an approach to certain knowledge-acquisition and knowledge-representation problems for a certain group of applications, rather than a discrete technology (and it does not yet handle well the problems of knowledge generalization or adaptation). The primary underlying technologies are pattern-matching, induction (as when an expert system derives rules from examples, or cases) and statistical techniques to measure closeness such as those used in text retrieval and classification. But of course the key to AI in general is simply the art of using 1s and 0s to represent and manipulate knowledge in a way that looks smart and flexible. CBR's ability to represent case knowledge in a way meaningful to and sharable among casual users is a genuine benefit.

Reusable knowledge

A primary feature of CBR is that you can retrieve not just cases, but related information. This information could be instructions, diagrams, code, multimedia sequences, or even active hypertext links to online documentation or an automatic phone dialer to call a live support person at some vendor's headquarters. Depending on the application, the desired result of a CBR session may be a single case --what's wrong and what to do -- or it may be a set of possibilities -- where to go on vacation, possible job candidates, potential factory sites, etc.

HAPPY NEW YEAR, ESPECIALLY TO NEW PARENTS OF 1991
While much of the current fuss about CBR focuses on help desks -- a practical, mainstream application everyone can appreciate -- we can imagine a wide variety of uses. Here are some examples:

- matching job requisitions to job candidates. For example: "We got any scripts with a spunky heroine, a handsome leading man and a happy ending, in the tropics?" for a director who's signed to do a movie with Annette Bening and Warren Beatty, and wants a winter vacation.

- selecting form letters in response to incoming letters. For example, candidate acceptance letters or rejections with personalized career counseling; thank-you letters targeted to particular gifts; performance appraisals with improvement suggestions; investment advice. The form letter would be selected by the CBR, and data would be incorporated with mail-merge to produce personalized form letters.

- Christmas present recycling, matching gifts and possible recipients.

- tailoring product extensions to match customer-segment needs.

- supplying jokes to match a speaker's points. For example: "You have to watch out what you say even when the boss is not around" is the input. CBR supplies a joke to make this point.

- selecting products to match a spec, such as software, factory sites, vacation spots, or even movies.

- finding legal precedents.

- identifying code modules for reuse.

- predicting cancer recurrence or other diseases.

- making medical diagnoses.

- personal ads. Juan desperately seeking Alice, or someone just like her.

- some day, an interactive "classified" ad bulletin board or on-line purchasing system based on a CBR system -- called "Twenty Questions," of course. The system can work bi-directionally: One man's case is another man's answer.

Moreover, CBR will become just another retrieval technique in many mainstream applications, just as expert system modules are routinely used in many applications today. Companies using or installing CBR systems include American Airlines, American Express, Boeing, and many banks and the ubiquitous "unnamed government agency." The most marketable appeal of CBR is that it lets you define cases in almost-natural language, answers to questions or forms, so that the system can be used and even maintained by domain experts instead of AI experts. Instead of painstakingly generating a set of rules, the builder-user can simply define a case base (a set of representative cases), and let end-users match their own new cases against it. The system "learns" as it is used, whenever users add new cases.
Of course, there's a little more to it than that. How do you define similar problems consistently, and how do you determine what's similar? What are the rules for generalizing or adapting cases? Nonetheless, for most domains, it's easier to build case bases (or better yet, to find them in existing databases) than to construct rules. Ease of maintenance tends to favor CBR over purely rule-based systems even more strongly. Of course, the real issue is simply matching the problem to the technique (a fine application for CBR). For example, laws or tax regulations are best represented as rules, but determining when a law applies -- was the victim truly afraid for her life? -- might be a job for a CBR system using descriptions of previous cases.

--

Chances are you'd rather have an old doctor than a young one, says CBR vendor Cognitive Systems' Ralph Barletta. Despite the younger person's possibly superior (or at least recent) formal education, the older doctor has more experience -- i.e., has seen more cases. Those cases give him a better knack for the job than the limited number of cases and the rules found in books -- and rule-based expert systems.

-----------

Representative cases

In the broadest sense, CBR lets a builder define situations and appropriate responses, so a user can apply this knowledge to new situations. (When a truly new situation is encountered, a new response is determined by a domain expert and added into the system. Of course, a CBR "solution" depends on the application. What are you trying to find out? Basically, there are three kinds of problems:

1. You are trying to find a single best-description -- as in diagnosis, design or planning -- although you want to present a number of options for an expert user to consider. The "answer" is not so much the cases themselves, as associated information: a diagnosis and suggested treatments, instructions to read to a caller over a phone, or ways other designers built a factory or car under similar constraints. (There could be a single answer, such as "Investigate further," if you're an IRS agent trying to flag patterns of suspicious bank activities for further investigation, but a look at the cases and the outcomes may give you an idea of how to investigate and what to look for.)

2. A special case of the first situation is when you are looking for a numerical answer -- the probability that someone will default, the appropriate price for a house based on appraisals of similar houses. You can assess a number of similar cases, and (in the better systems) you can also see what factors have the greatest bearing on the outcome. You can also "adjust" for those factors -- adding $20,000 for the extra bathroom, for example, or raising the rating for a particular job title even if there's no case with that title.

-----------

1. Another breakdown is select, adapt or compose: Find one, find one and change it, or find several and combine them.
You want to select an item (case) from the database to meet a user's specifications -- for anything from a vacation spot to a factory site or a dream house, a new hire or a new job.

One advantage of most end-user CBR applications over purely rule-based expert systems is that they leave more control in the hands of the user. The user decides which questions he wants to answer, and in what order (a "mixed-initiative user interface"). The user likewise can see the range of possibilities, rather than wait for the system to present a single answer. Of course hybrid expert systems, with more flexible construction and some CBR features, also have these advantages.

CBR cases are also easier to add to a case base or modify than rules, since there's no requirement to edit a rule set, reconstruct a decision tree or perform other reconfiguration; the system easily absorbs new experiences (it does automatic reindexing but also takes as much tweaking as expert builders care to do). CBR is not as elegant or precise as a rule-based system, perhaps, but it reflects the messiness and quirky cases of reality.

**Setting up the case base**

There are a number of functions in CBR: defining the cases, categorizing them, matching new cases to the case base, and refining them to generate new cases. The value of a CBR tool is to make building such a system as easy as possible, with matching techniques as powerful and accurate as possible.

The first step is to create the case base. The cases provide a context for the interpretation of new cases: Previous cases are assumed to provide useful information about similar new cases.

A case itself is a collection of tokens or data elements defining the case's significant features: the answers to a set of questions, values entered into a form, or data downloaded from a database. These tokens or data elements can be numbers, Boolean values, text, symbols, graphics (such as the kind of streaks a particular printer problem produces), or even other cases. There can also be executable links in cases; Esteem uses them in its case-matches (as a database might use a lookup table) for nested cases, so that similarity can be assessed across case base. Thus, you could search for new-widget cases with a feature such as "ship date," which would point to an inventory case base used for determining ship dates.

A case typically also includes non-indexing items, such as names, ID numbers, pictures and other information irrelevant to classification of cases although it may be important for users or for other tasks. For example, extra information might be a picture of a house, a repair diagram, or possibly a QuickTime sequence.

