ANATOMY OF GROUPWARE

We recently attended GroupWare '92, the commercial kick-off of the field. Miraculously, it had more buyers than sellers -- something many exotic new fields never achieve. Maybe the reason is that groupware is ultimately a set of tools for practical tasks -- using a variety of hardware and software technology to inform and coordinate people -- rather than a technology looking for applications. (Applications of, say, expert systems do exist, but they're properly being sold at vertical market trade shows rather than AI conferences.) The basic goal of groupware is tools normal people can use to assist or automate business processes they understand.

In the end, groupware uses fairly simple technology, but it requires people to think about their business activities in new ways. As with expert systems (a programming approach used in some groupware, by the way), builder/users must explicitly represent their information and processes in order to program them. Once groupware tools are good enough, the obstacle to adoption and implementation will be the user's business understanding -- or the consultant's. The adoption of groupware provides the opportunity for people to re-engineer their businesses, but it won't do the thinking for them.

Finally, modularity in all things: Groupware tools should be integratable with each other and with legacy applications, but not necessarily integrated. Previously "groupware" was either embedded deep and inexplicitly inside mainframe or mini applications or, more likely, in a set of business processes represented in procedure manuals, business forms, buck slips, interoffice memos and the minds of Juan and Alice, who always explain things to newcomers. Now it can be exposed in inspectable modules.

Modularity for us

Indeed, modularity works even for newsletters. This issue of Release 1.0 is the first half of a two-part exploration of groupware. In it, we explore these points with reference to the varieties and modules that comprise the broad field of groupware, and then focus on the platform offerings/promises of Lotus and Microsoft. We also describe two new examples of information-sharing groupware that elegantly extend a field led by Notes. In our September issue, we will look more deeply at workflow and scripting tools, with some examples.

WELCOME TO THE WORLD, TALIA!
Pardon us, but these are serious distinctions...

The whole concept of groupware has been subject to widespread confusion, the flip side of the success it owes to Lotus Notes: People tend to consider Notes the embodiment of groupware. Notes is only a single, albeit important, example of groupware (just as 1-2-3 is only one example of singleware). Now that there's a multiplicity of purchasable, installable systems out there that do more than calendaring and e-mail, users are beginning to understand the distinctions among them. In previous issues we've posited a framework for groupware (see Release 1.0, 11-90): information-sharing tools such as Lotus Notes, vs. transaction-oriented workflow tools such as FileNet, WorkMAN and Action Technologies' The Coordinator.

Basically, you can effectively build a modifiable, integrated system only after you have properly modularized it into its different components. That's why we're such a stickler for the separation of groupware into two kinds: As with client/server systems, until you've separated the parts, you can't put an arbitrary selection of them back together again. What is confusing is that most people consider it more natural to separate hard and soft content (data vs. text/image), or client and server. But the appropriate separation is active workflow vs. passive information-sharing -- whether that information is "hard" or "soft." (You may even use the same data storage for both workflow and information-sharing, which only makes it all more confusing.)

Thus, in this issue we divide the groupware world into three broad areas: workflow, information-sharing, and traditional applications that accomplish "work" -- database transactions, say, or a user's work with spreadsheets, (compound) documents, sales reports, etc. The first two are both considered "groupware," but are actually quite different.

The third is what you might call "legacy" applications: From groupware's point of view, a spreadsheet is as foreign as a mainframe reservation system. Legacy applications are what workflow groupware manages, and their contents (task data) can be made available to users by information-sharing workflow as well as traditional applications or query tools. (Finally, there are a few genuine groupware applications that involve information transactions between users -- primarily scheduling, negotiation and delegation. This is the province of Action Technologies, in our next issue.)

Platforms for groupware

Groupware has different data storage and management requirements from more traditional applications. In mainframe days, the assumption was one file, one user; and that user was generally some monolithic batch application that chunked and clanked away all night (unless it abended). When on-line transactions and multi-user access appeared, developers created databases to manage access and concurrency control to data elements on a more atomic basis. But at least the data was regular and transactions were fairly quick; the problems were mathematically elegant and the solutions precise.

In the world of groupware, you have more varied needs for access to less regularly structured and defined data. You can no longer divide files up into neat data records as a database does, nor can you simply assign one large file to one user at a time. You need access controls, allowing some

Release 1.0

31 August 1992
people to change certain items in certain ways under certain conditions, and preventing others from even looking at the data.

In particular, information-sharing groupware supports a variety of users who want to look at and add to data, and share it either broadly or with particular other people. There may be a defined flow for the information, but the basic activities are looking and adding, not sequential changes. This requires a storage system that makes the information widely available and allows easy additions. In practice, Notes replicates its databases on multiple servers.

Workflow groupware is more concerned with managing work processes that involve changing and updating information at an atomic level. Storing plain old files will no longer do: You want object management -- a better way to define, find and manage access to the objects, on a more fine-grained level, and you want to be able to deal with complex, compound objects (like a database, but for objects).

Like information-sharing groupware, workflow needs the primitives to build complex security and access and version control rules, and the ability to find objects wherever they are. Replication accomplishes easy access by putting identical objects in several places, but that's dangerous with workflow, since the assumption is that people who have access to objects in a workflow will change them (as opposed to adding items or comments, as in information-sharing). Thus you’re better off keeping a single copy of each object, rather than trying to manage and reconcile proliferating copies. (Certain resources such as directories and software itself may be replicated, usually only one-way, but the default is one data object, one copy.)

Meanwhile, not only do you need to store the data needed by the workflow-managed tasks; you need to store the data needed to manage the tasks. Who has what? Who should get it next? If Juan doesn't like it, who can override him? Where is the artwork for the fourth chapter and did Alice do the captions yet? In simple workflow, the "state" of the objects is typically stored with the objects; in the object-management model there's only one copy of each object, and so you can get clear, real-time information about each object's state. But there's no central repository of control or knowledge about the process as a whole, and no way to monitor process integrity. That's why most industrial-strength groupware, with multiple complex interactions and sequences dependent on each other, keeps its own process information in a database regardless of where the work content is kept.

To workflow groupware, the user is a server. It asks the user to do some work, then takes the results and sends a request for further work to another server/user.

