TOOLS FOR GROUPWARE

As our longtime readers know, we classify groupware along two axes -- information management and workflow management. Any system may have elements of both; also, a tool with one flavor may nonetheless be used to produce an application with more of the other flavor, although it's an awkward way to do it. Lotus Notes is the most visible of the group information-management tools, with an installed base of 65 large-company accounts with multiple sites and over 20,000 individual users. In the workflow-management area, there's no real leader, although the pioneers include FileNet, ViewStar (Release 1.0, 89-12), Systems in Concert (90-10), and Action Technologies' The Coordinator (86-10). But none of them has yet solved the problem of how to sell the stuff broadly (FileNet sells direct, but to a small universe).

In this issue, we propose another, complementary framework for categorizing and describing groupware tools. And we provide four examples to illustrate the diversity: Agility's Wijit, Beyond's Beyond Mail, Borland's new Sidekick 2.0, and the Knowledge Network, an information-dissemination system developed by Reach Systems of New York in conjunction with client Coopers & Lybrand. Although none of these classifies as full-fledged groupware, they are useful products that may in fact be easier to sell on the basis of their other features, without reference to groupware.

Commercial Issues

How to sell groupware? We spent an interesting lunch recently with a frustrated vendor of groupware. His product runs on pcs and networks and works fine on that platform. The problem is that his customers don't have network administrators; they know pcs are easy to use and don't need heavy-duty administration. Accordingly, whenever something goes wrong with the network (not with the groupware), the customers come back to his company for help. But he can't find the necessary people to support those customer networks, and he doesn't have the time or resources to train them. Neither, it seems, do network vendors have time to support groupware. Evernet, the nationwide company of local VARs specializing in servers and networks, is just getting its affiliates up to speed with NetFrame and Oracle. Groupware tools and applications are still off in the future.

The platform battle, in the end, may not be UNIX vs. OS/2, but UNIX vs. LANs. UNIX may well pull ahead not

ECHELON NEXT MONTH
so much because of product quality but because it will sell to UNIX users who (1) understand that it needs support, and (2) make such support and training are on site. This is good news for vendors such as Hunter Systems (see Release 1.0, 88-11) who are helping software vendors to make the transition to UNIX as painless as possible.

Likewise, groupware represents a big opportunity for the traditional vendors who lost out to IBM in the mainframe business, to UNIX in the mid-range, and to IBM and the clones in the pc business. Now their sales forces and support staffs can come into their own again, selling and supporting groupware -- if they can rise to the challenge of articulating these concepts not just to customers but to their own salesforces. Potential beneficiaries include NCR (Cooperation), Xerox, AT&T (Rh Rhapsody), HP (NewWave Office), Wang (with its procedure automation tools) and even IBM (with an enhanced OfficeVision).

However, when we went to the AT&T booth at Comdex, the salesman there told us Rhapsody was too innovative and difficult to understand to be demonstrated at a trade show. Maybe he was right. At the NCR booth, the company was demonstrating Cooperation, but the salesman couldn’t really figure it out beyond showing us the icons. (The only other people who had asked to see it, he told us, were from Seybold!)

Moreover, none of the groupware products has a compelling story yet. If all you want is personal productivity, a pc and some packaged software will do the trick. If you want real office productivity for tasks and work and business, you have to wait -- or have a specific problem to solve. Then you can buy something like FileNet at the high end or some of the products described below for specific tasks, but it’s early in the game.

The case for groupware

Ideally, groupware will integrate the freedom, ad-hocness and personalness of a spreadsheet (or e-mail) with the integrity, globality and connectedness of a database. Users want local power along with global reach. Yet they want to deal with the global system through the local medium -- be it a spreadsheet or a mail-style interface or a document (cf. Microsoft or Interleaf’s Active Documents, Release 1.0, 90-3).

Groupware can be powerful stuff, and the level of interest we’re seeing from large vendors is exciting. But it will be important for vendors to remember that it’s a concept that needs explaining -- albeit one embodied in products. Groupware’s power and the difficulty it will encounter both within vendors and out in the marketplace stem from the same fact: It’s orthogonal to most of what vendors are selling and customers have been buying. That is, it can link, integrate, coordinate and otherwise fit around existing applications, although that integration will not be a painless, automatic process. There won’t be a lot of money in selling the concept or even the products, but there should be a lot of revenue for consulting and implementation services. More than a way of coding or building applications, groupware is a way to define, structure and link applications, data and the people who use them.

Groupware incorporates both the grand picture -- users and information and action (work) -- and a tool for implementing it. In a world where not just hardware but even applications, user interfaces and tools are becoming commodities, competitive advantage will consist of the ability to define and
automate the right business processes. Currently, with widely available tools (and a little training) you can specify anything and (almost) automatically build a system to implement it; the remaining, unsolved challenge is to specify the right thing. (A handy marketing approach and a fundamental value-added offering will be preconfigured process templates for function- and industry-specific tasks, modifiable by customers.) Groupware will help both in defining, testing and implementing those specs, by helping customers understand and then automate their companies' missions.

But this design process is difficult. A groupware tool should be a thinker's aid: It should help the user to visualize and simulate a business process, to test it out before committing, as well as to implement and execute it later on. Current tools generally address only a part of the whole problem. Vendors tend to show you what you can do, without elucidating how hard it can be to do it.

A tool for change

If the challenge of the Nineties is change, this is a tool for implementing change in organizations, rather than just in data structures and applications. It deals with business, rather than with computer processes. The danger is that groupware will be perceived as one more module, both within vendors and by the world at large: "And for Mr. Jones, we'd like to order one decision support system, one financial planner and a business process management tool." This is what has happened with such efforts at companies such as AT&T, NCR and Xerox, where Rhapsody and Cooperation got buried in a flood of product announcements and were never well articulated to the public. ("Yes, we have workflow management; see section 23A-b of the brochure.")

Perhaps through fear of being grandiose or for lack of resources, vendors fail to comprehend the potential magnitude of the applicability of such a tool. It may be small in direct dollars, but it should be huge in impact (on account relationships as well as on business practices and system design). Larger vendors are uniquely positioned to market this kind of idea/product, which will require spreading the vision (as only a large, non-niche company can), providing support, and offering a truly useful tool to help make it work. Generally, large companies are too set in their ways to embrace such a vision (with an "installed base" of both systems and ideas), whereas smaller companies lack the resources and influence to do so. We'll be getting back to this problem later on.

What is information? Why, it's someone looking at and understanding data. Even a natural-language system or any other AI system doesn't really produce information until there's someone there to see it. The system may produce information and act on it, and then there's some value to the information. But beyond that, unless a person or a system acts in response, there's no value to the information. The action needn't be very physical; it could be a phone call, a command to a subordinate (cf. speech acts) that transfers an obligation, as well as a move to cut production, launch a new product, delay the price rise until after Christmas, solicit a certain customer. But until some action (or a decision to take no action) results, the information has had no value. Think about it....
A NEW FRAMEWORK FOR GROUPWARE

In the past, we have classified groupware as information- or workflow-oriented (see Release 1.0, 88-6), depending on whether the system managed information itself or simply the movement of information through a series of tasks where users manage the content. This is a useful and increasingly apparent distinction, but another framework classifies groupware by its "center:" Does control lie with the individual user or does it reside with the work object itself? Or is the system process-centered, focused on a task that may involve a variety of users and work objects, and that has defined states from start to conclusion (just like a transaction)?

