WORKFLOW

In our last issue we talked about groupware in general, examining two major platforms: Lotus's highly successful Notes, best-suited for information-sharing, and Microsoft's still vaporous Cairo, a modular operating environment that will support and could incorporate anything. (Microsoft's near-term groupware products, however, are targeted at scheduling and lightweight workflow.) We also examined two information-sharing tools, CM/1 and grapeVINE. In this issue, we look at the other side of groupware, workflow.

Information-sharing is basically a passive function. Although messages are filtered and routed to people according to content, the system itself makes no attempt to follow them, and the information isn't generally disseminated in any particular order. It's there for the user when and if he wants it.

By contrast, workflow is an active function, whether routing a message from one designated person to another, or a complicated, multi-part, multi-person application with conditional steps, parallel or subordinate activities and different documents or information sources. In this issue we discuss workflow and the underlying technologies that best suit it. We consider Lotus (again) and Action Technologies as technology providers, as well as other representative vendors. (The list of players is too long to cover in full.)

For contrast, since it is basically an automated information-sharing tool rather than a workflow manager in this implementation, we consider the new BeyondMail for Notes. Its very existence testifies to Lotus Notes' success.

In considering workflow systems, there are two broad issues. First: Does the workflow simply list steps or a route for a particular item of work? Or is it a set of transactions with a beginning and an end -- perhaps including nested transactions, alternative conditions and actions, multiple work objects. In other words, is it a script, or a full-fledged application with process integrity and some notion of completeness?

Second: How is the workflow controlled?

Why workflow?

Ostensibly, the basic reason for workflow is to automate routine work. Then you examine the exceptions or desired changes in the routine,

UNFOUNDED RUMORS -- PAGE 20

EDVENTURE HOLDINGS INC. 375 PARK AVENUE, NEW YORK, NY 10152 (212) 758-3434
figure out how to express them and make them routine and automate them, and so on ad infinitum. That is, the real reason for workflow tools is not to handle the routine, but to be able to define and handle exceptions reliably. The thousands of routine loan applications are easy; the trick is to handle the unusual ones without dropping the ball. The 99 operations that go smoothly can be handled with legacy-style systems, but it takes a good workflow tool to model the myriads of ways the hundredth operation can go astray -- and how to rescue it.

Thus, aside from power and the ability to handle sheer volume, the mark of a good workflow system is flexibility -- both in design and in operation. It should be easy for users both to build a workflow system -- to define ways to handle exceptions and to change the workflow in response to changing conditions (i.e., exceptional conditions that become permanent). In a service-oriented world that's moving ever faster, a company that can change its processes as well as its products has a powerful competitive advantage.

Think of the software video of the Eighties: Five yuppies sit around a table. One says, "What if we changed this figure here..." The graph on the screen magically redraws itself.

The video of the Nineties? A racially mixed group is discussing a merger. "Now that we've acquired SoftBusters, we have to link them into our customer-support process. When the message comes in, it should go to support first, and then to the Soft-Busters QC team..." The flow chart magically redraws itself. And of course the workflow code is automatically updated.

And what is groupware when tasks are fully automated -- that is, when users are no longer necessary? Basically, it becomes the equivalent of a scripting tool or job control language, but implemented with a tool that's easier to use and understand. Whether it involves users or just applications, workflow tools allow a user to build and visualize a business process as a sequence of business steps rather than sets of code modules.

Lite or loaded

As noted, we divide workflow into scripted workflows and workflow applications with process integrity. The first kind may be ad hoc or extremely repetitive; regardless, they are not very complex and operate in a single loop. Things can't easily go astray, and the sequence of steps is fairly easy to define. Beyond Incorporated probably has the most powerful rule-based scripting tool, hidden inside an easy-to-use mail "user agent." Now the company is expanding its reach; see page 13.

Other examples of script workflows are "forms-based or e-mail-based routing": The forms and message types give users an easy way to specify simple routing and branching (scripts). Vendors of this kind of workflow now include most notably Reach Software, and will include most forms and e-mail vendors: Da Vinci, Delrina, JetForm and Microsoft (with Windows for Workgroups). However, there's usually no underlying mechanism to monitor the progress of each item as it goes through its paces, and no design tool to validate the scripts/routes. (Of course you could add one.)
The workflow as "application with process integrity" approach is key to mission-critical applications, sets of workflows that need to interact, and other more complex environments. If your e-mail message doesn’t go through, you'll probably figure it out and follow up. But a major capability of industrial-strength workflow is to define and execute routine, repetitive processes in a way that does not require someone to follow up on them individually. The software should be able to find and either handle exceptions or refer them to a person who can.

Here, Action Technologies is the conceptual leader, although vendors such as FileNet and Staffware (London) already have large installed user bases. Of course, anyone can build such an application, even using COBOL. The contribution of a workflow tool is (1) to make it easy to do so, and (2) to provide some methodology and validation tools to help manage the process.

A number of vendors do (1), using a variety of graphical tools, menus and pick lists. But Action is the clear leader in (2). Even though some people find its philosophical underpinnings irrelevant or off-putting, they appreciate the results. Action alone understands and makes explicit the offer/counteroffer/delegation/commitment relationships that underlie the flow of work and provide the framework for completion of tasks as opposed to stepping through actions.

Lotus has licensed the Action Technologies technology, in a strategy to leverage its (Lotus's) established market position in information-sharing groupware into leadership in workflow as well. It will sell the technology under its own banner, complete with SmartIcons and other accoutrements.

Some basics on workflow

As we discussed in August, workflow is modular: There's the control of the process, and then the actual work that's performed. In theory, that's great, since the application can make decisions about the work routing based on its content: Documents about certain topics go to Juan, others to Alice; amounts over $500 go to Henry, under $500 go straight through to accounting. News items on Hungary go to Gabor; news items on Poland go to Bogdan; news items on Central Europe more broadly defined go to both.

The problem comes when you want to change something: Can you easily abstract out the roles of the users the work is routed to? Can Tibor replace Gabor? Suppose the content of the work changes, and you want to use another database, but that a step is somehow tied to a group with terminal access to a particular host? You need abstractions such as roles and workflows and data sources that can be resolved later into individual people, procedure calls and database tables.

What's needed is two separate modules that can communicate -- but that are not dependent on each other. As we've said, there is a logical distinction between the data managers that manage work transactions and records, and those that manage workflow. (Physically, they may be on the same hardware server or even use the same database software.) You do want seams for the developer, even if the user shouldn't see them. In general, the workflow knows about the data and the work but it does not interfere in the particulars of each work step or the details of the user's environment.