Obviously, the challenges in design and performance in both replication systems and object management (to say nothing of database integrity) are enormous... which is why it has taken Iris Associates five years to build Notes for information-sharing (although it could have done it faster if OS/2 had been out earlier) and why Microsoft's object-oriented Cairo won't be out until 1994 -- at the earliest, and in its earliest version. Notes is a multi-server structure that sits on top of a variety of operating systems,
Yes, this chart is confusing, because all the modules can and do talk to each other. The workflow shown at upper left is a "meta" system; it controls the flow and sequence of work, which is accomplished by users and legacy applications in the middle. The workflow may be managed by a workflow script or rules (which use data attached to the work objects), or by a full-fledged transaction-oriented workflow application which stores its own data. Workflow logic can drive the e-mail server, present items from the database for a user to work on or move documents around in an information-sharing system. The e-mail server can deliver messages to or from a text database/filter agent. And so forth. Most of the work content is kept in various server databases, shown at the bottom, but some may sit in files at a user's workstation or be managed by a (network) OS. Information-sharing groupware controls structure and flow of information (not processes). At the bottom are various data stores; they contain the work contents and possibly, in a logically separate database, the information that drives the workflow: where things are, who gets what, the status of any particular piece of work, etc. This database may also be a repository for workflow templates and applications. There may also be an e-mail server, or other application or communication servers. All these servers serve both users, and each other. The information store at the bottom may be a relational database, a Notes database, and object-oriented database, or even just a file system that holds documents, a name service, e-mail messages and the like.
Modularity in all things: The perils of openness

Groupware is all about how you put things together, in the gestalt, not just the technology. First, the plumbing underneath, as described above, must work seamlessly and flawlessly -- a challenge that Notes developer Iris Associates spent years to meet, and that Microsoft now faces. In the middle, as we'll see on examination of various tools and workflow modules next month, the components are actually quite simple, much as impressionist paintings are just a lot of dots when you look closely. But at the top, you need the vision to put the right components together the right way, with an interface that makes it all seem intuitive: "Of course it works like this!"

Just as users rejected vendors' offerings of integrated packages that do everything in favor of suites of tools that interoperate, so will they move to groupware tools that interoperate instead of single packages that do everything. The basic benefits are flexibility, and interoperability. Groupware tools need to interoperate with each other, since no one tool can do everything, and groupware tools need to interoperate with other applications that do the actual work.

Of course, modularity is not in the interests of the leading vendors, which offer a "complete solution." It is, however, in the interests of follower vendors, which want at least a share of the market: better a module or two than nothing at all. Thus Lotus is trying to define groupware as Notes or nothing (although you can buy the server standalone) and is adding workflow to Notes, while Borland (with its OBEX "collaboration engine") and Microsoft are offering groupware components that they promise will add up to [Notes + openness]. Microsoft owns (de facto) an operating system and should succeed in that environment whereas Borland will face a crowded, confused market.

On the other end of the spectrum NCR (with Cooperation) can afford openness because it owns not a product but a customer set; it is a systems integrator building complete solutions for its customers, using licensed tools plus a little proprietary interface and integration technology here and there. DEC, although it is less far along, is in a similar position. WordPerfect can also leverage its customer base. Then there are smaller companies which are in effect systems integrators with their own or licensed technology and a particular focus, such as Boss Logic, Cimlinc, Edify, ELF, FCMC, Verification, Computron and other image-oriented systems. (See our next issue.)

Separately, there are some module vendors that have no pretensions. They include grapeVINE (page 17), which provides intelligent information-sharing.
and filtering (it uses Informix and e-mail but it would be a nice front-end for Notes as well); Beyond Mail, an e-mail user or server agent development tool; Reach Software's WorkMAN, which comes with its own e-mail system (MailMAN) but can be used independently; and Action Technologies' process workflow tools. Corporate Memory Systems (page 21) sells its tool as a complete package, but it is talking with vendors such as Lotus in order to integrate it with existing server and communications platforms.

Market shape and share

Of course, Lotus Notes comprises a potential cross-platform server standard, a good substrate for e-mail and information-sharing applications and a potential complement for document-oriented workflow (along with an excellent but specific user interface). Lotus is now starting to encourage third parties such as Corporate Memory Systems, grapeVINE and others to provide complementary front-ends. Lotus has finally realized that otherwise those vendors would simply use other servers, and Lotus would get no revenues and lose market influence. In fact, even Notes server clones (under license, of course) would increase Lotus's market power. Lotus's goal should be to make its server a widespread standard for information-sharing groupware and indeed for storage of information for document-oriented workflow, while also selling hundreds of thousands of copies of its Notes front-end. (As a user, you can and could always buy back ends or front ends interchangeably, for about $500 each at suggested retail.)

By contrast, most of the workflow groupware companies offer application (client-side or server-side) tools and interfaces, which should eventually run with any relational/SQL database server.

Good interfaces make good neighbors: Encapsulating legacy applications

Much of the current discussion in downsizing concerns ways to encapsulate legacy applications -- how to take not just the data but the functions and logic of old applications and incorporate them into new systems. Sometimes that involves rewriting them, but sometimes it simply involves linking them into the new system, so that their inputs and outputs can be used.

That's what groupware does too. It's just that the legacy applications or data may be not mainframe systems, but rather existing pc applications, or manufacturing work nodes, or information kept on mainframes, servers or at users' workstations.

The fundamental issues are the same: The new software incorporates and understands the old. It operates at a higher level, and can manipulate the inputs and outputs into the old systems to manage business processes. Just as the logical separation of client/server relational databases made data management more flexible, so will the modularization of groupware make the management of business information and processes more flexible.

1 Action is a veteran in the openness business; it long ago hived off its messaging system, MHS, for licensing and eventually sale to Novell. Now it is an OEM supplier of workflow technology to -- or alongside -- tool vendors such as Lotus and Verimation; see our September issue.
UNDERWARE FOR GROUPWARE

Information-sharing groupware is about collecting, sharing and structuring information -- not changing it. The basic defined actions on such information are to add or delete an item, and to prevent or allow specified users to change items in undefined ways; all other changes are application- or content-specific and internal, like changing a word, italicizing a phrase, moving a line in a diagram by 2 centimeters, reformatting a page, etc. Or you can add a comment (with a pointer or other reference to the original). The information as a whole changes because new information is added. Typically, the new information subtly changes the meaning of what went before, but it's usually not simple data such as the value of a checking account.

Lotus: Notes across nodes

Lotus Notes is the single commercial product that best implements such a model. Developer Ray Ozzie spent almost five years at Iris Associates building the substrate to support such information-sharing, initially for the Notes client and now increasingly for third-party tools as well. (To build it now would probably take three years, he estimates, given that others could build on what he learned, and that there's now better operating system support from OS/2, UNIX and other platforms.)

The character of Notes compared with a typical relational (perhaps distributed) database is simple: Notes uses a replicative, additive data management system that collects information and adds it, keeping track of versions and relationships among items. With a database (object-oriented or relational), you have the tools and logical infrastructure to change information inside or about an object, whereas an image/text/Notes database is more concerned with the arrangement, structuring and presentation of these objects as complete elements that can't be changed but only added or replaced as a whole (within limits). When these rich pieces of information are changed, they are edited piecemeal rather than altered in a defined, quantum way.

Each Notes server typically holds multiple databases, each focused on a separate information domain. Notes manages copies of each of these databases replicated on servers in a number of locations. An authorized user can add information to a database at any of its replication locations.