This distinction is orthogonal to whether groupware is information- or workflow-oriented (although there's necessarily a workflow aspect to a process-centered system). For example, the e-mail applications described below (page 11), with their user agents, are user-centered, and can manage both information or workflows depending on the rule sets built by their users. Interleaf's Active Document technology is best-suited for creating work-object groupware, with intelligence residing in the documents. So is Sidekick, for even though the information may be kept on servers, it is managed by individuals. Some workflow tools are halfway in-between; although they create complete tasks, they do so by moving a work object through a sequence of steps and branches without a conceptual model of task completeness.

The differences are subtle but deep. User-centered is managed locally, by, say, a mail client application or a tool such as Agenda; work-centered makes the work an active object, with its own rules attached; process-centered sees the work domain as a whole, and manages work from end to end as a single, complex transaction from a central vantage (virtual or physical). Work-centered is better for sending work outside the system and recapturing it, while process-centered is better suited for querying and monitoring. (Where are bottlenecks? Who has the Borkovsky report? How busy is Sharon?)

In any of these cases, a single "unit" may contain many others: A user's mail-handler may trigger a series of mail-handling processes or rules (every Monday morning, run these three processes); a work document may contain a number of subsidiary documents, each with its own behavior and rules (such as a folder of forms to be filled in by various people with the results to be reconciled at the end); a single broad task (approving a loan, say) may contain many subsidiary tasks (credit check, collateral check, officer approval, and so forth). And in the end, there's a sort of hierarchy, with a complete task perhaps incorporating or interacting with work objects and user agents.

Although you can take any of these approaches to building a system, the process of building it (and modifying it thereafter) will vary markedly, as will the flavor and the perspicuity of the resulting system. The underlying technology may vary, but results are generally best when the technical architecture corresponds to the user model (i.e. mail for user-centered, objects for work-centered, database transactions for process-centered).

User-centered

Let's start with user-centered. This is where the user builds his own agent -- something as simple as a macro or some calendar rules, or as complex as
an expert system to execute rules he devises for interacting with other
group members and data. The system he designs sees him as the center, and
everything else as the outside world. He receives data and requests (com-
mands) from the outside, and sends data, responses and requests back. He
doesn't necessarily know much about what's out there; and even if he has a
mental model of the group or workflow, it's probably not in his software.
The tools he uses range from macro languages and "recorders" to higher-level
packages such as Agenda or Beyond's and Agility's mail tools. He may also
build views for himself in Lotus Notes (or he may use views built for him by
a Notes administrator). He may be managing either workflow, delegating
tasks, setting deadlines, moving work around -- or the content of the work,
sorting information by topic or automatically updating a report.

Work-centered

Next is work- or object-centered. The archetype here is the document that
knows how to mail itself, display itself, update itself from other sources.
Here the user writes instructions that follow the work around: Go here, go
there, get these approvals, come back. The work may even send itself out of
the system and rely on someone to send it back. The problem is the closure:
What happens if the document wanders around and gets lost? Who tracks it
down? This approach doesn't offer a high-level representation of the cycle
of work to be completed, but depends instead on a model in the user's or
programmer's mind. Validation of work completion depends on the users rath-
er than the system; it's as if you could make a withdrawal from the bank
without the system requiring you to debit your account.

Process-centered

The process-centered approach makes sure that the work is completed, treat-
ing it as a complex, possibly nested transaction. Its model of the domain
includes users, data (files) and applications, the cycle of work and the
state of the transaction -- what's done, what's undone, who has what, what's
next? (In work-centered, the object knows its own state, but not the state
of the overall process; by querying broadly enough, you can get that infor-
mation, but it's not integral to the system. It's like a flat file versus a
relational database, if you like.) If user-centered has a user agent and
work-centered has work "agents," then process-centered is closer to a group
agent, working on behalf of and conscious of the entire group. Technically
too, it's a more global system (perhaps physically implemented by a distrib-
uted database), typically managed by a database that handles both informa-
tion and work transactions.

The distinctions between work-centered and process-centered are subtle, but
the first focuses on the work steps, and the second on the work as a cycle.
In the end, it's an issue of the conceptual level of the task-building tool:
is it constructing a series of steps and branches, or is it designing a
coherent task?

A subset or refinement of process-centered is the approach pioneered by Ac-
tion Technologies, which focuses on the relationships between requesters and
doers of the work and "speech acts" such as delegation, commitment, counter-
offers and so forth. The cycle (or transaction) is initiated by a customer
or requester, and concluded when that customer is satisfied -- an excellent
model for business overall. More on this when Action makes its strategies
clear in the next few months.

Release 1.0  30 November 1990
There have been a variety of interpretations of the message promulgated at Comdex by Microsoft -- delivered by Bill Gates in person, but very much a corporate message. As the press saw it, it's about viewing the world as a document rather than as files. Political analysts see it as an attempt to forestall HP NewWave, NeXT, Patriot Partners and GO and their followers. That's true too, in part. But it's also about object linking and embedding. A database lets you reuse data in (and maintain data integrity across) multiple database applications; object-oriented systems allow you to do the same with objects -- that is, reuse not just the data but also the applications (functions) and structure directly associated with the data. More importantly, objects can keep their character across applications; that is, a person is always a person (with name, address, discount level, salary grade or credit rating to be called as necessary), regardless of the specific application and rules applying to that person as a customer, employee, health-plan subscriber, or addressee of a letter.

Microsoft showed us the world-as-document (cf. Interleaf's Active Documents, Release 1.0, 90-3). But not everyone wants to use the document as interface to everything, and the concepts limned by Gates are much broader than the demos. Some people may organize their lives around time: Everything relates to a meeting or an appointment (see Release 1.0, 90-6). Architects or engineers may see the world as items organized by a hierarchy of diagrams; a doctor may see the world as patient charts (a specialized kind of document), while her medical assistant sees it as a schedule. One value of object linking and embedding is that you can get to any data from the medium of your favorite application/environment. (See Release 1.0, 90-3, 90-9.) The killer app is dead; the killer library of objects is upon us.

The open get opener

Gates asserted that this won't be doable for another few years. Other vendors take exception, rightly pointing out that they can do this or that or the other right now. True, but can they do it with any other vendor's applications? They can do it only within their own proprietary, inward-looking environments, with imperfect connections to the outside world.