Release 1.0
30 September 1992
The tools: design and execution

There are really two aspects to process integrity in workflow: design and execution. In the design, you need good tools for design, validation and links to the outside world; in execution, you need management tools to manage and monitor the flows and detect problems. Of course, you can build those yourself, but a database can handle these as part of its basic operations as a specialized form of transaction management.

**Design** requires appropriate development tools, with support for building workflow applications. A database application-building tool is a natural for building a workflow application as an integral set of transactions rather than a sequence of steps. If you use a relational database as the foundation for workflow, you get many of the transaction-building tools and the process/transaction management for free along with the database engine (although you may want to customize them to talk in workflow terminology).

But you can also tidily keep workflow data, in, say, Lotus Notes, with each event that happened to a particular document recorded as a line on a form; it's just then that you need a separate workflow engine to define the workflows and perform the transaction management and monitoring (which is just what Action is providing with its Workflow Engine). If the content data -- documents, customer records or the like -- happens to be in Notes as well, then you end up with a nicely integrated system. Given Lotus's market share with Notes, that's likely to be a common situation -- and one that Lotus can now exploit.

In this context, e-mail and database can be complementary. If your groupware system is based on e-mail, you still need a place to store the information about the flow, as well as the messages themselves. Otherwise, that information is hidden in the content, as in an e-mail system that allows you to find a "conversation" or "discussion" by searching for all the messages with a particular item in the subject field -- or by all the items with Alice in the TO or FROM field. (What about the stuff that was only cc'd to her? Don't forget that!)

E-mail can provide a transport mechanism whereby remote (or local) clients communicate with the workflow application and database; alternatively, users must log in to the workflow database/environment directly -- or have some other means of communicating with it.

**Execution** requires an engine to manage the process, monitor execution, and so forth. These are all traditional database tasks, customized for workflow transactions and triggers. The workflow engine must monitor the state of the workflow much as a database monitors the state of data transactions.

---

1 People now write script workflows somewhat the way they wrote applications before databases; you simply go through a sequence of steps and make changes to a set of records in a file. There's no inherent guarantee of integrity or completeness.
ACTION TECHNOLOGIES

Although workflow applications were pioneered by companies such as FileNet, which handled the movement of images and documents through an organization as a tangible process and a practical application, much of the thinking about workflow and business processes has come from Action Technologies. The company had a reputation for being slightly strange, with its talk of "conversations" and "speech acts," but it was simply ahead of its time.

Action started out with MHS, a substrate and protocol set for messages, and the Coordinator, a tool for managing workflow conversations -- or series of related e-mail messages with defined meta-meanings (offer, counteroffer, acceptance, etc.). The Coordinator didn't really have defined transactions, but more the idea of states of a conversation, which remained open until both sides declared it closed.

What Action is selling now is yet one level higher -- tools for the design, management and execution of workflow templates. Action has licensed its Workflow technology to several companies, including Da Vinci, LaserData, IBM and Verimation (more below). That is, The Coordinator implemented simple, single-loop workflows, but Action Workflow lets you design complex workflow models ones so that interacting workflow instances (with variations) can be enacted over and over, managed by the Workflow server.

A workflow design tool needs to build and model the consequences of the interactions; a workflow server needs to handle those interactions. In the Action model, each workflow consists of at least four stages:

• an opening offer (or request for proposal), which initiates the process. This could be as simple as someone sending in a loan application, SoftBusters debugging a new payroll system, or a customer ordering a new jet plane.

• negotiation, where the parties agree to the precise terms and conditions of the contract, including timing. Do you pay your employees monthly or weekly? How many seats would you like in your airplane? What color? When would you like it delivered?

• performance of the work. This may involves lots of sub-workflows. The loan application may be routed through a number of sub-workflows. Each of these steps can involve several iterations between customer and supplier, and each may in turn delegate work(flows) to other parties. In order to build the jet plane, you may have sub-workflows with engine suppliers and with hundreds of employees.

• acceptance by the customer. Finally, the customer declares that the plane flies. A new sub-workflow begins as the maker asks for money.

Obviously, the world isn't quite as tidy as these bullet points suggest. Workflows aren't sequential; the customer may change his mind several times, you may need to change subcontractors, and the process of collecting the money usually moves in tandem with the performance of the work in such a big contract. There can be parallel workflows involving a large number of airplanes. One triggers another; a failure sets off a designated recovery workflow (in a well-designed system). And so forth.
But these four stages provide a framework around which to build workflows that interact with each other. To some extent at least, they clarify what is going on and make it easier to monitor and manage. They are a more powerful, more useful abstraction than simply: First Alice gets the paper, then Juan gets it, then Sam signs it, and it goes to Fred. "What is the state of the workflow?" "Who's still unhappy?" you want to know; not just "Where is the paper?"

Action provides not just drawing tools and hooks to messaging services and a database, but a methodology for building and validating interacting workflows. It comes in several parts:

- the Workflow Analyst lets someone design a workflow.
- the Application-Builder tool lets a programmer tie those workflows into the customer's real world of specific e-mail tools, databases, applications and other data sources, using Action's Standard Transaction Formats (sort of a protocol or set of workflow procedure calls for communication between the workflow engine and the variety of e-mail systems, databases and other applications out in the user's world). The resulting executable templates are stored in a Definitions database.
- the Workflow server (with programs written in the SQL-like Workflow Language) manages the execution of the definitions, creating a new workflow instance each time a process begins. It stores the data about the workflows and their states in its transactions database (in Notes, in the Lotus Notes case, or otherwise typically in a relational database). This is the instance data, individual records recording the particulars and status of each workflow (people and data involved, which steps have been completed, and any metadata that influence the sequence/selection of steps).
- the Reporter tool, which lets a user both manage the real-time execution -- or handle exceptions -- in workflows as they execute, and inspect the past performance of the system or any individual workflow. It also provides information necessary for maintenance and redesign.

The STFs are a standard way of building and representing the hooks, gateways or other links to the content data and applications, access to the system name server and other directories, and so forth (equivalent to an RPC toolkit). While a sophisticated end-user can build a workflow once the system is set up, it generally takes a programmer to install the system in the first place and to use STFs to provide connectivity to legacy data and applications already in place. For resellers, Action provides STFs for most standard tools and systems; VARs can also do their own STFs for less cookie-cutter environments. (Note also that Action's Workflow Language is independent of any particular tool; it can be generated by Action's tools, or by those of a third party. But for now only Action's engine executes it.)