The changes to each instance of a replicated Notes database are sent around the system on a careful schedule that keeps everything consistent within a specified lag period. This is akin to what many systems do with directory servers, except that beyond a certain range, directory servers typically aren't synchronized but incorporated by reference, to avoid excessive overhead or because "foreign" systems don't like to share their information. Moreover, Notes has two-way replication, whereas directory servers usually work on a master-slave basis where one copy is primary.

--------

2 In short, in the information-sharing model, you don't get back the original by reversing a series of transactions, other than restoring or deleting what was removed or added; you generally have to get a keep a set of time-stamped snapshots. But you don't really care; you want the summation of the information, not its current state.
Like a database, Notes also manages complex facilities to ensure integrity and consistency of the data, controls access privileges and checks authenticity of users (useful for signatures and approvals in workflow).

However, it does not manage atomic transactions as an object-oriented database does; for information that is additive that's not necessary (cf. the difference between a group editing system, with interactions, and a group idea-collecting/brainstorming tool, where everyone just contributes his piece and comments on others' work but does not change it).

With a distributed relational database, by contrast, you typically keep subsets of the basic data distributed among several locations, to be reassembled as necessary. In the distributed database model, one typically changes information in a single local database, and the changes are propagated as summary information to other databases through transactions as necessary. (Details are distributed when a query asks for them.) Other locations can be updated through transactions on individual data elements within records, but in general, "replication" is considered to be "redundancy" -- a potential source of inconsistency.

Even as software technology becomes more modular and Windows allows a variety of applications to interoperate, the packaged software business is becoming more integrated (read "consolidated"). Vendors are offering "suites" of packages or de facto site licenses at a single all-inclusive price, effectively closing out the competition through monolithic packaging instead of monolithic architecture. Although users like to see their suppliers compete, they also like the convenience of buying from a small number of vendors. (It's not just a devious vendors' plot; bulk purchasing is more efficient unless it allows vendors to get away with inferior products.)

Background: The benefits of Notes

Notes and similar systems sell on the basis of providing "corporate history," tracking or information dissemination. They are useful vehicles not only for internal information but also for dissemination of external information such as news feeds and electronic information services that will proliferate with the complementary proliferation of dissemination tools. DowVision, Ziff Communications, Desktop Data, Mainstream Data, Individual Inc. and the Internet's news groups are all part of this world.

Within limits, yesterday's and even last year's information are as useful as today's. If all you wanted was today's news, you could read the paper. What you want is to read today's events and then explore the background, to get from today to yesterday: How did this decision come about? What is our record with this customer? (When did Woody first take up with Mia?) What was that regulation issued last September that will be effective next month?

Precisely because such a database accumulates information and grows rapidly, it requires navigational and structuring tools to make it usable. Within each database, Notes does far more than, say, an ordinary set of text files or the typical index structure of a text-retrieval system. Its data-struc-
turing facilities make the information more intelligible to humans. They can query by keywords or values in fields such as date, topic, author, skills in a resume or the name of an applicant. Notes also includes tools for users to create a variety of data structures and easy-to-grasp views of complex data and relationships.

The Notes back-end helps maintain a broad and consistent database of information across locations for the Notes tools, but this same back-end would also be ideal for information-managers such as say, Corporate Memory Systems or grapeVINE or other information-oriented systems -- as it already is for Verity's Topic, SandPoint's Hoover, Desktop Data and IMRS's Executive Forum.

Notes also includes electronic mail, which is implemented as a separate routing service that sends messages to an individual's mail database (so that the user doesn't see the underlying difference). Most Notes databases are shared, but mailboxes by default are accessible only to a single person. This means that you can build an e-mail-based information flow.

The Notes database can be active, classifying items and storing them in the right places and distributing them to the right people. Moreover, you can create an application that will, for example, change a field from "pending" to "approved," or add an item to a resume. Because it is inherently multi-user, Notes scripts are inherently "group"-oriented, controlling information flows from user to user.

But Notes has no model for changing the internals of the items. The primary activity in such a system is adding information, not changing it. Thus if you want industrial-strength workflow with process integrity (notification, tracking, completion and other logic) you have to put it in yourself -- or use a tool or module that does so. Although Notes is ideal for information-sharing and adding, it's less appropriate for carefully choreographed step-by-step sequences of tasks. Specifically, it's not very suitable underware for industrial-strength workflow, just as a relational database system would be clumsy underware for the functionality of Notes (even though an rdb could point to text records).

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

With a database application or a process workflow you're interested in reaching the final state at the end of the application, whereas with information-sharing you're interested in retaining and perhaps structuring most of the information encountered along the way.

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

New, improved Notes

Lotus says it will include improved "workflow" tools in the next version (3.0) of Notes, although the company isn't yet discussing details. Lotus is adding this capability as a technically separate module. Basically, Version 3 will have better routing facilities and macros; the macros will be triggered by users, events or time.

This will not be a robust tool for database-based workflow with process integrity (see our next issue), but it will certainly do the job for many users who simply want a way of routing and tracking information rather than

Release 1.0

31 August 1992
a process-design tool. To ease the way, Lotus will also provide a set of workflow templates with Version 3, for example a purchase-order request, which means most users will be able simply to fill in parameters and won't have to see the macros at all.

It's not that process workflow is so difficult to add, or so special that only one or two vendors could do it. You can certainly build workflow to interact with Notes, as Action and ELF are doing (next issue); it's a good place to hold the information the transactions act upon when it happens to be textual information instead of database records. Lotus could do that too; it's just that it may muddy the waters by claiming to have it already. It will sell a "workflow development platform" as part of Notes Version 3, and it will also probably sell under the Lotus name a separate "workflow methodology and tools for applications with multiple related workflows."

Information and process workflows

As we will make clear in our next issue, there's a big difference between information flow from person to person, commonly called workflow, and what we call (by default) process or industrial-strength workflow, where the movement of multiple work items and sequential or parallel tasks among people must be coordinated and the completion through a series of conditional steps must be guaranteed; call that process integrity -- an analogy to transaction integrity in typical data-oriented database transactions.

Lotus's initial approach to workflow is to attach the state information to the object itself, rather than keeping it as a separate "workflow" database pointing to the items involved. That makes sense, we believe, as long as you're dealing mostly with user-to-user transactions. The user is presumed to have his own set of work objects that he cares about (or that get inflicted on him), and he can do a query if necessary for all the objects that have his name on them (both figuratively and logically). But you can't depend on an arbitrary replicated object to give you reliable real-time information as to its state, when it might have been changed elsewhere.

For a more robust workflow environment where an administrator wants to manage and coordinate a large number of workers on interrelated projects, it makes sense to keep a separate, transaction-oriented database. (Yes, you could assemble such a database dynamically each time you needed to, but it's simpler to construct things that way from the ground up.)

Replication vs. transactions

By contrast with rich shared information, the information managed by workflow groupware (the state of the work tasks as opposed to the content of the work) does change in simple, well-defined, easily quantifiable ways: i.e., transactions and states. Juan either has the document, or he has done his part and sent it on to Alice. It is approved or it is sent back for an overhaul. There are 49 overdue applications, with an average delay of 2.3 days. The document has completed five steps, and has from six to eleven to go, depending on the values of certain parameters that will determine who must do what to it.