No rules are needed in CBR, but they can be used along with it to reason about features, cases or even entire case bases in a hybrid system. For example, Palo Alto, Mountain View and Los Altos might be recognized as "Silicon Valley," while Berkeley or Emeryville would be recognized as "East Bay." Such enhancements can be arbitrarily powerful, up to the point of parsing natural language. You could also include calculated fields or rule-generated answers, such as "pregnancy defaults to 'no' if a patient is male, and case defaults to 'fraud' if answer is yes." Such hybrid applications...
are frequently the most appropriate way to represent reality, which itself is hybrid (to say the least). Does the system have appropriate metaknowledge? Only if a clever programmer puts it in.

Ideally, cases can be loaded from an existing database. In practice, however, many databases are too large or unrepresentative, and CBR developers typically do a fair amount of work in tweaking an existing database into a useful case base. In insurance applications, for example, there may be millions of records of customers and claims; Cognitive's ReMind, for one, is designed to handle a mere 100,000 to 200,000 cases at most. (Needed here: A tool to refine a database of incidents automatically into a useful case base of prototypical cases. We don't know of any yet, but there are certainly data inspection tools and rule-based systems that can help.)

On the other hand, when you're dealing with help-desk applications, it may be simpler to build the cases from the ground up, says American Airlines managing director of Knowledge Systems Lynden Tennison, whose development group did just that after wrestling with a messy base of trouble call reports that used inconsistent terminology and just didn't sort themselves out nicely. Overall, it's important to use explicit, consistent methodologies for CBR just as for any other kind of development. (See page 13.)

---

The presence of an existing electronic representation of cases is an important factor in building a CBR system. Some of the simplest cases are never written down (or typed in), because "everyone knows the answer." So some seemingly promising applications may not be viable just because of historical accident. Now is the time to get started keeping records! As business goes more and more on-line, more and more potential case bases will be created inadvertently.

---

Indexing or categorizing the cases

CBR experiences the usual trade-off between structure and speed on the one hand, and brute force and flexibility, on the other. Once you have the case base represented more or less as database records, you can index it in a number of ways: for hierarchical retrieval, template retrieval (database-style queries), or nearest-neighbor matching (similar to text search).

Generally, the developer decides which fields may be relevant for indexing. For example, ID number is used to identify candidates, but not to predict their career success. If it does predict career success, something funny is happening. Does the personnel manager know anything? Of course, finding such unexpected correlations is one place where induction can be valuable.

Decision tree indexing (hierarchical retrieval)

For hierarchical retrieval, the system derives a decision tree (or hierarchical index) from the data, determining which fields best discriminate among the cases by using ID3 (a well-known induction algorithm frequently used to generate expert system rules) or similar techniques. Each case is indexed by its location at a leaf of the decision tree -- or in several places, if there were unanswered questions. (When a user fails to answer a
particular question, a hierarchical CBR system typically follows all the branches from that point.\(^2\)

Similar cases are retrieved when an input case traverses down the decision tree, and ideally retrieves a small group of matching cases at the end. This is an extremely efficient search method, since only, say, 20 steps down a binary decision tree would discriminate among \(2^{20}\) (a million) cases or possibilities.

The difference from a rule-based system is that a CBR retains the cases, for examination by end-users or perhaps reindexing of the system as significant new cases or desired outcomes are added. Then the CBR system can use the decision tree as a search tool to retrieve cases that match the input case in all the differentiating aspects. These cases, however, may also include much rich unindexed information that would be lost if you kept the rules but discarded the cases as a rule-based system does. This extra information typically includes not just "the answer" -- a diagnosis, repair instructions, identities of possible job candidates -- but ancillary information that may be of interest to a user even though the system doesn't interpret it -- photographs, detailed case histories.

A user can then examine all the cases retrieved, or use nearest-neighbor or other methods to refine the search further -- the hybrid approach taken by Cognitive Systems and Esteem. As needed, the user can use these cases in making his own judgment or apply adjustment rules (by hand, or using facilities included in a high-end system).

A system designer can force some factors to come first in the decision tree (such as treating male and female as two separate case bases, or considering apartments and houses separately). Question weights can also vary for the same question in different contexts: Air-conditioning matters more in Atlanta than in Maine. Likewise, you can use ranges, lists of allowed values or approximations with the template-matching approach.

Template matching (database queries)

Another technique is template or prototype matching, where the input case is used as a straight query against the database to find matches, using traditional database indexing techniques, synonyms and ranges and other enhancements as appropriate. Note that CBR is an approach, not a technology; you can in fact use technology as mundane as SQL queries to retrieve and match cases if that works, although more complex cases require more complex discrimination and recognition methods. (In fact, many people may have been doing CBR all along without knowing it, just as they were speaking prose.)

\(^2\) Note that there can be several indexes for a case base: Each index is derived with regard to a particular outcome (or field value). That is, the factors that determine whether a person is a good credit risk and those that determine whether he's a good job candidate may be quite different.
Nearest-neighbor

A final approach is "nearest-neighbor" matching, where there's little structure or pre-indexing, except possibly a text-search-style inverted-list index. Thus it can easily handle incorrect or incomplete information, and can easily absorb new cases and new features that wouldn't fit into an existing template query or would require reindexing of a decision-tree system. Precisely because it's unstructured, it can easily absorb undigested -- i.e., unstructured -- information from the real world. Basically, you may pay at this time for the work you avoided earlier; such searches tend to be slow if care is not taken to build synonym lists, indexes and etc.

To find matching cases, a nearest-neighbor system takes all the information it can get, and calculates the relative closeness of the new case to all the existing ones (or to a subset determined by pre-processing), producing a ranked list. The system uses common text retrieval techniques, using tokens for features instead of words. While a word may or may not be a good indicator of the subject of a text, a yes answer to a question such as, "Is the flange leaking?" is a pretty good indication of a particular symptom (if we knew what a flange is!). Like a text similarity-ranking, nearest-neighbor typically uses a virtual n-dimensional space, where each token (answer to a question or value-attribute pair) represents a dimension, and the score is the distance/angle of each case across this n-dimensional space from the target case. (See Release 1.0, 1-89, or the work of Gerard Salton.) Lower distances score higher, of course.

The parallels between CBR and text-search tools are strong. In text search, you're trying to find a piece of text that matches a query (or a pre-defined category); in case-based reasoning, you're trying to find a matching "case." Just change the terminology, and you've got the same problem. Instead of a set of word matches and statistical correlations, each case is described as a set of features or symbols and values (answers such as yes or no to specific questions, rankings, symptoms and other values) and statistical correlations, but otherwise the process is pretty much the same. You can use either plain statistical techniques as in nearest-neighbor, akin to the Gerard Salton approach to text categorization, or hierarchical matching, akin to the technology in Verity's Topic (see Release 1.0, 1-88, 3-90). It all goes to show that the value of technology isn't in the technology, but in its application.

Cases generally come in clusters: printer problems, monitor problems, software problems, and so forth; ranch houses, Colonials and shacks. Of course, a software problem may look like a printer problem, or it could be both; a love story may look like a travel adventure...or a business trip. One problem's nearest-neighbor structurelessness solves painlessly is multiple inheritance: A single case may have aspects of a number of cases, so a rigid single-inheritance approach won't work.

However, the nearest-neighbor approach is somewhat brute-force. It can work well if the cases are well described, the questions are sensible, and so on. But it can also be a mess if the user doesn't know what he is doing; the

Release 1.0 31 January 1992
system may be confused by synonyms if it's not well-prepared, and it may get
littered with proliferating redundant questions and cases that can slow per-
formance and clutter the screen or confuse cooperating applications.