Action has fundamentally reshaped its business strategy over the last few years, shifting from a product company to an OEM supplier of technology. Overall, Action should generate revenues of about $6 million this year, mostly from royalties for its technology and related implementation consulting. Its most lucrative and most visible client is Lotus.
LOTUS WORKFLOW

Lotus has fundamentally reshaped its workflow business strategy over the past few months, finally deciding how it will work with third parties. An interesting test of Lotus’ new-found openness will be its reaction to BeyondMail for Notes, scheduled to be announced at NetWorld shortly.\(^2\)

Lotus has now decided to work closely with Action Technologies, and to sell the Action design tool (and runtime system with database APIs) as the foundation of its industrial-strength workflow offering, which will probably be called Lotus Workflow Processor. Note that this is Lotus Workflow, not Notes Workflow. In its initial form, it will keep its own (transaction and definitions) data in Notes, and use the Notes and cc:Mail cross-platform mail and data management facilities by default. Later on, it might be embedded in other Lotus products or operate stand-alone.

--------------
Customers know only that they can’t keep up with the outside world, with their customer requests, with internal information flow. But that’s not a problem; that’s malaise. That’s why groupware needs VARs -- not to provide solutions, but to define the problems before delivering the solutions.

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

Lotus is in an odd position offering workflow (as we noted in our August issue). It doesn’t inherently have the primary technical capabilities, but it’s in a stronger position to sell them than almost anyone else. Indeed, much of the issue with groupware is marketing and implementation; the technology is actually quite simple. Moreover, Lotus does have a strong market position in "groupware," and an installed base of networked-together users, some of whom need to manage workflow as well as share information. Some Notes users may be already trying to manage workflow, and will be glad of better tools for doing so.

Lotus Workflow comprises the basic Action development tools. The execution system consists of three modules -- what Lotus/Action calls definitions, or workflow templates; workflow instance data (or what we would call the workflow or transactions database); and interfaces to the work content, i.e. the actual data the users interact with as they perform the steps in a workflow instance. The first two are kept in Notes, while the work content could be anything reachable through Windows (locally) or over a network with the proper STFs, APIs and connectivity. Lotus Workflow includes interfaces to SQL and a variety of standard data formats; it can also easily handle DOS files as whole objects without being able to look inside them.

Transaction integrity is defined and guaranteed by the workflow engine, on an OS/2 server (UNIX later on), communicating with the process information held in Notes and with the users through Notes, and with the work content wherever it may be.

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

2 One reason why this issue of the newsletter shipped so late was in order not to preannounce this product. (Another must be the lack of effective workflow in our office!)

Release 1.0

30 September 1992
By default, from the user's perspective, most things happen from inside Notes: Data is presented, workflows are designed (or parameters specified), messages are sent and received via NotesMail. Of course, using STFs and APIs, Lotus Workflow can manage work in a multitude of other environments and interact with users through them. Long run, if Lotus Workflow is successful, it won't be regarded as Notes-only.

Nonetheless, there are already enough workflow tools out there that deal with images or database forms; Notes and Lotus Workflow are uniquely suited to handle the stuff of many of the more interesting workflows: documents, resumes, and data that can't be stored rigidly in a relational database but that has more inherent structure than your typical set of e-mail folders. (See also BOSS Logic, page 18.) When the work content data is kept in Notes, you get the benefit of replication, so that a disconnected user can take his work with him (although the obverse is that he may take along a lot of extra data). The alternative is to have to hook up remotely to some database. Each user will make his own assessment of the trade-offs.

Third party: Quality Decision Management

Quality Decision Management is basically the implementation team behind several pilot implementations of the ATI/Lotus workflow technology. Working variously for Lotus, for ATI or directly for the client, QDM has implemented pilot systems for Arthur Andersen (the CPA firm), Young & Rubicam, a workflow-enabled help desk for the General Services Administration and a 20-loop workflow for potential project assessment for the US Air Force.

These are also pilot implementations of version 2 of QDM's own Quality At Work Notes-based system, which will be announced at NetWorld. It uses the ATI technology and will benefit from the features which will be included in the final version of Lotus Workflow -- notably the STFs for linkage to non-Notes environments so that data from outside Notes is easily accessible.

QDM's basic business is premised on the notion that quality is a function of teamwork and synchronization -- and that therefore the technology for managing it should extend to each member of the team, enterprisewide. The QDM application suite is a set of Notes-based modules for such tasks as strategy planning (and subsequent follow-through), sales & marketing, project management, customer service and feedback, and follow-up. The Action technology has enabled QDM to make its applications more active -- managing activities and coordination of individuals as well as dissemination of information.

Third party: ELF Technologies

Another user of Lotus Notes and probable adopter of the ATI technology is ELF, a groupware-development firm that specializes in Notes-based systems for legal and insurance customers. ELF provides accessibility (cuteness) around the basic business of building agents and workflows. It calls its software agents elves (for Electronic Labor Force), "enchants" processes (by encapsulating and defining them, so they can be called from scripts or workflows), and so forth. The company has been very successful building Notes-based information-dissemination and routing systems on top of Notes; the Action technology will help it to build in more robust workflows. Customers include the Bullivant, Houser, Bailer, Pendergrass & Hoffman law firm in Portland, OR, and American International Group, one of the country's more far-sighted insurance companies.

Release 1.0

30 September 1992
In addition, ELF will be working directly with Lotus, building agents (or elves) to manage Lotus products including Notes, The Organizer and Ami Pro. It also has elves for a variety of other real-world systems such as Elite (an RS/6000 legal billing system), WordPerfect (including its all-important mail-merge capabilities), and Polaris PackRat.

The elves don't just manage workflow; they do the actual work.

OTHER ACTION THIRD PARTIES

Action also works directly with other software OEMS. They include IBM, which won't reveal details, and also:

Da Vinci Systems is selling Action's Coordinator as a distinct product (under an exclusive license). In its own current DOS eMAIL product it implements a rudimentary form of conversation management simply by linking messages that reply to one another. Its new Windows eMAIL, to be shipped for NetWorld, will have "intelligent routing" (multi-step forwarding without conditionals or branching) and forms. By year-end Da Vinci will ship a Windows version of the Coordinator which will include forms and predefined conversation templates (request, inform, question, what if and offer) with status monitors (so that a conversation has a defined type and state; for example, Juan made a request of Alice and she now owes him an answer).

Sometime in mid-1993 it will use actual technology (as opposed to ideas) from Action, to build structured workflow templates for tasks such as purchase orders, travel requests, document reviews, inventory requests, phone messages and time sheets. Those will be incorporated with the next release of Da Vinci’s mainstream eMAIL product line, along with the Action workflow engine underneath.