Workflow operates well on a more traditional database model, where the state of the objects in the system is maintained by the database and process in-
When something happens, it's a database transaction that changes the state of the flow's constituent tasks. Yes, you can get a history (just as you can get a history of a checking account), but the primary focus is the current state of things and the completion of the process. Workflow applications are like database applications -- series of steps controlling routine movements of data or performances of tasks.

Thus Lotus's addition of information flow is somewhat like a relational database vendor's addition of objects: Yes, it works, and it solves some of the problems, but it doesn't build an object-oriented database in the first case or workflow with process integrity in the second. Although Notes will manage simple, single-document routing -- and that can accomplish a lot -- it won't easily scale upwards in complexity (as opposed to volume). That requires a fundamentally different approach.

By contrast, workflow tools based on a natural transaction-oriented platform (and related database tools) will find it easier to extend their products naturally -- and will be able to benefit from third-party developments in database tools. That's why modularity helps rather than hurts... technically, at least.

A pointer to the past

Lotus has received considerable criticism for the alleged "closedness" of Notes. While you could argue that Notes was always "open" because its server had defined APIs, Lotus didn't loudly proclaim that fact. The next version of Notes, due this winter, will have more and better-documented APIs. Moreover, Lotus's marketing is also becoming more open, as it encourages a broader range of third parties with Partner programs and other support and starts promoting the product as a platform as well as an application.

People have also complained, with less justification, that Notes is non-standard. But "nonstandard" doesn't necessarily mean closed: Of course something can't be standard if it does something no other product does -- and that's true of Notes. Moreover, the product is gradually becoming technically more open on the client -- mostly as a result of its integration

---

3 A third kind of tool is the "database-based information manager," which uses a database as the structure to manage complex objects such as compound documents. The different components of the objects are represented as database fields, or referenced by pointers in the fields. (This kind of application would actually work best with an object-oriented database rather than an object-oriented front-end, but that's another story, covered in earlier issues of Release 1.0. Thus, one purpose of compound documents is to make them into modules small enough to add or delete through a database-style application; for example, you could automatically assemble documents by selecting the right paragraphs and merging in name and address files. Or you could use a product configuration-spec to produce assembly instructions and documentation simultaneously, by running the same parsable spec against two separate sets of text modules. A third iteration might produce instructions for some kind of automated assembly system -- or perhaps a workflow application. This is the kind of thing Cimlinc and Boss Logic specialize in. See our next issue.
with Windows 3, which allows for considerable interoperability with other Windows applications through OLE, DDE and other facilities (with SQL and CDA inevitably on the way).

Yes, there's substantial evidence of Notes' openness. But while originally Lotus seemed to have a deal to resell the Action Technologies technology as part of Notes, now it wants to leave the field more open by not blessing a single outside vendor -- even though it will still be reselling a separate product jointly developed with Action. (If you see a contradiction here, so do we. It should just bless a couple and have done with it.)

Lotus's attempt to have it all -- to provide a full system rather than a complement to workflow tools -- reminds us of the early days of 1-2-3. The product was such a success -- most people were buying 1-2-3 and something to run it on -- that many people used it as a word-processor. Yes, it was clumsy and awkward, but users dependent on 1-2-3 as a spreadsheet didn't mind the extra effort because it let them stay with something familiar. In the same way, many Notes users may well use Notes' information workflow.

Ironically, Lotus's follow-on to 1-2-3, Symphony, was an integrated system (in response to market pressures at the time) that included far better WP along with database, graphics and the like. It did okay by any standards other than Lotus's or Microsoft's, with cumulative sales of about a million copies. But most people stuck with 1-2-3, the clean original, and ultimately bought some other word-processor. (Lotus eventually entered that market the modular way, with Ami Pro.) Meanwhile, Symphony developer Ray Ozzie left Lotus on good terms to found Iris Associates, where he subsequently developed Notes for Lotus.

Lotus faces a presumption of closedness because of its legal tangles with Paperback, Mosaic and Borland over the 1-2-3 interface. We happen to think Lotus had a good case, though the PR cost may not have been worth it. The obvious answer, both for the 1-2-3 interface and the Notes server, is license others to copy it! (The same goes for the Mac interface.)

Release 1.0 31 August 1992
MICROSOFT: THE CHALLENGER

Technically, we're delighted that Microsoft seems to have all the answers. Cairo team leader Jim Allchin comes from a UNIX and networking background at Banyan, where he implemented an early distributed computing environment generally considered one of the industry's finest. Cairo, basically the follow-on to Windows NT, will have all the attributes of OSF's Distributed Computing Environment, plus a programming/user environment that will be familiar to a wide installed base.

Microsoft's primacy in providing the substrate for the next generation of groupware (since singleware is passe) looks unassailable. Even though it's late, Microsoft's soup-to-nuts marketing but modular technical approach should rapidly take over from vendors who supply only part of the platform. Yet it still troubles us that a single company should own it all, however convenient this may be in the short and medium run. In the long run, we firmly believe that the greatest value to users, and the financial rewards, will go to people who know what to do with all this technology -- i.e. system integrators and application implementers.

Microsoft, along with Borland and Novell among others, is trying to catch up with Lotus's lead in groupware not by cloning Notes but by providing its own sets of modules and underlying network OS facilities. It's worth noting that it will take the better part of an operating system to do so. The Microsoft technology, Cairo, will emulate Notes' functionality only partially, but will be better in its own right for certain workflow applications.

All your objects in one basket

Scheduled for release to users some time in 1994, Cairo will incorporate the object filing system for Microsoft's new operating system well as an object-oriented management layer. It will handle all the usual distributed computing tasks across systems, including not just object exchange but also (like Notes) security, integrity, authentication and the like. Basically, it's a network operating system with an object request broker, managing not just data objects, but also access to WOSA services and resources including communications gateways, and to users. Like Notes, it will also be scriptable (for groupware or user agents) with Object Basic and other tools.

But there are big differences. For starters, Notes manages files within the operating system, while Cairo will manage from the disk (physical storage) up, and will run the operating system's data management by default. Architecturally, Notes leans to replication of data, while Cairo defaults to one data object, one-place. On a more conceptual level, Cairo's design center is the connected, online organization, with immediate access to everything and with Windows everywhere, while Notes' is the heterogeneous, intermittently connected, widespread organization -- perhaps a more realistic vision for the foreseeable future for most organizations.

Both allow for the user who wants to carry a subset of his work home or on the road. Notes provides facilities for subsetting Notes databases and manages the replication process as a matter of course; you can also build version management and object reconciliation processes with the Cairo primitives, and Microsoft plans to, in a facility cleverly codenamed "briefcase."
Finally, Notes was built from an application demand-side perspective, focused on a single architecture/homogeneous experience for users, while Cairo is a set of system supply-side services for developers. That distinction is fine, but the two don’t quite meet in the middle: Things are meant to be replaceable in Cairo, despite Microsoft’s marketing, while Iris Associates designed a powerful underpinning to support a modular but more tightly fitted application environment.