That's where Microsoft has an almost unassailable advantage: Do it our way, and you're open. Do it any other way, and you're closed (even with the interprocess communication tools from Userland for the Mac, also promised for System 7; Release 1.0, 89-4). The problem is that the Microsoft way is incremental and is less likely to foster fundamentally new applications than a revolutionary approach. But the revolution may never happen if people are kept content enough with the gradually evolving, gradually improving DOS. (In fact, DOS will almost disappear under the increasing functionality offered by Windows, and will become almost indistinguishable from OS/2.) The business fact of market dominance leads to the technical fact that Microsoft's approach will garner the most users and thus become practically, tangibly superior because of its interoperability with the rest of the world.

Until we have a standard way of loading just the functions that are needed (a transaction or two, say, or just the print routines) it won't be of much practical use. (Early efforts in that direction include Borland's VROOMM, for Virtual Real-time Object-Oriented Memory Manager, Release 1.0, 89-6.)
Both Microsoft and NewWave are still wrestling with this object granularity problem, as well as with issues of maintaining integrity across multiple systems, what you do when an "object's" application happens not to be resident on the proper machine, or exists in a different version. Embedding copies the object and makes a new standalone version; linking makes a link to the original, which continues to be updated -- but you have to find it if it has been moved. (DDE is simply a protocol for linking in data and some display information, with no access to an application to manipulate it.) Long-run, integrity maintenance will require an object-oriented database (even if you call it an object-oriented environment) to keep track of all of these. Did you want the data you put there, or the most current version? What if the data in A depend on the data in B and so forth?

Every object a server; every server an object

Client/server devolves into object-oriented programming, where every object is a little server, doing something in response to the messages it receives. Long-run it's not practical (for memory and response-time reasons) to have entire file/application objects, which is how NewWave and Microsoft's Vapor-Wave (Object Linking and Embedding) do it currently. The trick is to call code modules only as needed. An object is a virtual assemblage, with pointers to the appropriate functions and data; each bit is loaded only as needed. And if you call two spreadsheet objects, you don't load the spreadsheet twice. A future system will let users and applications share small modules and data instead of whole applications and monolithic servers.

One move in this direction is stored procedures such as Sybase's, which let multiple applications share defined transaction sequences just as they share data. At the user level, mail tools such as Beyond Mail and Wijit (page 11) mediate among whole applications and let users define and reuse sets of rules for handling various data items, which are encapsulated as mail messages. But they are an extra layer rather than a tool for modularizing existing applications and creating genuine objects. Take the new Sidekick 2.0 (page 16), which can create a quasi-object out of a name and address in its address book (based on Paradox files) and paste it into a meeting in the calendar. But it doesn't "remember" it as an object when we want that person's name and address later. Instead, we select the text and use it for a database query; but if we know two Juan Tigars it doesn't know which one the appointment referred to. In the same way, Agenda creates "object classes" or categories by the simple act of creating columns and values, but those aren't robust enough to carry across for use in other applications (at least not without careful format handshaking).

Early examples of objects include not just the Windows graphical objects everyone is promoting, but more powerful, behavior-rich application objects such as HumanCAD's animated Mannequins, C++-based objects which come both with variable parameters, display capabilities and behavior such as walking, bending and reaching (more on this soon).

All these capabilities are headed in the right way, but as Gates says, if you use a nonstandard approach you end up in trouble later on.... So where else can we look for the standards?
MAIL IN MODULES

Whatever groupware looks like to the end-user, its technical underpinnings usually involve either a database or a file-transfer/mail system, which in turn hide the underlying network plumbing and remote procedure calls from the application developer (and save a lot of work). But the underlying technology also shows through, and e-mail is a natural for the user-centered approach to groupware. User-agent programming tools are ideal for building groupware, using the mail not as an application but as a utility or tool. Meanwhile, the very shape of mail is changing, from an integrated application that hides its architecture, to an open system comprised of clients and servers -- and open to attachment with foreign clients and servers.

Rather than disappearing into the network operating system at one end or the applications at the other, mail is resolving itself into a number of modules. Those modules will indeed turn up inside applications, but not because they’re glued there by developers; rather, they will be facilities easily called from inside applications (cf. Microsoft’s vision, pages 4-5).

Mail basically comes in four parts with a confusing variety of names that reflects the lack of visibility until now of these components. Traditionally, all four have been tightly coupled (quadrupled?) and hard to identify or distinguish and to communicate with from other systems. They are:

- the so-called "user agent," the client application or user environment (including tools to program applications for mail-management and to link the mail system's functions with those of local user applications). The breadth of this facility is in part dependent on the quality of the local application’s APIs.

- the transport mechanism at the back end, which knows how to deal with network operating systems and protocols, and transfer files from one environment to another. It knows about the underlying hardware and plumbing so that applications and users don’t have to.

- the name or directory service, which maintains the physical locations of users who can thus be addressed by logical or virtual names. This can be a single file, a central database, a continuously updated distributed set of files or database (cf. cc:Mail’s new automatic propagation service, for updating local directories).

- the store-and-forward component. The store-and-forward or storage component could be either a file system or a database on the server, or it could be a set of local, distributed mailboxes (awkward if most machines aren’t running most of the time).

In the past, you couldn’t easily get at any component individually, and mail worked best when everyone was using the same system. The interoperability people touted worked through the addition of extra layers, translating back and forth, and so on. A number of vendors have written front-ends to PROFS, which convert its output into something intelligible and easier to use, but they can do little to increase its basic, limited functionality.

Now, however, we’re seeing a sea-change in mail. Just as transfer of documents is about to disappear into a function that can be called from any

Release 1.0 30 November 1990
application, limiting the appeal of traditional e-mail front-ends, a number of companies are starting to build separate programmable mailbox-management tools which go well beyond the display-and-dispose functions of existing e-mail. At the same time, a lot is going on at the store-and-forward end.

For transport, we're still at the stage of hide the plumbing, please, and talk to as many other systems as you can. Transport includes both connections to other hardware, connections to other mail systems (with proper transformations and addressing), and so forth. The transport properly also includes directory services, so that any message can get to any other person, location or service.

The directory server is also a component that can be offered separately, as Banyan is starting to do with its StreetTalk, one of the best directory services available and one of the closest to complying with the forthcoming X.500 standard. Separate from the directory service or name server is the database that holds the actual messages. In the first version, it may simply be a central file system, which holds them until called for, or in serverless mail systems, in and out boxes dedicated to the individual users. In the future (and in systems such as cc:Mail's) it's a database, so that messages can be stored, retrieved and manipulated by as many attributes (fields) as anyone cares to assign to them. Systems based on this model include OracleMail and cc:Mail and, soon, Wijit and Beyond Mail, below.