Eventually, Da Vinci may resell the Action workflow engine tools, which would allow users (or VARs) to design their own e-mail-based workflows with flexibility and process integrity.

Verimation is a Swedish company (descended from Volvo) that has successfully competed with IBM's PROFS in the mainframe-based e-mail business, with 1.3 million users of its Memo at 850 mainframe worldwide and 250,000 in the US. Now Verimation has right-shaped Memo with pc (Windows, Mac, DOS and OS/2) and LAN clients. It has also incorporated Action technology into its own OS/2-based workflow engine server, which treats users, data and applications on the mainframes just like any other clients. This is a fine example of the use of groupware tools to incrementally re-engineer legacy applications.

LaserData, a vendor of pc-based image management systems, will incorporate the Action Workflow Management System into its next release, primarily for use by VARs as they build and customize systems for their customers.

Delrina is talking to Action about licensing its technology in order to add some predefined workflows to its forms tools and routing systems. Assuming an agreement is reached, it plans to announce the details not at NetWorld but at the Electronic Messaging Association meeting later in October.
WORKFLOW: CONTROL IS KEY

Aside from process integrity, a second issue in workflow is where the control lies -- both in design and in operation.

In design, the issue is simple: Who gets to build the rules (wherever they may be executed)? Who gets to change them?

In execution, where is the state information stored and managed? Who can change the workflow, and when? As we've noted before, there are three (at least) possibilities for the location of execution/modification control:

- at the server (typically a database application),
- with the user (typically a set of scripts or rules for responding to e-mail), or
- with the work object or agents (typically moved by e-mail).

Ultimately, despite talk about empowering users and the like, if you want process integrity you need to control execution of a workflow from a single logical point (a database, fundamentally, although it could be physically distributed). If you want process integrity within a system of interacting workflows, you need to design them at a single point. The trick to good, politically correct workflow design, then, is to build robust workflow templates that allow parameterization but not structural modification locally. This gives maximum power and flexibility to individuals, but maintains integrity overall. (Implementation of such systems is an exercise left to the reader.)

The central database approach

The advantage of the central, database approach is that it makes it easy for an administrator or for a user with authority to track what's going on. There's a central database that monitors everything and manages the status of each workflow instance. It either holds the application data or maintains pointers to it (that's an irrelevant detail from this perspective). The database does what a database should do: It keeps track of things.

Such a workflow typically executes like a long-transaction database system. Whereas in most databases the data sits quiescent and is changed in split-second transactions when something occurs, a workflow database operates essentially in inverse: Most of the time the system is in a state of suspense, of mid-transaction.

Take a typical workflow, initiated by Juan: He starts things off by creating a customer file for Johnny Start-Up, who wants a loan to launch his new Windows spreadsheet. That transaction triggers another one, which is for Alice to assess the quality of Johnny's business plan. Alice is busy, of course, so the workflow goes into a suspended transaction, waiting for Alice's response. For each workflow, the system spends far more time waiting for the human than vice versa.

Alice may use the workflow tool -- if it's easy enough -- to delegate some work to a subordinate, or she may simply send him an e-mail message with an
attachment, bypassing the system (and monitoring the integrity of the process by herself). Alice's ultimate decision -- no way! -- triggers a final step, the automatic generation of a rejection letter with a copy to Juan, which closes the transaction.

The workflow database keeps a real-time record of everything that's happening, and can provide reports or answers to a variety of queries. It also notices when something fails to happen, or a process fails to complete.

Once the work moves beyond the reach of the database, the database loses control. The workflow engine can only wait patiently for a response to come back -- or perhaps fire off messages triggered by a deadline that is missed. (You can work with distributed databases, of course, but that's not quite as easy as vendors would have you believe.)

In this formulation, the control basically lies with the developers, who built in the process integrity. Alice responds to their (programmed) sequence of events, kicked off by Juan.

The user agent

In its basic form, this approach allows an individual user to automate his interactions with other people or applications, to trigger future actions, etc. The user-centered approach lets the user automate his own responses to events and messages, but he can't really create a real workflow; in each cycle, the message reverts to his machine and his next programmed step. Of course, conceptually, he's playing the role of the database above, monitoring each step of the workflow.

The system of workflows consists of interacting user agents -- who may not be acting consistently. But call that single user a representative of the group (or an administrator) and put his "user agent" on a server, and you can actually have a centralized approach with this technology.

This is basically what many BeyondMail users are doing, placing house rules on a server, interacting with individual users' rules at their workstations. One of Beyond's customers is extending that approach by having BeyondMail send transaction records into an SQL database -- which then monitors the workflows at the database.

However, you still have the problem of writing the rules/processes/steps carefully, and of monitoring execution of the workflows. As we said, underneath it comes down to the same ideas. It's only the tools and their affinity for the models that changes from approach to approach.

The object/e-mail approach

In the centralized database models above, electronic mail can be used as an auxiliary function -- simply a way of getting messages outside the workflow engine's natural territory. Another approach uses electronic mail in a more fundamental way (although you could build a similar system using other transport mechanisms). Basically, the work carries its instructions with it. Each work packet (or object) has a list of people/sites/roles it should go to, in order (possibly with conditional branches), possibly derived from a workflow template/class/type but with specific instance data.

Release 1.0	 30 September 1992
Early practitioners of this form of workflow include Interleaf, with its Active Documents, and NewWave, with its agents. A number of e-mail tools and form tools take this approach too.

However, you still have to build an application (workflow engine) to guarantee process integrity if it's required. For example, Reach Software's e-mail-based WorkMAN actually is database-based if you look hard enough. It encapsulates workflow data into a typed object which travels as e-mail. When it arrives at a desktop, it maps into the local piece of a distributed type manager which contains the instruction to present the data as a form, execute or load the relevant applications, and execute routing rules based on local data and the object. Simultaneously, it sends back status messages to a central database. In essence, Reach creates a centralized tracking database, using e-mail rather than direct database access. The e-mail is a transport mechanism that enables it to expand its -- pardon us -- geographical reach.

**Contrast and compare**

The problem with the second two approaches is that they don't scale up gracefully (in volume as opposed to geography). A few local/individual workflows are fine, but what happens when they're running all over the place? Like uncoordinated people, they can be hard to keep track of.

Of course, you can jerry-build control by having an e-mail package report every message delivery to some central point (as Reach does), but that approach gets needlessly burdensome on a large scale. You still need something at the central point to manage all that information and what it means, an engine to monitor and react to the incoming tracking data as well as to store it. Conceptually, we like the ideas of user freedom, self-organizing systems and the like -- but they don't guarantee process integrity!