"We’ll do a for group productivity the equivalent of what GUIs did for personal productivity," says Allchin.

---

GUI = Group User Interface

Just as DOS provided the file system for simple single-user applications, Cairo will finally provide a powerful network file system/environment for powerful group applications, including those that use complex compound objects. For example, no longer will anyone need a separate e-mail server: E-mail messages will be stored and distributed as message objects, with arbitrary attributes while the e-mail front-end will be used to create messages (although you might want to use your word-processor, and then just send the message to the recipient by selecting send from the menu and selecting his name from a list). For more complex tasks, like routing the message or guaranteeing a response within a certain time (or you’ll both be reminded you’re waiting for it), you might have to go as far as a mail dialogue box, implemented as one of many services (from Microsoft or a third party) within Windows’ Open Services Architecture.

Indeed, the only thing that needs to store stuff outside Cairo is probably a powerful transaction-oriented database or object-oriented database. It could replace most e-mail databases, directory servers, and most BLOB databases. A full-text search system could store its indices as Cairo objects, with references to text items also stored as Cairo objects. Cairo will also, of course, compete effectively with whatever Novell is building.

Needles in haystacks at your fingertips

Cairo’s distributed object-oriented file system will have many attributes of an object-oriented database/object request broker. Generally, it will operate transparently to the user, and will allow him to find any file or object or resource or user (these are the same as far as Cairo is concerned, although with certain different attributes) he has authority for. (Beware the costs of transparency when you start dealing with geographically remote locations, however. PTTs will make lots of money on transparent access between, say, Walnut Creek, CA, and Bydgoszcz, Poland.)

The major visible features of Cairo will be this transparency, along with the ability to find things by arbitrary attributes. You define ‘em; you (or an application) can use ‘em. That is, you can look for something by name, by author(s), by application (although that becomes irrelevant as more and more objects are compound), by date created, by location, or by any attribute you care to define and to list possible values for. You can also locate things by full-text search if you’ve indexed them -- another possible WOSA service. (But you’d better keep your search to a limited part of the
haystack. In fact, most users will work in a default domain defined by a Cairo administrator; to search outside it, they will have ask explicitly, although they still won't need to know the physical locations of things.

You could also define and search for things by state, especially handy for groupware: "finished," "sent by Alice and waiting for Juan to look at it," "more than two days past deadline," etc. -- or by relationship: "part of the documentation for Version 4." Cairo will be able to manage complex compound objects, and it will have basic primitives for version management system and concurrency controls (simple locking), but it will lack transaction management and other database-like facilities.

Microsoft has workflow modules too

Separately, in a group run by Daniel Petre, general manager of the workgroup division, Microsoft is working on two workflow modules -- basically tools that cleverly use the rest of the Windows services to create and run workflows. The e-mail-based tool, code-named Calvin & Hobbes, should be available in the first half of next year, with the database-based system, Torque, out later that year. (We know they'll probably change these names to something more sensible, but we wish they wouldn't!)

Calvin & Hobbes is more an extension of Visual Basic than a full-fledged application (or SKU), and will probably come in the box with Windows for Workgroups or a new release of Visual Basic. Calvin's a toolkit for building active, routable e-mail forms (with attachments), and Hobbes is a set of MS Mail form templates created with it. Calvin is basically an extensible set of custom controls for Visual Basic that automate sending, routing and replying to messages. It can do variable routing and works through a forms-based, select-a-name interface.

Calvin & Hobbes shows the power of the modular approach: With only a little extra effort, Microsoft will be able to extend the capabilities of its Mail package and Visual Basic to create a simple workflow toolkit. A user can easily write a script (mostly via buttons) to define the route (with conditional branches) he wants an item to take via the MS Mail front-end. Optionally, it can report its progress back to the user as it steps through the system. Of course, it can do quite complicated things for users who want to get into the intricacies of DDE and OLE. To the workflow user, Calvin & Hobbes will look like e-mail messages with forms and attachments. Basically, Calvin & Hobbes is equivalent to Notes' information workflow (minus the rich text-handling), using MS Mail instead of the Notes e-mail and routing facilities.

Turbo torque

Torque is a robust database-based tool for industrial-strength workflows. It will have a graphical front-end for visualizing workflows, and a variety of tools and forms to make the development job easier. Initially, the system will work with standard Windows files and applications, eventually with Cairo. Torque will have its own user interface, with forms, to-do lists, check-off boxes and the like. Also, it will be able to go outside the system by sending e-mail or call-me-back links, which will prompt the user to log on to the system. Petre's goal is to make Torque database-independent, with interfaces to MAPI and ODBC.
Once Cairo is in place, of course, Calvin & Hobbes and Torque will automatically become much more powerful because objects will be so much easier to find, manipulate and monitor, with better-defined APIs. The benefits of an object-oriented environment will be more fully realized with Object Basic (see Release 1.0, 5-91), which will allow users to script the interactions of applications and users -- i.e. workflow.

(Microsoft also has plans for group memory and bulletin board products, but it's less forthcoming about those.)

LOTUS AND MICROSOFT IN CONTEXT

We're impressed with Microsoft's approach to these issues. Although details aren't yet available, the company clearly understands the basic requirements for workflow -- better than most people, in fact, because it sees it as a tool or module for manipulating other applications rather than as a stand-alone application itself. Third parties will have to compete on the basis of functionality, interface and application-specific business knowledge that they can use the tools to implement. In the end, workflow is a generic capability. It may not be true to say that once you've seen one you've seen them all, since vendors can certainly add their own little touches, but once you've described one, you've more or less described them all -- as we tried to do briefly on page 10. Thus we're delighted that Microsoft has figured this out, since workflow is a generic facility, even as we're sorry to see the industry leader address yet one more market.

The scheduling tool in Windows for Workgroups already indicates careful attention to the issues. As extended for cross-network use, the broad-based version will include a replicated, compact file that simply indicates the availability of everyone linked into it as busy or free for each time period. That enables a fast response. A second module allows dynamic but slower queries across the net to find out the whereabouts (with authorization) of any particular person, so that you can track him down. Thus the system maintains the proper amount of detail in the right places, with the overview available instantaneously and redundantly anywhere, and the details kept locally but available on request.

Where Microsoft does not yet have a solution is the sort of text-based, rich-format information-sharing tool that Lotus offers in Notes. You could certainly emulate the back-end by adding in replication logic to Cairo, but there's also a considerable amount of work Iris Associates has already done to manage authorization, consistency, integrity and authenticity. We have no doubt that in the long run Microsoft's Cairo team could do equivalent work, but Iris is not standing still in the meantime.