In all these respects, nearest-neighbor is no worse than the hierarchical-
index or template approach; indeed, it degrades more gracefully. However,
as a technique sold on the basis of ease of use, it may mislead users into
avoiding tweaking that could be useful. In fact, with nearest-neighbor, a
smart user or a system maintainer can easily rank questions by their impor-
tance by hand or add synonym lists. He can even assign an absolute score (a
sort of rule-in-disguise) to a feature, which brings a case to the fore or
discards it altogether: A no answer to "Is the plug in?" narrows the range
of possibilities considerably -- at least until the plug is replugged and
other possible malfunctions emerge.

The problem with nearest-neighbor from a scientist's point of view is that
there's no model in there; the search space is featureless. There's no
hierarchy of cases or abstraction of similarity; cases are classified with
respect to each other statistically at runtime, rather than by some overall
scheme. Question weights apply across the system, regardless of context --
unless you tweak the system. But that is also its power: The case base
exists; it doesn't need to be constructed.

---------
In certain aspects, nearest-neighbor is analogous to a rela-
tional database, while hierarchical-index systems are akin to a
hierarchical database. Performance of the hierarchical data-
base is better -- if it's well-designed and tuned to the par-
ticular application. However, a relational database is more
flexible and is better equipped to handle arbitrary, unanticip-
pated queries effectively. In short, nearest-neighbor usually
works without premeditation. "Cases" occur naturally, whereas
decision trees need to be built and derived -- and may also be
artifacts of a skewed case base rather than "true" rules.
---------

User interface

The "user interface" to a technology, of course, depends on the particular
tool chosen. We will describe three such tools (the major ones commercially
available, unless you want to include Verity's Topic as implemented in a va-
rity of applications that could be called CBR) starting on page 10.

But there are some general issues. While the developer's interface varies
all over the place, the typical end-user's interface is likely to be a form
or a sequence of questions to answer. Do you want to describe the case in
its entirety, or are you playing 20 questions, asking a customer at the end
of a phone line a sequence of questions aimed at finding out the nature of
his problem? A form is better for the first approach, while a dynamically
restructuring set of questions (depending on previous answers) is better for
the second. Building the questions to elicit the proper answers can be more
work than designing a form, but either one requires much the same thought
process. The developer can also produce pick-lists or sets of options for
many questions, which makes the range of possibilities much easier for the
underlying system to manage and is almost necessary for template matching.
Does the system conduct its search dynamically, or only at certain points in the session? There’s a trade-off between making the user wait each time, and lowering the search space quickly. (With nearest-neighbor, you typically have to recalculate a fairly large subset of the search space, although you can limit it with certain heuristic constraints.) Can it cache indexes and most-likely cases?

A third issue is the interface for maintenance. Can a naive user easily add cases to the system, or adapt cases? How often does reindexing occur for an induction-oriented system? How easy is it to force an arbitrary structure on the tree? And how easy is it to see what you’ve done? Who defines, solves and adds new cases?

Note, of course, that the case base and the new cases may not always come from a user typing into a friendly interface: A CBR system could catch data from, say, a network monitor, or by scanning customer letters, parsing a news feed, or watching bank-transfer activity -- as in a system Cognitive is building for an unnamed government agency.

Adaptation and maintenance

Naturally, it’s easy to add a new case to a nearest-neighbor or template system; it just gets included as one more case in the case base, perhaps with a few new questions (as long as there are no new features not in the template). In a hierarchical-index system, a new case can be stored and linked to the appropriate location in the decision tree. If it matches the current decision-tree structure exactly, then it simply adds richness to the case base. However, if it’s a case that might alter the structure of the decision tree, or an adaptation of an existing case, then at some point the case base should be reindexed with the new cases in it. (Once you’ve tweaked a decision tree manually, you have to do it again each time the case base is reindexed.) How frequently such updating occurs is generally up to the developer -- especially when it involves the creation of new indexing features. For example, you might not have paid attention to a person’s field of study, but now you decide it’s relevant. You could go back and reindex the case base with field of study as one of the index features.

More interesting than just adding cases by hand is the generally unsolved problem of adaptation and generalization. How do you derive adaptation rules automatically? Of course you can apply traditional inferencing and statistical techniques to find correlations and dependencies. Tools such as ART-IM and Esteem, in particular, let you build and apply your own rules for adaptation. But the most interesting problem is finding the underlying structure or model of a case -- which usually requires domain-specific reasoning and is hard to do in a general way.\(^3\)

\(^3\) Inference cites one example of automated adaptation (as opposed to automated generation of adaptation rules) in a CBR system built in ART-IM for NASA to model space-shuttle missions. It uses CBR to find simulation-code modules for reuse, and then applies a set of rules to adapt and compose them into a simulation of a new mission. For example, if you combine parts of two different missions, you may need to transform the spatial coordinate systems to align them properly.
What’s lacking is a more interesting way of measuring relationships among cases than similarity: analogy. CBR, basically, identifies objects with similar components -- locations, sizes, job skills, constraints, problems. A more interesting system would discern relationships among components within cases and similarities in the structure of such relationships between cases, based on techniques such as Cognitive founder Roger Schank’s scripts. In the end, this means the natural-language approach -- assessing meaning and actions, not just topics. "Boy and girl" is fundamentally different from either "Boy meets girl" or "Boy loses girl." The latter two are fundamentally different from each other in another way. (See Release 1.0, 3-90.)

Also missing is a standard way to solve the interesting problem of negative cases -- ones that relevantly indicate what not to do, rather than ones that are just irrelevant.

The case for CBR

Basically, CBR is not a new technology per se, but a clever and legitimate packaging of a collection of technologies to approach the basic problem of defining, classifying and matching cases. The particular technologies used can include natural language parsing, objects, rules, induction, statistics, pattern-matching, as well as friendly user interfaces, forms and multimedia.

The appropriate technology may vary from domain to domain, and also depend on the form of existing information, user preferences and other factors. Like most of "AI," CBR turns into a number of sensible, not so mysterious techniques on close examination. Moreover, any usable system includes lots of non-AI, such as databases, user interface and so forth. And finally, much of any such system depends on capable developers and users.

CBR, in a sense, turns rich information into objects; hierarchical indexing is an elegant way to build class hierarchies. Overall, CBR allows you to identify similar objects for presentation to a user or manipulation by a computer. Each case is simply an object/frame/schema, with a set of attribute/value pairs, or slots. One of the slots is the desired "answer," whether it's instructions to a user, a script to read to a troubled caller, a set of executable network-administration procedures, a form letter or a print-out of a hackneyed joke or a job candidate's or an employer's name, software packages to match a spec.

--------------------
You can't build bad cases and expect it to do the right thing.
-- David Subar, Inference

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

THREE CASES: ALL CBR, BUT WORLDS APART

Meanwhile, how does CBR work in practice? There are many approaches, so it's simplest to describe three cases from which the alert reader can generalize in a way that a CBR system could not.