**Control is in the hands of the owner**

Note the difference between execution control and original development. You could have a dp department that writes user agents for use by individuals, thus enforcing individual behavior. You could likewise have individuals write their own workflow applications for execution by a database server. The object/mail model is in-between: typically the applications or templates are written centrally, but individual instances, with specific data and routing lists, could be launched and modified by individuals.

Of course, you might assume that in the seamless networked world of tomorrow that doesn't really matter, but of course it does. Who controls the flow of work? Who controls the individual people it goes to? Who sets the deadlines and has the ability to follow up or inquire on progress? Who can change a workflow in real time, and who can change the standard workflows for his office? Does the user feel controlled, or does he feel empowered? Can he see where he is in the flow -- at the bottom, or on the top?
BEYONDMAIL AND NOTES

At NetWorld, Beyond will put the openness of Lotus Notes to the test.

Beyond has moved from being an upstart e-mail company with a clever idea to a serious industry contender (see Release 1.0, 11-90). Its original idea was simple -- and borrowed, with attribution, from Tom Malone/MIT's Information Lens. BeyondMail lets individual users build rules to handle their e-mail, performing a variety of actions depending on the values in the various fields of an e-mail message. The nice thing about Beyond is that you can program it by example: Take an incoming message as a model for "if" conditions to be met, and then specify how you want messages with selected characteristics from that model handled. The "then" part works with a pick-list of possible actions (verbs) and nouns (folders and other users, mostly). In the new Windows version, you can also launch almost any activity or use any data reachable through OLE.

Beyond that, you can build defined types of messages (basically, forms without a database behind them), and take actions based on the values of specific fields in those forms or the presence of specified strings in text fields. In essence, it's an easy way for a user to build a workflow script, with the proviso that all the rules execute only from his workstation.

Of course, that's easily broadened (as noted above); you simply create a virtual user on the server, which can now manage workflows for all messages that pass through the server -- not much of a technical stretch, but conceptually, a whole new world. In essence, that's the beginning of workflow, albeit without process integrity. Imagine what you could do with Beyond as your programming tool, and the Action workflow engine and STFs...

BeyondNotes?

As we said, that's workflow. Separately, Beyond will announce at NetWorld a tool to add automated filtering -- not workflow -- to Notes.

Beyond will announce BeyondMail for Notes. This product creates a code module that sits on a Notes server (looking like a client to Notes), filtering items of interest and sending them to its own clients, who are BeyondMail users (and only indirect beneficiaries of the Notes server). Thus the system fully complements Notes servers, but it potentially replaces Notes clients. Or as Beyond sees it, it allows users the full benefits of BeyondMail, and also allows them access to Notes information on a filtered basis.

The way it works is that the BeyondMail user fills in a Notes profile form (implemented in Beyond) indicating the topics, authors, key words and other criteria of interest to him. That form automatically generates the appropriate Notes code and APIs to filter Notes databases. The resulting module sits on a Notes server and forwards the selected information to the BeyondMail user. The BeyondMail user can also send comments back to Notes, which are handled appropriately.

Altogether, Beyond has accomplished the extremely subtle task of creating a new Notes front-end that is not a rip-off of the Notes front-end, but rather presents equivalent information effectively in the BeyondMail environment. Certainly the flexibility and ad hocness of Notes is missing, as is the

Release 1.0	 30 September 1992
ability to rummage through the system and follow links, move up and down hierarchies, and create Notes applications. BeyondMail for Notes is actually a scripting tool for automated filtering of Notes databases. If you want interactive access, buy a Notes client.

Beyond or behind?

Now the question is, How many people want which? For high-end folks, Notes itself is valuable. But for people whose managers simply want them in the loop (though only in certain loops), the BeyondMail approach makes a lot of sense. It also makes sense for people who don't really want the extra work of maintaining a Notes server. You can get much of the benefit without the resource cost of replicating a Notes database in a variety of locations.

Of course, customers will eventually need extra Notes servers to support the additional BeyondMail clients. The issue is whether the extra servers make up for the lost clients: If you believe that BeyondNotes will lure more users into Notes (even indirectly), and faster, than would happen otherwise, then you have to believe that this is positive for Lotus Notes. Certainly we think so; it helps establish Notes as a standard. Right now that, rather than revenues, should be Lotus's goal. (This also argues that maybe it's time for Lotus to change the Notes pricing structure, with a higher price for the server and perhaps a lower one for the client.)

Note, of course, that BeyondMail for Notes is strictly a derivative product; it depends on the existence of the Notes server (and the Notes APIs and scripting languages) to do its work. That is what standards foster: derivative products. BeyondMail for Notes certifies Lotus's success -- and Beyond's alertness to possibilities. We only hope Lotus can recognize homage when it sees it.

* * * *

WORKFLOW APPLICATIONS: CIMLINIC

Cimlinic, a workflow toolkit for manufacturing environments, is an excellent example of a workflow system, since it makes certain concepts clearer than an equivalent system in an office. The computer really has no more idea of what Fred is doing with his spreadsheet than of what Rosie is doing with her rivets. It may have a production plan for the riveter, and it may "know" that it should load Excel and a certain file for Fred, but it doesn't know what Fred is doing. A programmer may write a routine to look at a certain cell or a named range on Fred's spreadsheet, just as a routine may check a counter on Rosie's riveting machine, and make a decision on that basis.

The problems of office groupware are no different. A spreadsheet is almost as foreign to a database as a machine tool. It needs to communicate with it through a protocol; it knows something has been done only when it is told.

The problem with workflow groupware is that so much of what it normally has to deal with is undefined: The transactions between people are complex and
subtle. It turns out that the same problem applies in the manufacturing environment: Although the processes involved are tangible, most of them haven't been defined to a computer that can understand them in any global way. That is, even in an "automated" factory, each part of the assembly line or process queue is independent. Some stations may pull down inventory information from the same mainframe, while other stations may use a graphical workstation with CAD diagrams and engineering instructions. But just as physical goods pass directly from one location to another, bypassing the information system, so does information generally pass directly from one location to another, in the form of engineering change orders, special requests, etc. (There may be some information passed to and from the mainframe periodically, such as number of items received or assembled, or configurations of larger items might be noted -- although typically that's kept on a form taped on to the item in question.)