As mail modularizes, traditional vendors are threatened. Although they offer a complete solution, this is no longer a complete-solution world. Customers want interoperability, best-of-kind functions rather than a jack-of-all-trades complete solution. Third-party developers want something to hook into. Canny mail vendors see these modular third-party offerings as enhancements and support, whereas those reluctant to change see them merely as competition. Clearly, the old mail vendors will be squeezed on both sides, from (network) operating system vendors for transport and servers, and from application vendors for the user agent side. Certainly it requires clever strategy and good execution, but the model of Sybase -- which has opened itself up at both the client and server ends -- is a good one. Sure, people can compete with you at either end; but if you have the best solution what is there to fear? And if you don't, you can forget it anyway. The issue is to make your system if not a standard at least a widely used approach, and then to take a lead in the market you create by leading the way and (gently) controlling the evolution of the standard. (But don't be too successful, or some Open Hole Foundation will attempt to wrest it all away from you.)

cc:Mail -- the sum of the parts is greater than the whole

Perhaps the most visible exemplar of these trends is cc:Mail -- the leader in the pc network mail business with a rapidly rising share now at about 40 percent. As it happens, that still gave it revenues of only $12 million last year and an expected $20 million in 1991 -- consider it a base to grow from. cc:Mail has gone through a fundamental restructuring over the last 18 months, courtesy of new president and ceo Philippe Courtot, brought in from outside the pc business to run the company and rethink the strategy. Fundamentally, from a pc-based application that lets people send mail, cc:Mail has grown into a client-server system that will provide a foundation for people building mail-based tools, utilities and applications.
Of course, cc:Mail should be so lucky as to be perceived as a broad-based threat. Although the company has 750,000 users, or about 40 percent of the LAN-based e-mail market of 2 million pc users (figures from Dataquest via cc:Mail), it has a smaller fraction of all private e-mail installations, many of which are connected through mainframes or minis by systems such as PROFS (2.5 million users) or DEC's VMSMail or All-in-1 (about another 2.5 million). cc:Mail has just started shipping X.400 support to link it to these other systems, and can send and receive messages via Novell/ATI's MHS message-handling service. As it competes and cooperates with the rest of the market, its modularity will be a great advantage in garnering the support of a variety of third-party vendors, from Banyan at the back end with StreetTalk to the new-style user agent tools described below.

"Our goal is both clients and servers right now," says Courtot. "But eventually the network people will do the servers, so we'll concentrate our efforts on the client long-run." In fact, more and more intelligence will reside in the server, following the example (again) of Sybase's stored procedures. Why not let everyone share the rule sets? But call the server development tool a client, and we agree with Courtot. The server will belong to the network; the clients and their applications (whether in the server or owned locally) will belong to user or group agents. The programming tools will come from firms such as Agility and Beyond -- and cc:Mail, of course.

**MHS in flux**

Fundamental changes are also going on with Novell and Action Technologies' Message-Handling Service. Neither company will discuss them until the issues are resolved, but they involve (again) modularization of the product and shifts in emphasis on the parts of both vendors. Originally developed by Action Technologies as a simple, low-level mail-transport utility for its The Coordinator mail/groupware tool (see Release 1.0, 86-10), MHS was licensed non-exclusively to Novell, which offered it free on request with NetWare. Application developers can use it to give file-transfer capability to their applications, and a number of mail vendors such as Da Vinci use it as the transport for their mail applications. Now the Action version of the utility/protocol has been enhanced, specifically to include some store-and-forward and "fan-out" (replication of message) capabilities -- in essence, a low-cost mail server. But since MHS itself is more of a protocol or medium than a product, it's hard to sell effectively standalone, although Action is now also moving it to other platforms.

We're not sure yet what's happening at either company, but we suspect that Action wants to focus on business process management (i.e. groupware) rather than e-mail per se, whereas Novell probably wants to offer a full-fledged mail-server (directory plus storage) as a bundled component in its network operating system. That would allow each company to focus more on its own value-added. Both companies no doubt will continue to support existing products and customers, but such joint ownerships always seem to result in confusion (cf. Microsoft/IBM) when the two owners discover different priorities and the old relationship no longer works. In this case, it's more a matter of sorting things out than of fighting for the spoils, since the two companies are becoming less rather than more competitive with each other.

*Release 1.0* 30 November 1990
MAIL-ENABLED APPLICATIONS

You can build groupware systems relatively easily from the ground up, with modules that talk to each other -- the model used by most group schedulers and other networked office-automation systems sold as groupware, such as WordPerfect Office and INTO, and Borland's Sidekick, below. Some of these include mail as one of their tools, with extensive features for fax, address lists, and the like. But these are mostly sealed systems; they deal intelligently only with their own data, and they load users' applications as foreign objects. Also, they rarely have any notion of workflow or tasks. The calendar is shared information, not obligations among users; the to-do list is a list, not a queue of tasks that the system knows about. Moreover, these systems have mostly wired-in functions are aren't really programmable; there's group data, but not a great deal of group intelligence.

For the moment, then, the most appealing approach is to start with user-centered groupware, and let the user supply the intelligence and applications -- or programmed user agents -- via an e-mail front-end. After all, what better application-independent way to communicate with other applications, whether locally, across a network or even across a country? (At least until we have the kind of global object-oriented, message-passing system described on pages 6 to 7.) Moreover, the e-mail system takes care of time delays, so that not everything need happen in real-time, and events can be programmed to happen at certain times or triggered by certain events.

Moreover, mail (inadvertently) provides some of the abstractions needed for groupware. You can use mail not just to send messages to people, but to send commands and data to applications and to chain these processes. The implementation requires a couple of capabilities: A tool for users that makes the messages easy to construct, and something that translates those messages into something the applications at the other end can interpret. But the translation-for-applications facility depends in large part on the applications in question, and in the existence or quality of their application programming interfaces (API) and macro languages.

The mailman cometh

Two different approaches will be launched this spring, both offshoots of the MIT Information Lens project (see Release 1.0, 86-10). That project was LISP-based and ran on proprietary Xerox D-machines. It provided both information filters, user profiles and the ability to build and execute rules. A follow-on project, Object Lens, added some more object-oriented concepts, but both systems remain research projects without industrial-strength data management and other underpinnings. Information Lens is in the public domain; there are some patents applied for by MIT on certain aspects of the Object Lens system, but its licensing policies seem fairly unrestrictive.

Both new companies were described in Release 1.0, 89-10, and have yet to release their products. Although they share many concepts and features, they are addressing different market segments with subtly different positions. One company, Beyond Inc., has acquired (nonexclusive) rights to the MIT patents, and is run by Chuck Digate, formerly senior vp of "analytic products" (spreadsheets) at Lotus; Soft-Switch's Michael Zisman (who wrote his thesis on groupware 13 years ago) sits on the board. The other, Agility Systems, has MIT project leader (and professor) Tom Malone as a director,

Release 1.0

30 November 1990
and is run by John Landry, previously a senior executive at McCormack & Dodge, Distribution Management Systems and Cullinet.

Both handle the world in terms of mail messages, which they can filter, sort and manage according to information in the header fields or a full-text search of the message body. Both also allow the creation of custom header fields and default values, or semi-structured messages, and can customize rules to handle them. You can also use either of them to delegate mail not just to other people but to other times -- a tickler file, in essence. Consider them another way to manage a to-do list or a project, with messages spitting out at appropriate times to all your correspondents.