There are three major commercial CBR tools: ReMind from Cognitive Systems Inc., CBR Express from Inference Corporation, and Esteem from Esteem Software Inc. in Cambridge City, IN. Then there are richer hybrid systems: Inference's ART-IM expert system tool, which was used to develop CBR Express,
and a forthcoming system from Aion Corporation incorporating Cognitive's CBR technology. (Aion has signed a license agreement with Cognitive to incorporate Cognitive's C libraries as part of a CBR option with its Aion Development System, a rule-based development tool that works on both mainframes and pcs. That option should be available early next year.)

Cognitive's ReMind is closer to a toolbox than a tool; it lets the savvy user-builder manipulate data and cases with just about every version of CBR technique around, including many techniques described above. You can build an extremely clever system if you know what you're doing, and you can also make it fairly intelligible to a normal end-user if you have the interface-building skills and sensibility -- but the system doesn't do it for you. Where ReMind shines is the underlying technology of matching, indexing and induction, and the richness of the development environment.

Esteem has an expert system flavor, focusing not just on case retrieval but on generally rule-based case adaptation. In this respect, although it's a less ambitious system, it's closer to ART-IM than to either ReMind or CBR Express. One other distinguishing feature is Esteem's ability to nest cases, so that a case can derive a feature from cases in another case base.

By contrast, Inference's CBR Express is an easy-to-use focused tool, targeted at help-desk applications and similar interactive selection tasks. It is itself an application written with Inference's ART-IM expert system toolset, and can be extended with it. The user interface is written with Asymetrix ToolBook and is flexible, but the underlying technology is done for you, which -- take your pick -- saves you some work but limits your options. It is simple and hardly requires a programmer; on the other hand, it isn't very flexible unless you use ART-IM. You could also consider ART-IM itself an extremely flexible, high-end tool for building hybrid CBR/rule-based systems, but we focus here on CBR Express.

Cognitive/Aion: We Invented It!

Cognitive Systems Inc. was founded in 1981 by Roger Schank, a Yale professor who more or less invented the field of CBR, along with the notion of "scripts" for knowledge representation, sort of a holistic approach to understanding the context in which events are to be interpreted. "Cases" are a similarly holistic approach to diagnosing or recognizing problems, diagnoses, precedents...in a word, cases.

A better scientist than businessman, Schank left the company in 1988 and now runs a research center for Arthur Andersen. The company is barely profitable, with $3 million in revenues last year, but it has built a nice business consulting and selling bankwire-parsing systems to banks (see Release 1.0, 3-90). Last spring it started shipping ReMind, the product of an embarrassingly long period of development. So far, the company has sold 15 beta copies of an earlier LISP version at $5000 each, and 14 of the current C++ beta for $10,000 each. That $10,000 unit price for a pc or Mac development tool will continue when formal shipments start this spring. Standalone runtimes with the ReMind GUI will cost $3000 for five (the minimum), but Cognitive expects most customers to use its $4500 C function library to integrate a case library in other applications. They will be charged $500 per user for any resulting applications.

Release 1.0 31 January 1992
Cognitive takes a hybrid approach, combining decision trees and template matching with nearest-neighbor searches, and stresses its flexibility as a competitive advantage. Generally, matching cases can be retrieved using the decision tree. The answer to each question rapidly eliminates a huge portion of the case base and allows the search to focus on a tighter and tighter section of the case base. When a selection of cases is retrieved, or when there is no exact match, nearest-neighbor searching comes into play.

It's hard to describe the typical ReMind system, since the tool lets the builder-user build almost anything. Customers so far include some government agencies, but also commercial companies such as Motorola (help-desk, warehouse management to find appropriate parts and computer security auditing), Boeing (engineering decisions and production scheduling), British Airways (747 maintenance), American Express (to detect when its small-business accounts are spending out of pattern), Nestle, Barclay and NCR. The firm also has a long list of prospects, many of which it has done research for.

Esteem: We made it cheap!

Esteem Software Inc. is the multiple-inheritance child of two development/consulting companies, J.A. King Associates and Stottler-Henke Associates, and Klein Associates, an organization of cognitive scientists studying human performance and behavior. The company was created with private funding (and is looking for more) to commercialize five years of work on case-based systems for a variety of customers including DARPA, the Air Force and even a few firms such as AT&T, NCR, General Electric and Household International.

Esteem's hybrid CBR product Esteem sells for a mere $1995, which seems to be still too high to garner a large customer base -- but too low to support the kind of marketing efforts Aion, Cognitive and Inference can muster. It has sold 21 copies since shipments started last October. That was Release 1, built in Intellicrop's Kappa PC, with C functions for nearest-neighbor and template retrieval. The company is working on Release 2, for shipment this summer, which will be written in C++ and will include database access and other improvements. The pc version will still cost $1995, and a UNIX version (new in release 2) will cost $2995. Each developer license also includes a runtime; additional runtimes start at $450. The pc product runs under Windows; 5 megabytes of memory is recommended. Its runtime interface is "bland," company consultant Jim King allows, but you can dress it up with ToolBook or other GUI tools.

The Esteem product grew from a number of customer-specific applications. One example is the Bidder's Associate, currently in use by salespeople at Enginetics, an aerospace parts manufacturer in Dayton, Ohio. Those salespeople use spreadsheets to record and manage their bids for parts manufacturing contracts. Basically, the spreadsheet is a smart form into which salespeople enter the customers' requirements and do a few calculations; then they draw up a proposal. The requirements and constraints all follow a similar pattern, but each set is unique; moreover, some win bids and some lose bids. The Esteem system took its end-users' familiar spreadsheet, with 14 significant features in it, as the basic format for a case. Now the salespeople can retrieve past successful bids and associated proposals to aid them in drawing up new ones.

Release 1.0

31 January 1992
CBR Express: We commercialized it!

CBR Express, from AI vendor Inference, was first shipped last spring. The mainframe (MVS) version costs $50,000 for the first developer copy, including certain facilities of ART-IM and 10 runtimes. Additional developer tools cost $10,000, and additional runtimes cost $4000 or less. The pc version, with the pc version of ART-IM, starts at $10,000 for the developer version, with runtimes also starting at $4000. (Unit pricing is the same, but you can't buy the mainframe version without at least 10 runtimes.)

In the Windows-based pc version (on a 386 with 8 megabytes of memory recommended), the frames and other data are stored as records in Raima's db_VISTA; on mainframes, they're stored in DB2 (with VSAM support on the way). The pc version has links to a call-tracking system also developed by Inference with db_VISTA. It has an Asymetrix ToolBook interface, which makes it somewhat slow but extremely malleable. Using ToolBook and the OLE/DDE facilities of Windows, you can link to all kinds of other pc applications. Customers include Compaq, with a case base used by all its support reps in its North American customer-support operation, on its own machines. Compaq is also working on getting its Tech Support Alliance partners to submit cases and use CBR themselves -- illustrating the value of CBR (and knowledge systems in general) in sharing knowledge broadly.

CBR at American Airlines

We recently took flight 758 to Dallas to see American Airlines, an early CBR Express customer. It is building a CBR help-desk system for its SABRE Travel Information Network division, which sells a turnkey back-office system, Agency Data Systems, to travel agencies.

STIN's Agency Data Systems division is a typical systems vendor; the systems it supports little to do with flights, but with billing, accounting and other business basics. Its help-desk is also typical, dealing with such issues as printer drivers, memory management, application interactions -- and of course the ubiquitous computer that wasn't plugged in. It supports 8000 customer locations with a wall of hard-to-update manuals for several hundred separate software modules and a variety of hardware.