"We rashly assumed that just because people automated manufacturing activities, they had automated manufacturing," says president John West of Cimlinc, a maker of manufacturing groupware. "But in fact, they had just automated the individual steps." The process was still as ad hoc, paper-ridden, and unorganized as ever. Each manufacturing station was linked to a mainframe, perhaps, or to a CAD system, but they were not linked to each other. There was no way to manage the flow of work from station to station. There were no links in the mainframe, and no links on the floor -- except for paper forms, memos, meetings, phone calls, and hurried consultations as engineers tried to fix things, change production amounts, etc.

In fact, manufacturing is never routine. Machines go down, part specifications and formulas change, special orders come in, people go on vacation, new products or processes are developed.

Cimlinc got into the groupware business a while ago, with an image-routing system. Thus there were three orthogonal tracks: Computer information, which went up and down between people and various information sources; physical goods, which went through the manufacturing/assembly process according to a mostly pre-determined path, perhaps with detours for rework, quality testing or other exceptions; and images, which provided an automated way to route information which had frequently floated around on slips of paper: requisition forms, engineering diagrams, engineering change orders and the like.3

You could query a given system, and get some kind of information (such as planned production), but there was no single information system that could reliably integrate and report what was going on. The images were not "alive." Usable information was divided among brittle mainframe or mini systems, people actually on the line, and pieces of paper or images floating among them. There was no reliable way of either sharing information or sending comprehensible, live messages from one place to another.

3 To this list, Action Technologies would add business transactions, while combining computer information and images into the single category of information.
Fast-moving but static

The image system solved the problem of losing pieces of paper, and got them from one place to another faster, but they were inert. A person had to look at the image, do something about it, and then generate some information of another form. Or perhaps he merely did something physical, such as turning a knob or changing a paint color, without informing the system that he had done so. Now a person querying the mainframe would have no way of knowing that the next batch would be green.

Cimlinc basically has everything except a workflow engine; it’s the environment within which a workflow engine can operate. (Originally it did have its own Concurrent Engineering Control System, which incorporated a database and some workflow templates built with the Cimlinc scripting tools. However, Cimlinc’s direct customers, who are mostly systems integrators within manufacturing companies or third parties, preferred to use their own database or workflow engines, and use Cimlinc’s tools to hook into the manufacturing environments.)

Thus Cimlinc’s contribution is basically in integration tools that enable the builder-user to hook the workflow into the legacy manufacturing systems that actually run the end-users’ businesses. Cimlinc’s value-added toolkit, like ELF’s internal tools, let builder-users link end-users’ information infrastructure into the workflow control structure. Both enable users to operate within a familiar world, with new tools controlling things discreetly from behind (or for the back-end, so to speak.)

PLEXUS -- SUPPLIER TO...

Plexus aims to become a basic workflow tool vendor, supplying technology to companies such as NCR as well as directly to its own VARs, who focus on imaging systems. The company began in the early 1980s as one of the many UNIX box start-ups. Within a few years it realized the future was brighter in software and developed an image-management system, XDP, but that wasn’t enough. In 1989, Recognition Equipment of Dallas bought the software assets and hired the people. Plexus released its UNIX workflow product, FloWare, last June; among its 20-plus customers so far are Du Pont, Fidelity and American Express. Its prize OEM is NCR.

Where ELF has elves, FloWare has "couriers" who take the work around.

...NCR for ProcessIT

NCR, which once touted workflow as part of its Cooperation end-user environment, has now settled on a very different approach with a robust, more MIS-oriented system called ProcessIT, scheduled to ship by year-end.

The original Cooperation workflow was NCR’s own, a sort of graphical tool for end-users which showed roles and activities but never got as rigorous as flowcharting; it was an extension of NewWave’s agent capability. The next
iteration was the workflow in AT&T’s Rhapsody, which was itself a rework of Dublin-based Workhorse’s Workhorse product. Workhorse made the mistake of being something of a closed system; it had its own word-processor and database, so you had to use its tools to do the work as well as to manage the work. This (putatively final) version is based on the Plexus/REI workflow technology. It will cost about $1000 to $1200 per client in systems of 60 to 100 users, including server software. (That gives it an initial cost about twice that of Notes, for what it’s worth.)

ProcessIT is as open as you can be. Basically, it’s a set of APIs and the Process Activity Manager (PAM), Plexus’s workflow engine based on a database (Informix, but kept well hidden) that manages and records the workflow transactions. NCR includes a number of Plexus’s graphical development and analysis tools, and has implemented them in a Windows environment. To FloWare, NCR has also added roles⁴ (so that you can route to types of people rather than an individual), folders (to contain work data and applications or pointers to them), priorities, a friendly user interface (WorkView), and a set of administration tools (Workflow Admin).

In client-server terms, ProcessIT is a server application/tool. The workflows are also executed at the server, while the actual work is performed by user clients (which may in turn call on other servers for data or for processing). NCR provides heterogeneous communications capabilities to the TCP/IP-based FloWare to hook up almost anyone to it, through a combination of online access, server-to-server communication and messaging. It does not require the user to be within the Cooperation environment, although clients must be in Windows to take advantage of the user-friendliness and to use pc applications.

You can use a variety of tools, from Borland’s ObjectVision to Oracle Card, the REI/Plexus 4GL, the Progress 4GL, Visual Basic, Smalltalk/V for Windows, Micro Focus COBOL/2 or a variety of C and C++ compilers, to manage within communication with the actual work (to do the work, or to find data that might govern the progress of a workflow), while the workflow itself is created with the FloWare/ProcessIT MapBuilder. It and a set of other graphical tools govern the assignment of roles, the routing of work, including branching and conditionals, and a variety of monitoring and security functions.

The work is done by users, using applications and files or data sets retrieved from outside the system; that is, FloWare/ProcessIT assembles the proper information and tools for the user (or another program) to do the work. It can as easily hand work to a program as to a person. In either case, the person or program must somehow report the work as done to the FloWare/ProcessIT engine, which then takes over and hands the work off to the next step. The user remains in the ProcessIT environment, which appears to him as a task within Windows.

In the design phase, ProcessIT can generate flowcharts around programs built with any of the APIs or tools it works with. You can move icons around on-

---

⁴ A person may have several roles; an activity may require work performed by several roles. Thus Juan may be able to perform certain jobs but not others; Mary can also perform some of those jobs, and others as well.
screen to change a map. It manages FloWare's Process Maps, graphical representations of the workflow. You can also have submaps, which can be reused as is or modified in other Process Maps. In the same way, multiple process maps could use the same work activities; for example, both mortgage and consumer-loan workflows might involve the same credit-check step.