Meanwhile, we don't think that by offering Notes unbundled Lotus will suddenly find that the Notes interface is widely replaced by third parties, despite our enthusiasm for grapeVINE and CM/1 (next page). We believe the Notes application front-end will be broadly used and is gaining wide acceptance already. It offers a uniquely accessible, intelligible way to organize and view a wide variety of mostly textual information with its combination of hierarchies and tables. It's just that by "opening up," even if that's something of a spurious issue, Lotus stands to gain even wider, more trusting acceptance of Notes. Openness is marketing as well as technology.
INFORMATION-SHARING

You can automate only what you can express -- for example, a five-step workflow. On the other hand, you can assist people in structuring, organizing and sharing information that is not meaningful ("expressible") to the system as long as you have some explicit way of classifying and structuring it. The easiest way is simply by putting it into fields and records in a relational database, but there are other ways (see Release 1.0, 3-90, 4-91). You can also "structure" it by indexing it, either alphabetically or according to a table of contents or a hand-built hierarchy such those used with Verity's Topic, or you can represent the document as arbitrary-length fields (BLOBs) in a relational database.

Such simple text search-and-retrieval tools can be enhanced by capabilities like those of CM/1 and grapeVINE which let users assess information; then the tool automates the structuring and sharing of it. Even Notes and plain old bulletin board systems also implicitly let users classify their texts by where they place them in a discussion. The software gets the user to declare explicitly not the "meaning" of something, but its importance, its relation to other points, or other attributes, and then works with those to manipulate the item usefully.

The two tools described below are quite different, and make the point that even in the information-sharing world, there are things Notes never thought of. Broadly speaking, grapeVINE classifies while CM/1 structures; both use the assistance of humans to do so. grapeVINE works as a broad-scale information-sifting tool, finding items on a broad range of topics and using both electronic and human filtering to classify them and get them to the right people. CM/1 works on a smaller-grained scale, helping people involved in a particular discussion visualize and make explicit the relationships among all the relevant facts and arguments (and winnow out what's irrelevant).

grapeVINE: everyone an editor

grapeVINE is the commercialization of a University of New South Wales research project that could only be realized by launching it into a world of real users focused on real problems. It is marketed by Office Express Pty Ltd. in Sydney, Australia (as opposed to a group contact manager and scheduler of the same name from Finland). The name suits the Australian product well; it's a value-added information-spreading tool.

Basically, grapeVINE starts like all the others: The system sucks up any kind of documents, including news feeds, e-mail, corporate bulletins, draft documents or sample sales promotions, customer feedback, even user questions requesting information or Internet news feeds (for which it uses a special parser/scanner). The text items are matched to topics of interest by a thesaurus, a rich set of keywords and synonyms.

The keywords/topics are distinguished as broad, medium and narrow. All items about, say, Microsoft are considered to be about software, but someone interested in software will not necessarily get all the items about Microsoft, just the ones of high priority (as described below). grapeVINE creator Cyril Brookes notes that any given group probably has only about 20 broad topics, but no one wants to see all the items under the broad topics -- just the important ones or the ones specific to their subtopic.

Release 1.0

31 August 1992
This part of grapeVINE is akin to Verity's Topic; like Topic, it requires substantial upfront effort by expert users or consultants to customize a topic set and word lists for each server location (though several servers could share the same thesaurus if they shared interests). Of course, you could reuse industry-specific starter sets, but that's off in the future. The keywords for a topic can be added to or altered by any user with the proper authority. There's also the notion of soft topics, or ones that change regularly. For example, "rumored products," "announced products" and "shipping products" would change keyword product names from time to time (although more slowly than planned, usually!).

By corporate fiat, each topic must have at least one person who is interested in it (or assigned to it, anyway). Items that match no topics are discarded, since no one would receive them. Then comes the interesting part: People can express a threshold level of interest in any topic -- low, medium, high or urgent. Each user will be alerted to any item of the specified level of priority or above. That is, only the person immediately responsible (with a low threshold) wants to read everything about, say, Microsoft. (That alone could be a full-time job.) But this person can then effectively reclassify what he reads, which the system by default has listed at low interest. This is done by appending a comment at a higher priority level, which will then reach a wider audience. Readers can then use grapeVINE's threads (vines?) back to the original item if they want to see what sparked the comment. (Anyone can de facto raise the priority of an item simply by appending a high-priority comment, but a lower-level comment won't lower the priority of the original.)

Thus important items can be picked out of the chaff by someone who presumably has the topic-specific expertise to recognize them. While a keyword system can easily recognize a topic, it can't easily assess the value or importance of any piece of information; that's a task best done by humans.

Capturing intellectual energy

grapeVINE accomplishes that wonderful trick of getting people to do useful work as a byproduct of their normal actions. It builds an inbox for each user and fills it with all the items that match his profile. (Actually, the user just gets a pointer to the item, which is stored only once on each server.) The user has three choices to handle the item (and get it out of his inbox): delete from in-tray (not from database), retain, or comment.

Commenting is the innovative piece. In addition to changing priority, users can add value by providing perspective, details or other relevant information or cross-references; they can also assign new keywords to an item. For example, a knowledgeable reader might add the keyword Microsoft to an article about bugs in Windows that never mentioned Microsoft -- and he might also add the keyword "undocumented APIs" to the "Microsoft" topic. A reader can also restrict access to his comments, if he's authorized to do so.

---

4 This is why groupware didn't work before things were electronic; people wouldn't stop doing "useful" work in order to record their activities on some separate electronic monitoring system. For what it's worth, television rating people have a similar problem getting people to report what they do.
If all that's added are comments, then those comments will be sent only to people who would see the original item. But if the reader raises the priority of his comment, the item reaches a large circle of readers (unless limited by security). Thus a routine report about bugs in Windows 3.1 might reach only five specialists, but a medium-priority comment (about how the grapeVINE-using company could respond competitively) might reach a larger audience of 50 people in product development and marketing who are interested in medium-importance items about software generally. They wonder if they should formulate a competitive response.

At that point, a salesperson gets a complaint from his customer Worldwide Amalgamated about the bug. He looks up Microsoft in the grapeVINE system, finds the bug report, and adds a message about his customer complaint, escalating its priority and adding the customer name as a keyword/topic for that message. Suddenly the item is escalated further, to another 50 people in marketing. Moreover, it is now also assigned to the Worldwide Amalgamated topic and ranked urgent in priority in that topic, so that all the salespeople calling on WWA offices are alerted and can call their customers to say that they're working on a third-party solution to the problem. Okay, development team, get busy!

The system thus manages multi-threaded discussions of topics, drawing in potentially interested parties. Value is added easily, and is easily accessible. Users can see a semi-graphical representation of the structure of such conversations, shown as a hierarchy through which they can browse to display items and comments. (The system doesn't distinguish between items and comments; both are indexed and pointed to by database records, and can point directionally to each other.)

Where the magic ends

Obviously, grapeVINE requires some initial human intervention: The topics and thesaurus must be defined, and at least one person must be assigned to monitor each topic. That's easy enough to require in an organization that takes information at all seriously. (Only such a company would be a candidate for grapeVINE anyway.)