American acquired the system last summer and plans to put it into use this summer. American picked CBR Express for the usual reasons -- to help it compete, to set up a call-tracking system, and of course to handle customer calls efficiently, quickly and consistently. Originally, the plan was just to use Verity's Topic text-retrieval system to manage access to on-line documentation, but they had misidentified the problem (so to speak): it wasn't finding the answers to problems so much as identifying the problems, especially for new customer-support reps, says project manager Lynden Tennison. Now American is still planning to use Topic for on-line documentation, but CBR Express will handle the first step of determining the problem. (Long-run, there will be hot links from CBR Express diagnoses to Topic.)

Despite CBR's purported ease of use, development issues are similar to those for any large-scale system, Tennison discovered. He originally wanted to mass-load existing trouble-call report files into the system. Instead, developers are now constructing sample cases, with properly worded questions for the operator to read off the screen: "Did you remember to plug it in?"
With a smaller, hand-built case base, it's easier to avoid redundant cases differing only by a word or two. One important task was creating a synonym list, so that knobs, buttons, switches and the like are all recognized as the same thing. In addition, Tennison created a style guide to force consistency on the four developers and eventually four domain experts (senior support reps) who create cases for the system.

As a practical matter, it's best to write the questions rather than the problems first: It's no use dealing with a symptom or condition that you can't get a user to tell you about. In addition, if you start with cases, you end up with a lot of redundant, slightly different questions.

Each case is actually an ART-IM frame, which can be reasoned about and manipulated with the full ART-IM toolset. Questions get default weights from CBR Express, or the builder-user can weight them according to relative importance, including absolute scores. Using ART-IM, a developer could write a program to determine correlations to refine the weightings automatically, but Tennison doesn't think it will be necessary to do so. All you need is a system good enough to present the right answer to the operator with a couple of iterations of the process. (It's more useful simply to eliminate useless questions.)

The point of CBR Express is to make it simple, and in its basic form -- without ART-IM extensions -- it is. The problem with CBR Express is that it's so simple it misses some powerful constructs that even the most naive user could understand and benefit from. For starters, each case is unique: Answer one question differently -- for example, whether you checked the plug -- and you have a whole new case on your hands. The system makes it easy to add cases, but you end up using some tools to stem the inevitable proliferation of similar-excepts. Inference's Chuck Williams argues that simplicity in development and maintenance more than overcomes any minor performance problems -- and CBR Express does include automatic heuristics to reduce the search space to a subset for most queries. Moreover, a hierarchical system is fairly rigid in its classifications and expects correct answers; nearest-neighbor is extremely tolerant of inaccuracies and missing data -- especially appropriate for help-desk problems.

A typical CBR Express case

The user starts by typing in a short problem description. For example: "This dog don't hunt." The first section of the runtime module uses simple text-search techniques to matching tool to rank probable cases by matching the significant words and trigrams (three-letter sequences designed to accommodate misspellings and word variants), throwing out (temporarily) any cases with no words in common with the input case. This is similar to most natural-language categorization or database front-end systems, which just throw out extraneous words anyway -- as opposed to parsers, which depend on grammar and more verbose formulations. The top-ranking cases/questions are then displayed to the user. For example, "This sucks! It can't even print a memo; I feel really burned!" would retrieve not just printer cases but possibly also "Toaster temperature too high" and "Bilge-pump failure" (if you had a really broad case base!).

Obviously, the operator has to enter relevant information, using words that discriminate well between cases. Whereas a direct user of a public system
might type a lot of irrelevant stuff, an experienced operator will type only terse, useful phrases, such as "Printer failure," whatever the customer on the line might say. In the end, the system depends on sensible operators, and a particular case base contains a limited domain.

Now the user must answer some of the questions listed with the top-ranked cases; usually five or ten of them are displayed, a number set by the user. The questions are listed (unduplicated) in order of weight for the highest-ranked case, the second case, and so on. The user answers whichever question he or she prefers -- or whatever someone on the phone has already mentioned. In demo mode, the system conducts a search after each answered question; most operators would probably turn this off and let a few answers collect, since each search takes a second or two even with the various performance aids activated.

While there's a large, possibly proliferating number of discrete, unique cases, the system's efficiency is maximized and the user's time is minimized if questions are kept to the minimum. Redundant questions are the worst: "Is the quality poor?" and "Does the output look bad?" The more cases a single question can discriminate among, the better. For example, if you've asked "Does it have a 386?" "Does it have a 486?" and so forth, you're better off asking, "What kind of processor does it use?" An enhancement is the use of background rules added in ART-IM, such as "If the user has said the self-test works, don't ask whether the unit is plugged in" -- or a rule to that effect. (You could also build an ART Monte Carlo simulation to run through the case base and eliminate questions that didn't discriminate among any cases; in essence, that's what an induction tool does. The problem is that such a situation may reflect a badly worded question; or it may reflect an insufficient case base, and the question may need to be reinstated later. A domain expert can use CBR Express's test-case facility to find redundant or "missing" cases.)

Using the new answers, CBR Express conducts a second search, to find cases that match the answers given for the input case. It scores a subset of the case base again, allocating an optional proportion of the score (typically 20 percent) to the description match, and the rest to the values of answers to the questions. (It avoids handling the entire case base by using an index to determine which cases include answers to the questions under consideration, and rescoring only those.)

Once again, it produces a list of the highest-ranked matches to the target case. At this point, the operator can select a case and see the recommended action, or answer further questions based on a new set of questions that appears, corresponding again to the highest-ranking possibilities. Once again, the system adjusts the scores of the cases involved, and produces a new list. This goes on until the system flashes a match, the user gives up, or the system declares that nothing matches the user's answers.

In that case, the user tags the new case as "unresolved," saving the description and all the questions and their answers. The case is forwarded to an expert who solves it, and either determines that it is in fact an existing case (using a test-case facility that runs through the entire case base) or enters it as a new case -- a simple process. The expert just stores it with a new name, along with any new questions and their answers.
MOBILE COMPUTING: IS THE PEN BESIDE THE POINT?

As part of PC Forum, we had hoped to offer each attendee the use of a pen computer for the duration of the conference (in return for a credit card number). Each machine would have had the conference program, an indexed list of attendees and descriptions of company presentations, and note-taking software. It would also have had the requisite tools to communicate with the Forum Notes server. The idea was that on Wednesday (Pednesday) morning, the audience and the vendors would gather to assess the experience.

Unfortunately, no vendor has yet proved able to deliver 500-plus systems of the power we needed. (The logistics of managing a multi-vendor base totaling 500, even if we could get it, overwhelm us.) Instead, we will use a limited number of pen computers for some minor data-entry chores at registration -- to be performed by attendees themselves -- with applications from Slate and Pen Pal.

Momenta bravely tried a small-scale loaner program of the sort we wanted, at Stewart Alsop's recent Demo conference (which, however, lacked the forum for debriefing afterwards). There were a number of problems, some specific to Momenta, and some more general. And at least Momenta had machines available. But that was the first problem: There were only about 40 (plus spares), so that there was no critical mass of users to share experiences. Second, the machine itself was heavy and bulky enough that you couldn't casually carry it around -- and it was too dark to see the screen in most of the conference sessions anyway. Yes, we should have left our keyboard behind, but the message is clear: Physical form factor matters!