Thus, both can create database applications acting as client to a server -- for example, processing an expense report or a subscription request. They can build rules to print out an invoice, send a welcoming (boilerplate) letter to the new subscriber, and finally post the transaction to the corporate database. Sure, you could build these another way, but the mail tools make it potentially within the capabilities of a single user (perhaps with help from a database expert and someone who knows how to work mail-merge in wp.)

Beyond is focused on the local user, allowing him to develop rules to deal with his mail -- call it inbox management -- and ultimately with applications and the outside world. The goal is both to reduce the information overload on each individual, says Digate, and to broaden the dissemination of information that people do want. The system is designed to work on pc-based local-area networks, although it can of course reach further through gateways. The target user is an individual within a group, dealing with messages from other users who have e-mail but may not have Beyond Mail.

While Beyond is focused on personal interactions, while Wijit is more information-intensive. Agility's target user is an individual dealing with external information, whether from outside databases or corporate or department applications. It starts off more outwardly-focused, allowing the user to go out and select what comes into his mail from external data sources.

Another differentiator is style: Beyond's tool, the aptly named Beyond Mail, will initially use character graphics, while Agility's Wijit is the ultimate in Windows glitz. Beyond has a simple, homespun feeling, addressing the largest possible audience at a low single-user price, while Agility is a more high-end product. Although Wijit offers a variety of templates for various information services, it's more likely that some kind of information center, help desk or MIS person will be involved with its installation (especially for a senior executive), whereas Beyond has more of that I-can-do-it-myself feel. Call Beyond a VW, with its capabilities clear, and Agility a BMW with an electronic dashboard -- all associations intentional.

Extending the group

While we would use Beyond Mail in the office, for interaction with other people, we would be more likely to take Wijit along on the road. It has a dial-up feel to it; it keeps you in touch with the outside world. Ultimately, with the developer's kit for either product and a willing customer base, you could use them for that holy grail, channel marketing. In other words, get your customers up on the system, so they can send orders directly into your system by e-mail. Just build them a form to fill in, and on your

Release 1.0

30 November 1990
end, take the form and convert it into the order format required by your corporate mainframe -- probably not something either your MIS department or your customer really wants the customer to deal with directly. Of course, both systems could be used to build the same application; it's a question of the design center of each product -- which, if the two companies are smart, they will take care to differentiate themselves as much as possible.

BEYOND: BEYOND THE ORG CHART

Beyond Mail is character-based (with Windows 3.0 soon) to work in a wide variety of environments; it won't tax even the smallest system. It has the flavor of a personal tool -- not alarmingly smart, so that the user can feel he is in control. The philosophy is that groups aren't defined by org charts; rather, they're whoever the user chooses to communicate with.

Beyond Mail feels like a typical mail system, with the usual header information for sender and receiver. It can filter the mail by values in those header fields, or search the body text for (combinations) of words. In addition, with a developer's kit that will be available a few months after the product itself next spring, the headers can be expanded with new fields to create smart forms (or semi-structured messages, in MIT parlance). Those forms can be manipulated according to the values in the fields: for example, salespeople could exchange memos about certain customers; support people could exchange information about certain kinds of bugs; users could send and respond to messages about meetings at specific times on specific topics. Or a database, with a little programming, could take such a message as an update, creating new records or modifying existing information.

But even the basic program provides a great deal of power, and can be programmed to handle most messages not just from other Beyond Mail systems (initially a rather small base!) but also from a range of "foreign" systems such as your standard DOS mail systems and also PROFS and others (assuming the presence of a mail server with a gateway). Beyond hasn't yet said what mail systems it will be the user agent for, aside from announced support of Novel MHS -- which de facto means interoperability with Da Vinci, The Coordinator, Framework, WP Office and other MHS applications.

When used alone, Beyond Mail will provide its own mail-server (storage system); otherwise, it will happily use a third party's. That is, Beyond Mail will do it all where necessary, but would rather be complementary than competitive to such powerhouses as cc:Mail. Because of Beyond Mail's ability to parse other systems' messages, a single user can get a lot of value out of Beyond Mail by automating his own mailbox.

Forms by example

When you want to send a message, Beyond Mail's menu goes through an extra step: Which message form do you want -- the standard one, or a special one such as telephone, or a custom one specific to your job or group or company? (Or, if you have the developer's kit, do you want to create a new one? The user can change the names of the fields and the corresponding menu picks, but he can't add fields or default values without the developer's toolkit. While building exotic forms is fun and useful, it requires someone at the other end with Beyond Mail to receive them in executable form. Otherwise the recipient just gets some text appended at the beginning of his message.)
Avoiding the mistake made by Lotus (since corrected) when it launched the initial version of Agenda without any sample apps, the initial, text-based version of Beyond Mail will come with a set of standard forms, including phone message and generic request forms. Others such as expense forms or purchase orders or meeting/scheduling notifications are yet to be determined on the basis of beta tests in progress now.

The rule editor operates with the same simplicity. What kind of message form do you want the rule to handle? You select from the menu. And then how do you program the rules? It's quite simple -- by example. With the rule tool you fill in an empty form to make your "if" statement: "If [the form looks like this]..." You can enter specific values, ranges or Boolean combinations, both for the fields and for the body text. For example, "If any messages with Juan or Alice or Mrs. Haynes in the FROM field, and with a figure bigger than $10,000 in the body, arrive while I'm on vacation...."

For the "then" action, the rule can launch a number of actions, many of them available from a pull-down menu, such as forward, reply, file, delete, move a copy to various folders, run or enable or disable a set of rules (wake up a process), launch an app or run a batch file tickle (a verb). In each case, the system prompts for to whom or into what file, proposing the list of known users or mail folders or DOS files to save to (but not batch files to run). An experienced user, for example, could build a small routine (a user agent) to look up the sender's name in a database, put it into a mail-merge file, and generate a customized letter (not just a standard message with a header) in response. At this point the system's power depends more on the user's facility with his own applications than on Beyond's capabilities. ("Over to you, dBASE!"

The user can build rule sets with names -- for example, Vacation/Reachable, Vacation/Unreachable, Working at Home -- with different rules for forwarding mail and responding to situations. Rule sets can be combined or edited, so that, for example, you can easily change the person to whom you delegate a particular type of mail. You can also nest rule sets, so that the Vacation/Reachable rule set includes the rules for Messages from Alice.

@Mail

Separately, Beyond will announce @Mail, a 1-2-3 add-in which lets users use all the facilities of regular e-mail from within 1-2-3 and Symphony and easily send files to collaborators; it doesn't deal with integrity or updating, however. (Lotus is exploring this issue with a group spreadsheet project.) More interesting is @Mail's ability to use data within a spreadsheet to trigger the generation of an e-mail message and control its content. In essence, it allows you to add mail services to 1-2-3 macros, so that, for example, any value higher $50,000 in cells D10, E10 or F10 could trigger a congratulatory message to the sales manager of the region named in the label at the left. Writing the macro is an effort left to the user.

Like Beyond Mail, @Mail basically works as a client to a number of existing mail servers and transports including Novell MHS, but details have yet to be announced. @Mail will probably be priced as an add-in, while Beyond Mail will have what Digate calls "mid-range pricing -- lower than a spreadsheet more than a typical mail front-end or utility."