What actually happens, of course, is that junior people whose time is cheaper or whose expertise is more specialized act as "editors" for senior or distant people less concerned with a specific subject area. These local experts could be salespeople handling a particular account set or region, competitive analysts focused on a particular market segment, technicians focused on a particular technology. As described in Release 1.0, 2-90 and 6-91, we can imagine a commercial version of this system, where subscribers might pay for referrals of information by a specified set of editors whose judgment they trust.

The benefits of grapeVINE are that the classifier/commenter needn't know who is getting the stuff that he escalates; it gets distributed automatically. Someone interested in a topic need notify only the system, not all the other individuals who might possibly forward something. Information gets distributed as effectively as any other resource, and newcomers can get into the loop easily. Yet the first-line filtering eliminates a lot of the chaff.
The ground beneath the vines

Underneath, grapeVINE is a database application that runs on a UNIX, OS/2 or VAX VMS server with OS/2, Windows or Mac clients. It can also talk to dumb clients via e-mail, and they can send their comments back via e-mail, using any standard e-mail package. The system considers a reply to an item a comment on it and places it in the right place in the discussion and forwards it to appropriate users, and an unsolicited message a new item, to be classified and distributed. The logical structure of the thesaurus is somewhat hierarchical, akin to Topic's, with categories and subcategories. The system can handle any kind of information, internal or external, that can be fed into it as text files or e-mail messages with sufficient text for parsing and indexing (images can be handled as attached files).

The database (currently Informix for pricing reasons, but it could be any SQL database) maintains a record for each text item. The record contains pointers to the text items and related comments, which are kept as text files, while the topics and priority rankings are stored as attributes in separate fields of each record.

In addition, you can set up additional remote servers as "users," so that the server sends them items just as it would to a user -- selectively by topic and priority. Thus servers can send information to each other, and servers can filter each other's output as appropriate for each local group. The headquarters can get important information from all branches, while each branch can get relevant information from headquarters without getting the details about other branches' customers. On the other hand, all the branches may get each other's information about Worldwide Amalgamated, so that they can present a united front.

Finally, the same process in reverse can be used for selective archiving, for items of low interest or with no comments. An automatic archiving process nominates such "dead issues" for the administrator. (Sic transit gloria mundi: This year family values; next year, who knows!)

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

grapeVINE uses fairly standard filtering techniques, but it enhances them with just the right amount of user input (without redundant work) to assist in the process of getting the right information to the right people. It solves the problem of so many companies of applying the huge amounts of internal soft information that normally are wasted because Juan didn't know that Alice need to know what he knew.... How could he know?

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

Practical matters

grapeVINE costs about $400 for the first 100 users ($40,000), and drops to about $180 at the 500-unit level ($90,000). Consulting, which is necessary, is extra. So far, the company has 12 customer sites, including the Australian tax authority with 200 users (don't tell the IRS about it!), M.W. Kellogg in Houston, and various consumer-goods companies. It has been shipping for only six months, and we see a bright future for it.

Release 1.0

31 August 1992
Corporate Memory Systems, Inc., probably should have kept its maiden name: It's well known in the groupware community for gIBIS (graphical Issue-Based Information System), a tool developed at MCC over six years and with $6 million in funding. Developers Jeff Conklin and Michael Begeman spun out on good terms from MCC in the spring of 1991 to form CMSI, using gIBIS to explore the nature of the product and its market (see illustration). They eventually settled on the name CM/1 for the product -- yes, the same as Thinking Machines' CM/1.

CM/1 is a beautiful example of an intelligent application of hypertext: It uses typed links to allow people to visually represent the structure of an ongoing argument -- pardon, discussion.

It illustrates clearly how far text-structuring and retrieval can go beyond simple word-search or even concept hierarchies such as Verity's Topic. Not only can you find ideas related to, say, protein folding, you can find items related in a particular way to each other -- the arguments for and against experimental or computer-based techniques to predict protein folding. CM/1 explicitly supports these link types: "suggested by, generalizes, specializes, responds to, questions (the polite form of contradicts or refutes) and
The links can also point to external items, such as spreadsheets, stored news articles or memos.

CM/1 itself, of course, is an example of the more general notion of typed-link hypertext. Looking at it, you can easily see how it could be generalized; technically, you could add your own, context-specific link types, such as requires answers to, examples of, earlier or later versions of, and so on, although the commands for users to do so aren't in the current version. On the other hand, CM/1's value lies in its specificity, and its application to a set of corporate problems. (A broader product would probably sell for less.) Time enough to widen the scope later...

The structure of the argument/display is usually tree-like, but is not a strict hierarchy; items can be related to several other items with various kinds of links. For example, a single point could refute one argument and affirm another. A single situation could be an example of two principles.

CM/1 assists in the process of communicating about, making and implementing decisions in three aspects: visualization, documentation and reuse of decisions. A "decision" need not be yes or no; it could be a set of policies, a series of discussions on an issue that was not resolved, etc. The common assumption is that decision-making is about getting people to agree; but frequently, it's about getting them to understand the problem even before they come to the point of agreeing or not.

----------

People new to a discussion can grasp it more easily than they could by reading piles of e-mail or worse yet, simply walking in cold. New programmers at CMSI spend their first few days on the job reading the issue nets on the design of CM/1. Within a few days, says CEO Michael Begeman, "they have picked up the designers' mental model of the system."

----------

Show me my mind; show me yours

Groups use CM/1 to grapple with and map out the components of complicated problems. It provides the solution to this complaint: "I have too many conflicting facts and opinions. I know most of them are relevant, but where do they all fit? I can't think of everything at once!" It helps people organize their own thoughts and communicate them to other people, who can receive them more effectively if they fit into a shared framework that clearly lays out the reasoning behind the decisions and the facts behind the reasoning. The system reveals hidden (or faulty) assumptions, helps focus people on the issues, and makes decisions and reasoning explicit. It provides a structure to a discussion that is precisely as complicated -- but no more -- than the content of the discussion.

This is the capability that gets people to buy the tool in the first place, says Corporate Memory's Begeman. Later on, the documentation and reuse aspects come into play -- frequently with different sets of users. In retrospect, people need documentation of decisions and policies -- not just a decision alone but how it was arrived at. One big class of CM/1 applications is, bluntly, cover-your-ass -- explaining to a superior or perhaps an outside regulator why certain actions or decisions were taken. Here are the
factors we considered when we hired him; here are the interim conclusions we reached; here is what we knew when we made that overoptimistic earnings projection.

There you go again

The third broad capability of CM/1 is reuse of work that went into the decision-making process, just like reuse of code. It helps you avoid a lot of arguments and revisiting of decisions, just as object-oriented programming lets you avoid redundant code. You can’t reuse either code or decisions whole-hog, but there’s a lot of work done that can be reapplied because the structure of the original is laid bare: "We decided such-and-such; now we face a similar situation, not quite the same, but close. Let’s put the new facts onto the old framework as a basic organizing structure." That helps both in deciding the implications of the new facts or arguments for the new situation, and in making sure that all factors are considered: "Oh yes, we tested that new keycap material, but unfortunately it dis-integrates when smeared with peanut butter."