Finally, there was no compelling reason to use it. If we were to run our experiment after all, we would offer a general training session and omit all paper documentation for the Forum, forcing people to rely on the system -- a decision that might well backfire. (People could surreptitiously fax themselves copies of the schedule to their hotel rooms, for example, and PR people might hand them out to favored journalists, while earnest start-ups would vie to present them to, say, Novell's Darrell Miller or venture capitalist Ann Winblad.)

One thing that might have made the Momenta (or any other pen computer) more compelling would be to have the user's own data in it. Momenta's John Rizzo notes that the Momenta folks themselves used their machines for taking notes (in ink) it was useful for sending faxes and Momenta registration forms to hotel guests, redoing a budget and other tasks. Except for the note-taking, all of these required specific kinds of information and software that the guest users didn't have.

A second capability we hoped to include was some sort of connection to the conference Notes system, so that attendees could take notes or do other work during the sessions, and then communicate with each other through Notes afterwards. The guy with the snappy comeback to the speaker's outrageous statements, the developer with the almost-ready package that just matches the needs of the dissatisfied customer in the audience, the would-be investor trying to set up a meeting, could all communicate with their targets through Notes. Unlike last year, they would not have to sit long at a terminal to do so; instead, they could just plug in their pen tablet and upload their comments, messages and questions.

Release 1.0

31 January 1992
Active doodles

The vision we have is a transformation of those doodles that many people make during meetings: "Call Juan." "Send Alice article on case-based reasoning." "Buy -- and read!!! -- Regis book!" "Introduce Harry to Fred." "Follow-up on Williams proposal?" "Lunch with Julia." The amount of text produced may be small, but the amount of thinking involved and the activity potentially generated may be substantial. But usually, when people walk out of these meetings, the notes stay on paper. Some of these doodles could now be active, and could be implemented by software agents: With the right tools, articles can be faxed, memos sent, databases queried, boilerplate letters composed and personalized, by the host computer system with a few commands and specific data from the user's pen machine. Even buying the

PC FORUM UPDATE

The PC Forum is almost sold out. If you still wish to attend, please check with us before sending in your registration.

GENERAL SESSION SPEAKERS

| Barry Berkov         | CompuServe          |
| Jim Cannavino       | IBM                 |
| Michael Dell        | Dell Computer       |
| Brian Dougherty     | GeoWorks            |
| Kamran Elahian      | Momenta             |
| Doyne Farmer        | Prediction Company/Santa Fe Institute|
| Bill Gates          | Microsoft           |
| Jeff Hawkins        | Palm Computing      |
| Frank Ingari        | Ontos Corporation   |
| Bill Joy            | Sun Microsystems    |
| Jerry Kaplan        | GO Corporation      |
| John Landry         | Lotus Development   |
| David Liddle        | IBM                 |
| Tom Malone          | MIT Center for Coordination Science |
| Frank Moss          | Tivoli Systems      |
| George Perry        | Prodigy Services    |
| Vern Raburn         | Slate Corporation   |
| June Rokoff         | Lotus Development   |
| Richard Schwartz    | Borland International|
| Ralph Terkowitz     | Washington Post Co. |
| Allan Weis          | Advanced Network & Services |
| Gil Williamson      | NCR/AT&T            |
| Dave Winer          | UserLand            |

In addition, there will be company presentations from: America Online, American Information Exchange, Channel Computing, Client/Server Technologies, Da Vinci Systems, Delta Logic, Edify, GeoWorks, HyperDesk, IBM, IntelliCorp, Maxis, Ontos, PenPal, Reach Software, Serius, Servio, Slate, Transarc, Trilogy Development, UNIX System Labs and XcelleNet. We will also hold two roundtables, with participants from Eastern Europe and the dealer/distribution community, and a special Electronic Frontier Foundation information session.

Release 1.0 31 January 1992
book will become possible in the future, as more and more transactions occur on-line rather than in stores. Reading the book, however, is still up to the user. (See Release 1.0, 6-90, 1-91.)

The real issue is live connectivity to the connected world; a machine must be "connected" both to the user (small enough to carry around, handy enough to use at all times in all places) and to the outside world (through data communications). The pen is a handy part of the user-to-machine connectivity, but none of it matters if the machine itself isn’t connected.

Thus we listened with great interest to a variety of connectivity announcements -- beepers, wireless e-mail, global connectivity -- from a variety of vendors. Anything that lets you keep your mobile machine closely connected to the stationary source of data, server applications and connections with other people is a good thing -- and the greater the bandwidth and the more persistent the connection the better. We are still working on providing such a system to communicate with the Notes server for the Forum. This system -- this year, anyway -- will work more as a notifier than as a full two-way communications device. It will probably simply encourage people to pick up mail and messages waiting for them on the Notes system.

---------
Imagine a world without "mobile drinking": There would be no glasses, cans or even those ubiquitous promotional coffee mugs. You could drink only from taps or faucets; beer barrels and coffee urns would be considered "portable." Innovative people would run hoses all over the place, but messy leaks and other mishaps would be common.
---------

What goes over the wire

So the question is what do you do once your pen machine is hooked into Notes -- or any other "home" system? How do you easily get your files in and out with closely coupled software? (The number of suppliers involved in any single wireless communications announcement is daunting.) Traveling Software -- alone, more or less -- has come up with a solution in conjunction with a number of pen OS vendors.

As we noted in the summer of 1990, the pen computer is really an interface to the larger, grounded home system. That's why it can effectively use a tailored operating system rather than a pen-retrofitted version of existing standards. Consider PenPoint or Momenta's MADE (for Momenta Application Development Environment) as the intelligent equivalent of an X terminal; the server functions go on at the server, but there's a standard piece of local intelligence that communicates with the server and also is configured for the individual client. However, the "X" component's job is not just interface, but interface intelligence -- intelligent interaction with the user. That interaction translates not just simple actions such as "penup" or text strings, but turns, say, a user's selection of a person object into an SQL query, or a memo into a formatted, properly addressed and routed e-mail message. (Okay, maybe not this year.)

Just as the purpose of X is to limit the amount of data and code that needs to travel between client and server, and to allow a variety of clients and
servers to communicate, so does the pen-X minimize communication by maximiz-
ing intelligent interpretation on either side of the client-server inter-
face. Of course, since the server environment -- the networked pc with all
its files and standard applications and protocols -- was there first, the
pen OSes have to figure out how to communicate with the existing infrastruc-
ture rather than the other way around.

Traveling Software’s Universal Communications Object

But there’s also the communications infrastructure to deal with, a specialty
of Traveling Software -- which grasped the concept of mobile computing much
earlier than most of us. The user community, meanwhile, has been waiting
(knowingly or not) for machines that were pleasant and useful to carry
around. Now the machines are becoming so easy to carry around that the next
big market will be for wireless communications; plugging into a stationary
telephone or network isn’t enough.

When the communications are established, however, the software on either
side needs to know what to do. TSI’s Universal Communications Object is a
code module that manages a variety of communications functions, sensing what
machine it’s on and what machines/protocols it’s talking to. Any applica-
tion simply needs to talk to the UCO, instead of to a modem, a comm port, a
printer or parallel port, a network protocol, and so on. The UCO module is
already part of a number of mobile systems, including Poqet’s and the HP
95LX as well as TSI’s new retail LapLink Pro product.