Release 1.0 30 November 1990
We figure that's a pretty good deal. Beyond Mail (like Wijit) is a way of giving the user control over the information he gets. It leads him gently into understanding the routines that govern his life -- the people he gets messages from, the things they concern, the times things happen -- and converting that knowledge into rules to handle them routinely. Call it just-in-time automation. Of course, as he gets more sophisticated, or as a group leader starts building customized forms as well as rules, the whole set-up can get quite sophisticated. As Digate says, groups start with individuals.

AGILITY'S WIJIT: CENTRAL INFORMATION AGENT

The biggest mystery about Agility is not so much what its Wijit tool does, but how it will be marketed. Short for With Information Just In Time, it's an e-mail front-end, resident on client or server or both, which can talk to a broad variety of mail and information services. Wijit will offer (either bundled or for an extra amount) front-ends specific to Dow Jones, CompuServe and operates directly as a front-end to AT&T Mail, CompuServe Mail, Dialcom, MCI Mail, MHS, VMSMail, Da Vinci, and All-in-1. It is not a pc e-mail, like Beyond Mail, and cannot run standalone over a network without a network mail system, but it can dial any of the wide-area mail or information services with a modem. "Our design criterion was that you could use it an airport," says the peripatetic Landry.

Consider Wijit a generation or two beyond the automated log-on programs contained in such packages as Microphone and Procomm. It is both easier to program (since many templates for access and filtering are included) and more intelligent in its interaction with the various sources of information. A significant portion of its value-added lies in its custom utilities for dealing with various external information sources, so that they all become similarly structured mail messages that can be handled automatically by a single system. Whereas Beyond Mail receives messages, Wijit also actively goes out and creates messages for its inbox by monitoring external information sources as programmed by the user and assigning keywords to the message headers to make subsequent handling of the inbox easier. (Of course, you could also do this from scratch with Beyond Mail, but it is a central part of Wijit's feature set and raison d'etre.)

Mailmaker

Once it gets the messages created by the user agent with its filters, Wijit can load your favorite stock prices into Excel, update your daily stock file, or sort your news releases into folders you define. Overall, the agent goes out and collects information for you at regular intervals, composes the information into messages and then filters and distributes them for you. Its queries can rely on a service's classifications, or do a full-text search, depending on the nature of the services themselves.

Of course, Agility defines those sources broadly -- not just Dow Jones and other information services, but perhaps your own company's internal account-

---

Cute, huh? It's one of those names that will seem brilliant if the product succeeds. In fact, it's far more memorable than the company's original roundheeled notion of using InfoLens after the original MIT project.

Release 1.0

30 November 1990
ing system, or a fellow employee's spreadsheet (with her permission, of course). With a little help from a savvy programmer, Wijit could also monitor your corporate accounting system, send orders (across a phone line) to your company order-fulfillment department, or file an expense account with the proper authorities. (In a sense, all these tools are a way of making "foreign" applications into servers, or objects.)

Wijit's user programming interface is a set of forms and dialogue boxes (akin to Beyond Mail's), which generally prompt the user through his job of specifying the information or filters or actions he wants. Wijit's programmer interface, where someone designs an interface to another application, is a full-fledged Pascal-like language. However, a user can at least explore the system if not create it, by inspecting the folder of tasks (rules), which are stored in the format of <taskname> applied to <folder> for all items of status <all, new, deleted, read, or user-specific>. By clicking on a particular task, he can see the script or rules it executes, perhaps including other, nested tasks. Here, Agility's support for Windows goes beyond the surface glitz, since it gives Wijit's rules easy access to Windows DDE and Dynamic Link Libraries; in essence, you get the same facilities @Mail offers for 1-2-3, but available to or from any Windows application.

GROUPS IN THE REAL WORLD: BORLAND'S SIDEKICK 2.0

Also deserving mention here is Borland's Sidekick, a programmer's tool which has grown up to become quasi-groupware in line with the sensible notion: "Groups start with individuals." In other words, to organize your individual life, you have to start coordinating with other individuals. Although it is not mail-based but in fact uses Paradox files and a server-based group scheduling capability, the locus of control in Borland's new Sidekick 2.0 stays with the user. In this new release the ASCII tables of yore are long gone; instead there are name & address files that can be printed out, organized, pasted into letters or appointments, multi-user calendaring, and even a pager option (so that you can send an alarm or a text message via pager to someone who's not otherwise connected).

While the first generation of pc software simply assumed that the user was standalone, the second generation (not yet fully implemented) is trying to understand how to get him connected. We've moved from ForComment with disks passed around to ForComment online and Aspects and the Farallon and Mainstay mark-up products. We've gone from file-at-a-time database-sharing to algorithms for protecting integrity by row.

But perfect database integrity assumes that the world is connected; Borland knows that it is not, and has already begun to tackle the next problem -- the asynchronous connection. That is, he's connected sometimes; how do you get him back into line, reconcile his calendar (and other information) with everyone else's? Although Sidekick is not programmable, it implements some clever techniques for doing so which should be of interest to anyone trying to manage group information in an imperfect, intermittently connected world.

Rather than try to do it automatically, Sidekick 2.0 takes the simple, fool-proof approach of pointing out the inconsistencies and letting the user handle them. There are four states for A and B: A has some things B doesn't and B has some things A doesn't; B has everything in A and more; A

Release 1.0
30 November 1990
has everything in B and more; both are the same. The initial goal is to get from the first state to one of the intermediate ones. (The system still can't recognize the same appointment at different times -- e.g. lunch with Juan & Alice -- as being the same event; and of course maybe it isn't. Nor does it, as HandyWidget would, realize that Juan and JT may be the same person. Either way, these are conflicts that have to be reconciled.)

This can be done by day, by week, or by month. The system has simple, appropriate menu picks: add the item from A to B; delete the item; move it; and so forth. You can also add all the A-only items from A to B, and so forth. It's all very simple, easy to understand, and not too automatic. The user is still in control. Longer term, you may want to automatically add all items from A to B, and flag only if there are time conflicts. You can so this, and still flag all the new appointments. All this can be tailored to reflect the way you work with your secretary, or with a travel and a stationary calendar.

Obviously, this is an approach that can be scaled up, gently, to manage the reconciliation of a group schedule on a server with the calendars of group members. This capability is available now; long run, the same functionality, based on manipulation of Paradox files, could easily be moved across to other areas, such as intelligent reconciliation of address books. How often have you changed an address in one place, only to have the old one turn up years later in some other place? (For example, we still get mail for Computer Industry Daily, which died in 1985. It's from people who know better, but their personal calendars never talked to their corporate databases.)