While it probably takes an expert to build the workflow applications, the Workflow Admin tools allow users to assign or change roles, check on the progress of a workflow, look for bottlenecks and otherwise manage the flow of work (within security/authority constraints). FloWare/ProcessIT also includes some useful debugging/design tools: Trailer (for record-keeping) and Exerciser (for simulating workflows).

WorkView

In the Cooperation version, PAM also has its own client application, WorkView, a Windows application which is basically a window into the system so that workers and managers can see what's going on. Built with the FloWare APIs in C++, WorkView is the shell that a user uses to log in and out of the system. For individual users, WorkView displays pending tasks as a to-do list after you've defined your role and priorities and manages the process of loading Windows applications that are not workflow-aware, fetching related files and so forth. With the proper authority, you can use WorkView to change the contents of folders -- either for a workflow template, or in a single case.

WorkView uses Windows DLLs for its communication with the user's applications, and will use OLE 2.0 and DDE. (Otherwise, FloWare/ProcessIT talks directly to other applications, but that requires the use of APIs and other programming to integrate the applications; cf. ATI's STFs.) It also manages the process, and sends the work back into the queue after a given time if the user doesn't do his job. The workflow may then remind the user, send it another user with the same role, or send a notification to a manager, depending on what it is programmed to do.

BOSS Logic

We faced an embarrassing situation vis-a-vis Boss Logic. We had been planning to write about the company for some time, and so instead of filing the information, we kept it in one of the piles near our desk. Well, of course, those piles keep proliferating, and we never could find the information.

We ended up calling Boss Logic to ask them to send over a new set. On their end, of course, they keep all their documentation in Boss Logic, neatly organized by kind of document (spec sheet, price list, white paper, cover letter template for customer or press, follow-up letter and so forth).

Boss Logic offers a combination of low-end workflow and high-end information structuring, currently all based on Sybase's SQL Server and in the NeXT environment (with Windows early next year). Where Boss Logic differs from other more-or-less traditional document-management and image-flow systems is its support for compound documents. It is built in a clean, clear client-server architecture that makes it easy to understand, modify and extend (and
will make it easy to move off its current NeXT platform). The current version is written for Sybase but could easily use any other SQL database.

It uses a relational database as the storage mechanism for a server application that manages content and workflow for document/image-intensive tasks. The Boss Logic information-management capabilities are fundamentally compound document management, including version control, revision control (who may make changes), document structuring and assembly, and management of auxiliary information, such as links to clients, projects, etc. The client side is built with standard C++ tools and NeXT features; the back-end with Sybase tools.

Each document is represented as a (usually) hierarchical set of components -- chapters and diagrams, sections, captions and pictures within chapters, say, or sections of documentation relating to the different components of a product. They could also be the parts of an annual report, including formatted spreadsheets, artfully presented sales charts and flattering photos of the chairman surrounded by adoring employees. Need to revise the sales figures? No problem. Boss Logic reminds you that they're linked to the sales charts, so you'd better use the revised version of those too. Fire a board member? Get rid of his picture, quick!

The second part of Boss Logic manages workflow (also in SQL Server) -- in the scripted rather than process-integrity sense. Users can use forms to design workflows with routes, due dates, and different views for different users. You can query it about the status of a job, all the projects related to a certain client, etc. You can define an approval cycle, say, or a trip to the art department for creation of appropriate decorations, etc. For example, Bozell & Jacobs (now formally BJK&E) in Dallas works for American Airlines -- doing just-in-time ad development in response to the fast-paced airline industry. It is installing BOSS Logic to manage a semi-automated approval cycle to get ads out fast.

Fundamentally, Boss Logic sells a highly customizable workflow application, using some of the powerful NextStep tools to build logic and a pleasing interface, along with the Sybase engine. Currently it is focusing on environments with structure-intensive compound documents, such as advertising agencies, technical and reference publishers, and people like investment banks publishing long, defined-format documents. Interestingly, it received funding of $1.5 million earlier this year from Frame Technology, which also recently acquired Datalogics, a specialist in SGML and document assembly that sells to government outfits (who are big on lengthy documentation and repetitive filings).

If we can automate everything, why bother with people? That's a valid question, but it's one we won't face in most places for some time. People will keep doing value-added and design work: painting or assembling things machines can't; making decisions or judgments machines can't; interpreting documents machines can't. There will be more of this kind of work as more and more of the automatable work is relegated to machines.

Release 1.0

30 September 1992
UNFOUNDED RUMORS

We reprint a news item from "Computer Publicity News" in order to refute it.

"Rumor has it that Esther Dyson would like to sell RElease 1.0 [sic], her authoritative monthly newsletter, due to other projects and activities that require her greater attention. She has become a limited partner in Mayfield Software Partners, which is investigating the future of operating systems and other directions in computing..."

Well, all we can say is that one reason we're "authoritative" is that we bother to check out rumors rather than just print them. Had the editor had the courtesy and minimal professionalism to call me, I would have told him this: Rather than sell the newsletter, which would require me to work for some new owner (I almost did that back in 1985!), I'm in fact about to hire someone to work with me. He will help in writing Release 1.0 and will implement some more technology in our day-to-day operations. (Perhaps the search for that person led to the rumor.) We'll announce the details next month...

Separately, Mayfield Software Partners is investing in software companies, not exploring new technology. (Maybe he mixed it up with Interval Research?) As a limited partner, I contribute money and random advice rather than time. This is not a job but an investment--one which allows/invites me to attend one West-Coast meeting every two months and to forward promising business plans to the general partners, who actually do the work.

What I am spending more time on lately is a number of related activities: the East-West High-Tech Information Service, a new business devoted to discovering and fostering start-ups in Eastern Europe; and membership on the boards of the Electronic Frontier Foundation and the Santa Fe Institute. Like all my activities, they both enrich and extend what I do for Release 1.0--even as they take up extra time.

I hope this clarifies matters. I remain deeply committed to Release 1.0 and its mission of exploring new technology and its implications, even as I have broadened my life with additional related interests. If you have any questions about the future of the newsletter or of EDventure Holdings, please call me. I'm here at (212) 758-3434!