Traveling Software has just announced two new UCO products, PenConnect and
Pen LapLink, which allow the PoqetPad, NCR’s 3125 and other DOS or Pen-
Windows computers to work in client-server mode with DOS/Windows pcs. The
pen machine can work as a server for the stationary machine, or the other
way around. Thus the traveling CPA or saleswoman or tech rep can bring his
data or software and use it with the customer via the customer’s stationary
computer. How did he get it? By using his pen computer as a client to his
own stationary machine, of course.

LapLink Pro for PenPoint, developed with Traveling Software but marketed by
Slate, also uses the UCO but adds considerable functionality. It works over
a modem, which the other tools don’t do for performance and other nudgy
technical reasons. It answers a basic complaint that people would have had
if PenPoint were already shipping -- its lack of a DOS file format. LapLink
Pro builds an installable DOS file system on the PenPoint machine -- some-
thing PenPoint allows but DOS itself would not. (In theory, you could also
install a UNIX file system or Microsoft’s High-Performance File System when
it comes out.) Thus PenPoint can easily load things from or send things to
any DOS machine. The software resident on the DOS server can be loaded re-
motely, as long as the local recipient puts the machine into receiving mode
with a couple of keystrokes -- a useful security procedure.

To the communications capabilities, Slate has added queuing and other fa-
cilities, so that a user can fill his outbasket with timed or when-connected
tasks. When the pen machine gets connected, LapLink Pro manages the se-
quence of work, sending out e-mail, uploading database transactions, down-
loading files and doing whatever other work has piled up. Yes, we still
need a way to reconcile files -- but this is a start in the direction of
sewing up the seams created by mobile computing.

Release 1.0

31 January 1992
RESOURCES & PHONE NUMBERS

Jan Aikins, Bernadette Minton, Aion Corporation, (415) 328-9595; fax, 321-7728 or (415) 328-0624
Lynden Tennison, American Airlines, (817) 963-9431; fax, (817) 963-2464
Ralph Barletta, Steve Mott, Cognitive Systems, (617) 742-7227 or (203) 773-0726; fax, (617) 742-1139
Jill King, Esteem Software, (317) 478-3955; fax, (317) 478-3550
Chuck Williams, David Subar, Inference Corporation, (310) 322-0200; fax, (310) 322-3242
John Rizzo, Momenta, (415) 969-3876; fax, (415) 969-3717
Vern Raburn, Dan Bricklin, Dottie Hall, Slate Corporation, (602) 443-7322; fax, (602) 443-7325
Mark Eppley, Traveling Software, (206) 483-8088; fax, (206) 487-1284

For a fuller technical/practical discussion of case-based reasoning, see Paul Harmon's Intelligent Software Strategies newsletter, volume VII, numbers 11 and 12. It is published by Cutter Information Corp., (617) 648-8700; fax, (617) 648-8707.


COMING SOON

- Forum issue.
- Mail matters.
- Constraint-based tools.
- The attention society.
- And much more... (If you know of any good examples of the categories listed above, please let us know.)
## Release 1.0 Calendar

<table>
<thead>
<tr>
<th>Date</th>
<th>Event Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>February 11-13</td>
<td>*NetWorld 92 - Boston. Sponsor: Bruno Blenheim. Call Annie Scully or Mark Haviland, (800) 829-3976 or (201) 346-1400.</td>
</tr>
<tr>
<td>February 23-26</td>
<td>Organizational computing, coordination and collaboration: Theories and technologies for computer-supported work - Austin, TX. Sponsored by The University of Texas, RGK Foundation and NCR. Call Andrew Whinston, (512) 471-8879.</td>
</tr>
<tr>
<td>February 23-26</td>
<td>**EDventure Holdings PC (Platforms for Computing) Forum - Tucson, AZ. New alliances and new technology lead to &quot;A New Landscape.&quot; You read the newsletter; come meet the players and try their tools. Call Daphne Kis, (212) 758-3434.</td>
</tr>
<tr>
<td>March 1-7</td>
<td>Computer science conference - Kansas City, MO. Sponsor: Ass'n for Computing Machinery. Call Don Nowak, (212) 869-7440.</td>
</tr>
<tr>
<td>March 2-6</td>
<td>*Artificial intelligence for applications conference - Monterey. Sponsored by IEEE. With a panel on CBR featuring Ralph Barletta, Cognitive; Chuck Williams, Inference; and Jim King; moderated by Esther Dyson. (See this issue.) Call Bernardette Minton, (415) 688-8607.</td>
</tr>
<tr>
<td>March 2-6</td>
<td>OpCon West - Santa Clara. The west-coast session of Softletter's twice-yearly conference for operations managers. Call Tom Stitt, (617) 924-3944.</td>
</tr>
<tr>
<td>March 4-7</td>
<td>Computers in Libraries - Washington, DC. Sponsored by Meckler. For librarians and information managers. Call Suzanne Li, (203) 226-6967 or (800) 635-5537.</td>
</tr>
</tbody>
</table>