Managing agendas, not actions

In the context of groupware overall, Sidekick is an information manager rather than a workflow tool: The meetings are shared information rather than obligations the system knows about. Meetings are scheduled by posting them tentatively to someone else's (public) schedule, and then confirmed by the second person; then the meeting is listed as confirmed in the proposer's schedule. All this is through the medium of Paradox applications and files rather than the e-mail messages that could achieve the same results in Wijit or Beyond Mail. Of course, you can build your own Paradox applications or call the Paradox engine through DDE to generate custom reports or queries or applications on the data in the schedules or the phone books. However, there is no easy way to extend the tools as you can do with semi-structured messages in Beyond Mail or Wijit. (Sidekick does not have e-mail or an API that would make it easy to manipulate through Beyond Mail or Wijit.)

Sidekick starts in a small, easily understandable domain, but it is the precursor to more. These are the kinds of things you have to resolve first in dealing with group interactions. As we progress, we will find that software consists more and more of agglomerations of special-purpose tasks. The trick will be to add them in an integrated fashion. For now, for example, you can paste someone's name into the calendar, but if you point at the name in a calendar entry, it has to do a full-text search to find it in the address book; it is not linked. This means that you need to do a full-text search the other way to find all your appointments with Fred, or all your meetings in Tucson. That is, you can create objects on the fly, but they don't yet really exist within the system.
THE MARKET FOR INFORMATION MARKETS

Information markets are themselves part of a market: Do you want a wholesale subscription service such as Individual's First!, which manages contracts with the various news services and delivers you the news you want with no muss or fuss? (See Release 1.0, 89-7, 90-2, 90-3.) Or do you prefer to build your queries/search routines yourself, and deal with the different flavor of each service? Wijit tries to make them all look the same, but some are feeds -- where you get and pay for all the news at a bulk rate in real-time and select some -- while others are databases, where you get responses to specific queries, and pay more per use but maybe less overall.

More questions: Do you want to integrate the information with other electronic data? First! offers convenient fax delivery and bulk rates for shared profiles (filters), but many of its large customers switch to e-mail; Desktop Data's NewsEDGE customers have asked for and gotten better integration of news feeds with their application environments. Finally, American Information Exchange (Release 1.0, 90-7), is targeted more to ad hoc, one-time use, where you're looking for a specific piece of information or even a custom piece of work. You tend to pay more, for information you want more intensely, whereas the others are a continuing service. It's the difference among a commuter ticket (First! or NewsEDGE), a car (Wijit or Beyond Mail; you still have to pay for gas...), and a one-time airplane charter (AMIX). The Knowledge Network, below, assiduously avoids the transportation analogies in favor of broadcasting, but suggests the analogy of an Official Airline Guide for frequent flyers -- choose what you want to watch.

C & L'S Knowledge Network

How do you organize large bases of shared information? The natural response is to try to stick it into a database, keep it orderly, and build careful group-centered hierarchies and viewing forms, as with Lotus Notes. But especially when the information is changing rapidly, it may just be too much for a single administrative group, or a single orderly database, to manage. And users may not have the skill or inclination to build Notes views and forms. Better to build a less-automated, self-organizing system that will do it by itself (even if there's a database underneath) -- in short, an information market.

But instead of taking the market notion to its full (cf. American Information Exchange; Release 1.0, 90-7), Coopers & Lybrand and its partner Reach Networks have chosen the quasi-market model of television and media -- where success is measured in ratings -- rather than the more common metaphors of information highways, libraries or utilities or information refineries. (Those ratings translate into dollars of resources and funding eventually, but the individual users don't pay from their own pockets. When the system is eventually rolled out to C&L customers, or implemented by other Reach client firms, that may change. Then the model might be pay-per-view...)

Designed and implemented by Reach Networks, a 10-person firm in New York City, the system is called the Knowledge Network and currently has 1700 users, or about 40 percent of C&L's 4000-odd partners and managers. That amounts to 90 percent of those who have computers in their offices, and about 600 different users log on each day.
The assumption was that people in the same practice would have similar interests, and would want to "read" a "channel" devoted to that practice. People within a practice take on the responsibility of producing weekly "shows" (news reports), with financial support from headquarters for outside writers. They promise the firm guaranteed subscriber levels in return for the sponsorship (just as the networks do to their sponsors). Shows with poor ratings ultimately get canceled, as has happened a couple of times at C&L, to such shows as a weekly report on government economic releases.

The initial pilot went quite well, to an audience of 100 information/communications partners, using a UNIX box as a server and pcs as clients. The system now runs on Sun servers, and still uses pc clients. In fact, the clients aren't even Windows-based; they use character graphics, with menus and dialogue boxes that make the system friendly and unintimidating rather than glitzy and high-tech. The technology is quite simple and practical: Like a mail system, it hides the complexities of communications, data transfer and remote procedure calls from users, requiring them only to select items and commands from the screen, and to post data simply by selecting it or by naming a DOS file. Those who use it picked it up with no training.

Good demographics!

But to make the easy-to-use point with ratings rather than words, Zimmerman cites some figures: The audience is partners and managers, not computer folks, he notes, and 25 percent of them are over 50 years old. For the firm as a whole, the average age of a partners is 44, and so is the average age of the Knowledge Network users. A third of the partners in tax and computer auditing say it has played an important role in generating revenues (in response to an electronic survey administered through the Network, of course). Forty percent of viewers of a series intended to encourage cross-selling of services from consulting to auditing and tax, said that the series had been effective. Roughly half of the viewers of an outside tax news service said they had used it to solve a client problem or keep a client informed.

As noted, Zimmerman has all the interest of an obsessive tv executive in the ratings -- which amount to that elusive goal of technology executives,
measurement of the benefits. How many people watched? How many people pressed F2 (to get further information)? Users can vote after they've finished reading a show. Topics range from the relevant ("Tax issues arising from the budget compromise") to the riveting ("Considerations when preparing a partnership tax return"). But we prefer anecdotes: The deputy chairman who first backed the project retired this year, says Zimmerman. He is now 62, and he still uses the system from his home.

Media, not multi-media

Although the structure is media and the approach is show-biz, the Knowledge Network is focused on content rather than multi-media. The system comprises a large text base, but users generate the text and decide what they want to see. In other words, the company left most of the structure and selection of information up to the users, rather than trying to automate the construction of a possibly too-rigid information model. Users simply choose what they want to see from a menu, just as television viewers do.

Users are encouraged to produce shows -- which are up-to-the-week or up-to-the-day reports classified by practice area, written by outside writers for "clarity, wit and bubbliness." Rather than the user doing a text search, the producers of each channel set the agenda, determining what it's important for subscribers to know -- much as a newspaper shapes its coverage and highlights stories on the front page, or a network leads its nightly news (usually with a story with good visuals); see Release 1.0, 90-2.

The information is presented as an electronic slide show of text (10 slides would be typical), which is easier to read than scrolling. There are four major services: Focus, customized shows for various practice areas; Trades, reprints (with royalties) and abstracts from outside publishers such as the Bureau of National Affairs and the Harvard Business Review; news feeds from UPI, PR News, DowVision and Reuters; Discuss, issue-oriented bulletin boards; and Mail, for direct, private communication with other users or outside the system to customers (via e-mail, fax, or printer). You can't annotate items in Focus or Trades directly, but you can copy them and send them out through Discuss or Mail with your comments. The REACH technology also provides electronic forms within shows for such tasks as ordering a full-text printout or responding to surveys.