-- Esther Dyson
RESOURCES & PHONE NUMBERS

Tom White, Action Technologies, (510) 521-6190; fax, (510) 769-0596
Chuck Digate, Beyond Inc., (617) 621-0095; fax, (617) 621-0096
Peter Rip, BOSS Logic, (415) 903-7000; fax, (415) 903-7009
John West, Jack Thornton, Cimlinc, (708) 250-0090 x1661; (708) 250-8513
Don Rainey, Nelson Pratt, Da Vinci, (919) 881-4320; fax, (919) 787-3550
Burt Amato, Mark Skapinker, Delrina, (416) 441-3676
Jon Ramer, ELF Technologies, (206) 232-7808; fax, (206) 236-1586
John O'Connell, FCMC/Staffware, 44 (71) 262-1021; fax, 44 (71) 262-3956
Cindy Wild, FCMC, (617) 239-8221; fax, (617) 239-8223
Langley Steinert, JetForm, (508) 840-8181
Rodney Rogers, LaserData, (508) 649-4600
Brownell Chalstrom, George Gilbert, Cindy Schuyler, Lotus Development, (617) 577-8500, 693-4351; fax, (617) 693-1149 or 693-4663
Joerg Rothfelder, NCR, (803) 739-7736; fax, (803) 739-7588
Dean Cruse, Plexus, (214) 579-6000; fax, 579-6400
Andrew Jeffrey, Quality Decision Management, (508) 688-8266; fax, (508) 688-5181
Anand Jagannathan, Reach Software, (408) 733-8685; fax, (408) 733-0482
Bruno Giversen, Verimation, 46 (31) 355-500; fax, 46 (31) 355-565
Adam Sroczyynski, Verimation, (201) 767-4795; fax, (201) 767-4885

COMING SOON

- Simulation for education.
- Pen stuff.
- End-user programming.
- Constraint-based reasoning.
- A map of the wireless world.
- Borland's groupware strategy.
- And much more... (If you know of any good examples of the categories listed above, please let us know.)

Release 1.0 is published 12 times a year by EDventure Holdings, 375 Park Ave., New York, NY 10152; (212) 758-3434. It covers pcs, software, CASE, groupware, text management, connectivity, artificial intelligence, intellectual property law. A companion publication, Rel-EAST, covers emerging technology markets in Central Europe and the Soviet units. Editor: Esther Dyson; publisher: Daphne Kis; circulation & fulfillment manager: Robyn Sturm; executive secretary: Denise DuBois; editorial & marketing communications consultant: William M. Kutik. Copyright 1992, 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 $495 per year, $575 overseas.

Release 1.0
30 September 1992
## RELEASE 1.0 CALENDAR

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>October 18-22</td>
<td>OOPSLA '92 - Vancouver. Sponsored by ACM. Call Rebecca Wirff-Bock, (503) 242-0725</td>
<td>John Richards, (914) 784-7731.</td>
</tr>
<tr>
<td>October 20</td>
<td>New York PC user group meeting - New York City. Sponsored by NYPC. With Andy Grove.</td>
<td>Call Jim McMullen, (914) 245-2734.</td>
</tr>
<tr>
<td>October 27-29</td>
<td>PC EXPO - Chicago. Sponsored by Bruno Blenheim. Call Annie Scully, (201) 346-1400</td>
<td>or (800) 829-3976.</td>
</tr>
<tr>
<td>November 1-4</td>
<td>*Computer-Supported Cooperative Work '92 - Toronto. That is, groupware... Sponsor: ACM.</td>
<td>Call Ron Baecker or Marilyn Mantei, (416) 978-5184.</td>
</tr>
<tr>
<td>November 9-12</td>
<td>Software maintenance '92 - Orlando. Sponsored by ACM. Call Vaclav Rajlich, (313) 577-5423.</td>
<td></td>
</tr>
<tr>
<td>November 16-20</td>
<td>Comdex - Las Vegas. The biggest US show of all. With panel on information-sharing groupware with Chuck Digate (Beyond), Cyril Brookes (grapeVINE) and Jeff Conklin (CM/1) moderated by Esther Dyson. Contact: Peter Young at Interface Group, (617) 449-6600; fax, (617) 449-6953.</td>
<td></td>
</tr>
</tbody>
</table>

Release 1.0 30 September 1992
**Contact:** Mark Weber, (708) 260-9700; fax, (708) 260-0395 or Marjolein Jacobs, 31 (30) 955662; fax, 31 (30) 955539.

**November 30–December 4**

**Hypertext '92** - Milan, Italy. Sponsored by ACM. Contact: Julie Eitzer, (212) 626-0604; Paolo Paolini, 39 (2) 2399-3520; e-mail, paolini@ipmell.polimi.it.

**December 8–11**


---

**1993**

**January 25–29**

**PenExpo** - San Francisco. Co-sponsored by Boston University Corporate Education Center and PenWorld. Call Tom Zaralides, (508) 649-6926 or (800) 733-3593.

**January 31–February 3**


**February 21–24**

**EDventure Holdings PC (Platforms for Computing) Forum** - Phoenix. "Content is key." You read the newsletter; come meet the players and try the tools described in this issue. The usual suspects, plus Tom Ray, Jeff Hawkins, Vern Raburn... Call Daphne Kis, (212) 758-3434.

**February 21–26**


**March 1–5**


**March 3–7**


**March 9–12**

**Third Computers, Freedom and Privacy conference** - San Francisco. Social responsibility can be fun! Sponsored by ACM. Call Bruce Koball, (510) 845-1350.

**March 15–19**


**March 24–31**

**CeBIT '93** - Hannover, Germany. Sponsor: Hannover Fairs USA. Everything you're interested in, and a lot more. Call Mette Fisker Peterson, (609) 987-1202; Geide Schlichting, 49 (511) 89-0; fax, 49 (511) 89-3 2626.

**March 28–31**


**March 31–April 3**

**Seybold Boston** - Boston. Sponsored by Seybold Seminars. Call Beth Sadler or Kevin Howard, (310) 457-5850.

**April 19–23**


**April 24–29**


**April 25–28**


**July 11–16**

**AAAI/IAAI 93** - Washington, DC. Sponsored by The American Association for Artificial Intelligence. Call Carol Hamilton, (415) 328-3123.

**October 25–29**


---

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

*Release 1.0* 30 September 1992
**Subscription Form**

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 rates available on request. Satisfaction guaranteed or your money back.

<table>
<thead>
<tr>
<th>Name</th>
<th>Title</th>
<th>Company</th>
<th>Address</th>
<th>City</th>
<th>State</th>
<th>Zip</th>
<th>Telephone</th>
</tr>
</thead>
</table>

- [ ] Check enclosed.
- [ ] Charge my
  - [ ] American Express  
  - [ ] MasterCard  
  - [ ] Visa

Card Number ___________ Expiration Date ___________

Name on Card ___________ Signature ___________

- [ ] Please send me 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, NEW YORK 10152

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

Daphne Kis
Publisher