Meta-information

A further interesting potential use of CM/1 is in analysis of the discussion process itself. As we’ve mentioned elsewhere, an interesting facet of groupware is how it makes the inexplicit explicit. Spurious arguments can be exposed as irrelevant: This particular point doesn’t really contradict anything, for example, even though it sounds negative.

Additionally, you can visualize a company’s style with CM/1. One of Corporate Memory Systems’ clients tends to be fast-moving, and you can see it in their graphs, says Conklin, president. They adopt positions and abandon them quickly; their webs of positions spread out long tentacles. Another client, by contrast, argues each issue deeply; each position statement is surrounded by pro and con arguments, comments on comments, in a dense web. Although Conklin isn’t impolitic enough to say so, we imagine you could also see companies where most of the comments were negative rather than positive, or a situation where no one contradicted any of the chairman’s positions.

Clients include Southern California Edison, Nabisco, Idaho Power (consulting only, so far), and the Institute for East-West Studies, which will be using it to manage interactions between New York and Prague for complex issues (once communications infrastructure issues have been sorted out).

CMSI works closely with its clients; the software is only a tool. Each installation starts with a few days of investigation and interviews by CMSI professionals, followed by a three-day customer offsite working on real projects. The users then take the beginning of their discussion web back with them to use as their exploration of the issues progresses, while CMSI continues to coach them for the next six months.

As yet, there is no published pricing schedule, but a project runs about $100,000 for a 20-person group, including everything but hardware (anything that runs Microsoft Windows and NetWare).
RESOURCES & PHONE NUMBERS

Tom White, Action Technologies, (510) 521-6190; fax, (510) 769-0596
Jeff Conklin, Michael Begeman, Corporate Memory Systems, (512) 795-9999;
fax, (512) 794-5921
Cyril Brookes, grapeVINE, 61 (2) 389-4800; fax, 61 (2) 387-8585; e-mail,
brookes@cumulus.csd.unsw.oz.au
David Wilkie, grapeVINE, (404) 455-8014; fax, (404) 455-0167
Ray Ozzie, Iris Associates, (508) 692-2800 x211; fax, (508) 692-7365
Brownell Chalstrom, George Gilbert, Cindy Schuyler, Lotus Development, (617)
577-8500, 693-4351; fax, (617) 693-1149
Scott Calder, Mainstream Data, (801) 584-2800; fax, (801) 584-2831
Jim Allchin, Daniel Petre, Microsoft, (206) 882-8080; fax, (206) 936-7329

COMING SOON

- Workflow -- structure and process.
- Simulation for education.
- Pen stuff.
- Constraint-based reasoning.
- A map of the wireless world.
- 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 communica-
tions consultant: William M. Kutik. Copyright 1992, EDventure Holdings
Inc. All rights reserved. No material in this publication may be repro-
duced without written permission; however, we gladly arrange for reprints or
bulk purchases. Subscriptions cost $495 per year, $575 overseas.

Release 1.0 31 August 1992
### Release 1.0 Calendar

<table>
<thead>
<tr>
<th>Month</th>
<th>Event</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>September 22-24</td>
<td>*UNIX EXPO - New York City. Sponsored by Bruno Blenheim. Ties are sprouting everywhere... Call Annie Scully, (201) 346-1400 or (800) 829-3976.</td>
<td></td>
</tr>
<tr>
<td>September 22-25</td>
<td>Seybold San Francisco - San Francisco. The basic show for electronic publishing. Sponsored by Seybold Seminars. Call Kevin Howard or Beth Sadler, (310) 457-5850.</td>
<td></td>
</tr>
<tr>
<td>September 23-25</td>
<td>Virtual Reality '92 - San Jose. Sponsored by Meckler. Call Gloria Allen, (203) 454-5840 or (800) 635-5537.</td>
<td></td>
</tr>
<tr>
<td>October 8-9</td>
<td>Lap &amp; Palmtop - Chicago. Sponsored by Laptop Expositions. Call Peter O’Connor, (212) 682-7968.</td>
<td></td>
</tr>
<tr>
<td>October 11-14</td>
<td>*Third annual East-West High-Tech Forum - Prague. Sponsored by EDventure Holdings. Meet potential partners and competitors in Central and Eastern Europe -- a market where demand still exceeds supply. Contact: Daphne Kis, 1 (212) 758-3434; fax, 1 (212) 832-1720; e-mail, MCI 511-3763 or CompuServe 75140,761.</td>
<td></td>
</tr>
</tbody>
</table>
| October 18-21  | TIFAA fall management conference - Colorado Springs. Sponsored by The Information Technology Association of America. Call Shirley Price, (703) 284-5355. |}

*Release 1.0* 31 August 1992
October 18-22  OOPLSA '92 - Vancouver. Sponsored by ACM. Call Rebecca Wirff-Bock, (503) 242-0725 or John Richards, (914) 784-7731.
October 20  New York PC user group meeting - New York City. Sponsored by NYC. With Andy Grove. Call Jim McMullen, (914) 245-2734.
October 27-29  PC EXPO - Chicago. Sponsored by Bruno Blenheim. Call Annie Scully, (201) 346-1400 or (800) 829-3976.
November 9-12  Software maintenance '92 - Orlando. Sponsored by ACM. Call Vaclav Rajlich, (313) 577-5423.
November 11-14  First general conference on Nanotechnology - Palo Alto. Sponsored by The Foresight Institute. Speakers include Esther Dyson, Marvin Minsky, Paul Saffo. Call Jane Nikkel, (415) 324-2490.
November 16-20  Comdex - Las Vegas. The biggest US show of all. With panel on information-sharing groupware moderated by Esther Dyson. Contact: Peter Young at Interface Group, (617) 449-6600; fax, (617) 449-6953.
Nov 30-Dec 4  Hypertext '92 - Milan, Italy. Sponsored by ACM. Contact: Julie Eltizer, (212) 626-0604; Paolo Paolini, 39 (2) 2399-3520; e-mail, paolini@ipnell.polimi.it.
December 8-10  DEXPO - Anaheim. Sponsored by Miller Freeman. Call KoAnn Tingley, (617) 232-3976.

Release 1.0  
31 August 1992
### 1993

<table>
<thead>
<tr>
<th>Month</th>
<th>Event Description</th>
<th>Sponsor/Contact Information</th>
</tr>
</thead>
</table>

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

*The asterisks indicate events we plan to attend. Lack of an asterisk is no indication of lack of merit.*
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.

Name __________________________________________
Title __________________________________________
Company ________________________________________
Address _________________________________________
City____________________________________________ State________ Zip________
Telephone_______________________________________

☐ 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