Release 1.0

31 January 1992
<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
<th>Details</th>
</tr>
</thead>
</table>
| March 10-12| International conference & exposition on multimedia and CD-ROM - San Francisco.  
With Bill Gates.  
Sponsored by Microsoft.  
Call Min Yee, (203) 352-8254. |                                                                                                                                  |
| March 10-15| Founding workshop in adaptive computation - Santa Fe.  
Sponsor: Santa Fe Institute.  
Setting the agenda for the study of complex adaptive systems, with the field’s leaders.  
By invitation; call Ginger Richardson, (505) 984-8800. |                                                                                                                                  |
| March 11-18| Hannover Fair CeBIT ’92 - Hannover, Germany.  
Sponsor: Hannover Fairs USA.  
Everything you’re interested in, and a lot more.  
Call Donna Peterson Hyland, (609) 987-1202. |                                                                                                                                  |
Contact: Lance Hoffman, (202) 994-4955; fax, (202) 994-0227. |                                                                                                                                  |
| March 18-21| SPA spring symposium - Seattle.  
Call Karen Johnson, (202) 452-1600. |                                                                                                                                  |
| March 24-26| DE/EXPO ’92 - San Francisco.  
with Philippe Kahn, Charles Bachman, Roger Sippl.  
Sponsored by NDN Enterprises.  
Call Victoria Lukanovich, (415) 941-8440 or (800) 2DB-EXPO. |                                                                                                                                  |
Sponsored by the American Association for Artificial Intelligence.  
Call Annette Eldredge, (415) 328-3123. |                                                                                                                                  |
"Distributed object computing."  
Sponsored by Patricia Seybold’s Office Computing Group.  
Call Deborah Hay, (617) 742-5200. |                                                                                                                                  |
| April 6-9 | Spring Comdex - Chicago.  
Sponsored by the Interface Group.  
Call Elizabeth Moody, (617) 449-6600. |                                                                                                                                  |
| April 6-9 | SunWorld Expo - Santa Clara.  
Sponsored by World Expo Corporation.  
Call Ron Toran, (508) 879-6700 or (800) 545-EXPO. |                                                                                                                                  |
| April 6-10 | *COMTEK ’92 - Moscow.  
Call Steve Woods, (203) 834-1122; fax, (203) 762-0773. |                                                                                                                                  |
| April 22-24| *Asilomar microcomputer workshop - Asilomar, CA.  
By invitation only, featuring some of computerdom’s more original minds.  
Sponsored by IEEE.  
Call Brian Berg, (408) 741-5010. |                                                                                                                                  |
| April 25-29| Computer game developers conference - San Jose.  
Sponsored by CGDC.  
Call Ernest Adams, (408) 374-4340. |                                                                                                                                  |
| April 28-30| *NetWorld 92 Europe - Frankfurt, Germany.  
Sponsor: Bruno Blenheim.  
Call Annie Scully, (201) 346-1400. |                                                                                                                                  |
| April 30-May 1 | National Irish software conference - Dublin.  
Sponsored by the Irish Software Association.  
With Esther Dyson.  
Contact: Tom Winter, 44 (232) 738-507; fax, 44 (232) 738-493. |                                                                                                                                  |
| May 3-6 | ITAA (ADAPSO) spring management conference - New Orleans.  
Call Shirley Price, (703) 284-5355. |                                                                                                                                  |
| May 3-7 | Human factors in computing conference - Monterey.  
Sponsored by ACM.  
Call Rachel Greiper, (212) 869-7440. |                                                                                                                                  |
| May 5-7 | Executive Uniform symposium - Santa Barbara.  
"Migrating to integrated open systems: Tools, tactics and tradeoffs."  
Sponsored by UniForum, X/Open and Patricia Seybold’s Office Computing Group.  
Call Deborah Hay, (617) 742-5200. |                                                                                                                                  |
| May 11-13 | International Markup/Documentation Europe ’92 - Amsterdam.  
Sponsored by Graphic Communications Association.  
Call Joy Blake, (703) 519-8160. |                                                                                                                                  |
| May 18-22 | INTEROP 92 - Washington, DC.  
Sponsored by Advanced Computing Environments/Ziff.  
Call Wendy Gibson, (415) 941-3399. |                                                                                                                                  |
<table>
<thead>
<tr>
<th>Date</th>
<th>Event Description</th>
<th>Contact Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>May 31-June 2</td>
<td><strong>ATRE - Sahoro, Japan. Executives meet.</strong> Sponsored by Dasar. Call Alex Vieux, (415) 321-5544.</td>
<td></td>
</tr>
<tr>
<td>May 31-June 3</td>
<td><strong>SPA Europe's third annual conference - Cannes, France.</strong> By Europeans, for Europeans, with Americans invited. Sponsor: Software Publishers Association. Call Ken Wasch, (202) 452-1600, or Viviane Lemonnier, 33 (1) 4692-2703; fax, 4692-2531.</td>
<td></td>
</tr>
<tr>
<td>June 1-3</td>
<td><strong>Collaboration '92 - San Francisco.</strong> Sponsor: Graphic Communications Association. Call Marion Elledge, (703) 519-8160.</td>
<td></td>
</tr>
<tr>
<td>June 1-5</td>
<td><strong>FGCS '92 - Tokyo.</strong> Sponsored by the Institute for New Generation Computer Technology. Contact: Hidehiko Tanaka, 81 (3) 3456-3195; fax, 81 (3) 3456-1618; e-mail, <a href="mailto:fgcs92@icot.or.jp">fgcs92@icot.or.jp</a>.</td>
<td></td>
</tr>
<tr>
<td>June 15-18</td>
<td><strong>International Computer Forum - Moscow.</strong> Sponsored by the International Computer Club. Call Levon Amdilyan, 7 (095) 921-0902, or &quot;levon&quot; on MCI mail at 439-1034; or Esther Dyson at (212) 758-3434.</td>
<td></td>
</tr>
<tr>
<td>June 15-19</td>
<td><strong>Artificial Life III - Santa Fe.</strong> Sponsored by the Santa Fe Institute. How to grow your own. Call Christopher Langton, (505) 984-8800.</td>
<td></td>
</tr>
<tr>
<td>June 23-25</td>
<td><strong>PC EXPO - New York City.</strong> Sponsored by Bruno Blenheim. Call Annie Scully, (201) 346-1400 or (800) 829-3976.</td>
<td></td>
</tr>
<tr>
<td>June 29-July 3</td>
<td><strong>ECOOP '92 - Utrecht, Netherlands.</strong> Sponsored by Software Engineering Research Center. Contact: Gert Florijn, 31 (30) 322640; fax, 31 (30) 341249; e-mail, <a href="mailto:ecoop92@serc.nl">ecoop92@serc.nl</a>.</td>
<td></td>
</tr>
<tr>
<td>June 30-July 1</td>
<td><strong>First international conference &amp; exhibition on advanced service and HelpDesk automation - Strasbourg, France.</strong> Sponsor: Applied Workstations and ServiceWare. Contact: Jeff Pepper, (412) 826-1158; Tim Lewis, 44 (306) 773331; fax, (306) 77696.</td>
<td></td>
</tr>
<tr>
<td>July 14-16</td>
<td><strong>AAAI/IAAI '92 - San Jose.</strong> Sponsor: American Association for AI. Call Mary Livingston, (415) 328-3123.</td>
<td></td>
</tr>
<tr>
<td>July 20-23</td>
<td><strong>Object World - San Francisco.</strong> Co-sponsored by The Object Management Group and World Expo Corp. Businesspeople’s answer to OOPSLA. Call Bill Hoffman, (508) 820-4300.</td>
<td></td>
</tr>
<tr>
<td>July 26-31</td>
<td><strong>Siggraph '92 - Chicago.</strong> The major computer graphics shows. Sponsored by ACM/Siggraph. Call Ann Leuck, (312) 644-6610.</td>
<td></td>
</tr>
<tr>
<td>August 1-4</td>
<td><strong>GroupWare '92 - San Jose.</strong> Sponsored by Lotus Development and SRI International. Keynote by Jim Manzi. Speakers include Irene Greif, Thomas Malone and Esther Dyson. Call David Coleman, (415) 282-9151 or (800) 247-0262.</td>
<td></td>
</tr>
<tr>
<td>August 13-16</td>
<td><strong>International BBS and electronic communications conference - Denver.</strong> Sponsor: IBECC. Call Terry Travis, (303) 426-1847.</td>
<td></td>
</tr>
<tr>
<td>September 10-12</td>
<td><strong>ETRE - Vienna, Austria.</strong> Sponsored by Dasar. Call Alex Vieux, (415) 321-5544.</td>
<td></td>
</tr>
</tbody>
</table>

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

Release 1.0 31 January 1992
Please enter my subscription to Release 1.0 at the rate of $495 per year in the U.S. and Canada. Overseas subscriptions are $575, airmail postage included. Payment must be enclosed. Multiple-copy rate on request. Satisfaction guaranteed or your money back.

Name ____________________________________________________________
Title ____________________________________________________________
Company _________________________________________________________
Address __________________________________________________________
City __________________________ State _______ Zip _________________
Telephone _________________________________________________________
How did you hear about Release 1.0? __________________________________

☐ Payment enclosed.
☐ American Express #_____________———-———— Expires ________
Signature _________________________________________________________
☐ Please send me additional information on your multiple-copy rate.

Please fill in the information above and send to:

EDVENTURE HOLDINGS INC.
375 PARK AVENUE, SUITE 2503
NEW YORK, NY 10152

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

Daphne Kis
Associate Publisher