Viewers are profiled demographically -- in C&L's case by region, practice, job title, task-force assignments, etc. REACH maintains a matrix of who should see what, and then manages notification whenever a new show is aired, so that each user can see what is available that he hasn't yet read. It's his option to read it, delete it, or store it for later (re)viewing. This allows the system to compile ratings -- although just as with TV ratings, you can't determine the degree of someone's interest or attention.

Behind the scenes

REACH uses two databases. The regular system information -- user profiles, usage reports, and pointers to and information about the text base itself -- is kept in Informix, with the documents pointed to by Informix. The system also holds a large, in-memory "database" on the server (using vectors and such) so that it knows in real-time each user's filters and what he has and hasn't seen (just as phone companies keep forwarding numbers and other such
data in memory; cf. Nucleus, 88-1). As a user pages through his system, his menu immediately displays the appropriate selection of shows, properly tagged for whether or not he has seen them. The extra dot next to "Solutions..." (across) indicates that the user hasn't yet seen this show.

Reach's technology includes a News Engine, which classifies news stories along the lines of NewsEDGE, First! or Third Eye (see Release 1.0, 88-1). Reach chose to build its own indexing and comparison engine, although it uses the same similarity-ranking approach based on the classic work of Cornell's Gerard Salton as the others (except Verity, which is based on topic hierarchies, and Reuters' classification scheme, which uses rules). However, says Stumm, "We don't see the value of going the last mile; we do just enough work to get useful classifications, not perfect ones [if there could be such a thing]. At Cornell they had one graduate student too many."

Notes on market positioning

In terms of what it can accomplish, REACH's closest competitor is certainly Lotus Notes (although technically it may be closer to CompuServe). From a market/economic perspective, both make sense only for large groups, unlike the mail tools assessed above. But the flavor of Notes is quite different from that of Knowledge Network. "Groupware is introverted; we're extroverted," says C&L's Zimmerman. "We're trying to reach out across practice barriers within the firm, and to clients outside the firm."

Although there is ample peer-to-peer information-sharing, REACH has the feel of television, with shows professionally produced, whereas Notes is closer to a bulletin board, with facilities for annotation throughout, and with a more granular structure to its information, with tables and outlines. Notes is more of an information management tool, with facilities for application development, while the Knowledge Network is really a publishing medium, with the information management mostly left to broadcasters (and the financial controls to their sponsors). Notes probably works best with tight groups focused on specific tasks, even though the most visible placement has certainly been the approaching-10,000 installation at C&L competitor Price Waterhouse. (Pricing is more like a service, $50 to $200 per person per year for large groups, rather than a software package such as Notes, which costs about $300 per person, starting at $62,500 for 200 units.)

In the long run, Reach's aim is to let its customers broadcast to their clients as well. While that's the aim of Notes too, we see REACH as a
marketing tool, whereas Notes would be a collaboration tool, where companies work together on a project. In the end, as with Wijit vs. Beyond Mail, you could more or less create either system with the other (although Notes is certainly more programmable), but each has its own flavor.

Over all, the REACH system isn’t technically grandiose (although there’s been a lot of attention to plumbing, as with Notes). It doesn’t organize the information in any grand scheme (hierarchical or object-oriented); it doesn’t use natural language or message parsing or fancy tree structures. But it works. Ad hoc or no, it gets information out to people who can use it; it’s more important that the information be seen than that it be perfectly organized. Users find out more than they might otherwise, and they communicate it to clients and colleagues.

"Imagine what the television industry would have been like had home video developed before network television. The family TV set would have VCR and camera attachments but no link to the outside world. It would remain for a future generation of Sarnoffs, Paleys and Turners to come up with the idea of a network that turns these free-standing boxes into a vital communications medium.... There are now over 40 million desktop computers in the US alone, and most of them are not connected to anything beyond the office down the hall."

-- the Reach Networks brochure
RESOURCES & PHONE NUMBERS

Tom White, Action Technologies, (415) 521-6190
John Landry, Agility, (617) 899-5549
Chuck Digate, Beyond Inc., (617) 621-0905
Glenn Weyhausen, Borland, (408) 438-8400
Richard Schwartz, Borland (France), 33 (1) 42.25.28.81
Philippe Courtot, cc:Mail, (415) 961-8800 x 110
Andy Zimmerman, Coopers & Lybrand, (212) 903-8881
Norm Block, Evernet, (213) 544-2242
Ken Goodman, HumanCAD, (516) 753-3550
Ralph Wagner, Keyfile, (603) 883-3800
Larry Moore, Lotus, (617) 577-8500
Christian Stumm, Doug Kester, Reach Networks, (212) 686-0929
Mike Zisman, Soft-Switch, (215) 640-9600
Kamran Kheirolomoom, ViewStar, (415) 841-865

COMING SOON

- Echelon.
- Axon.
- High-tech resume-management.
- Network navigation.
- Machine-assisted translation.
- The Douglas brothers -- Hofstadter and Lenat.
- Upgrades: Aftermarket comes into its own.
- The coming information crash.
- Application servers.
- And much more... (If you know of any good examples of the categories listed above, please let us know.)

Rel-EAST: UNIX in the Soviet Union.

Release 1.0 is published 12 times a year by EDventure Holdings, 375 Park Avenue, New York, NY 10152; (212) 758-3434, MCI: edyson. It covers exotic software, including CASE, groupware, text management, connectivity, cooperative processing, AI and intellectual property law. Editor & publisher: Esther Dyson; associate publisher: Daphne Kis; circulation & fulfillment manager: Lori Mariani; executive secretary: Denise DuBois; consulting editor and copy chief: William M. Kutik. Copyright 1990, EDventure Holdings Inc. All rights reserved. Subscriptions cost $495 per year, $575 overseas; multiple-copy rates and reprints or bulk purchases are also available. NO MATERIAL IN THIS PUBLICATION MAY BE REPRODUCED WITHOUT WRITTEN PERMISSION.

Release 1.0 30 November 1990
We have begun planning the Fourteenth Annual PC (Platforms for Computing) Forum, which we will hold in Tucson from March 10 to 13 next spring -- two months later (and warmer!) than last year. The theme will be "Beyond the desktop: Networks, notepads and legacies." As a subscriber, you will receive registration materials in mid-December. (If you aren't a subscriber and would like to attend, please contact us. We would be glad to sign you up for both the newsletter and the Forum. If someone else in your company -- not your PR agency or your investor! -- is a subscriber, you may be able to register through that subscription; two registrants per subscription.)

As it happens, each of the title items actually refers to a particular product on the one hand, and a topic area on the other. Networks is Lotus Notes, or information management. Legacies is Sybase Open Server for CICS, or application development and interoperability, and how to achieve them with object-oriented approaches. Notepads is GO, or new operating environments. Our goal is to provide more focus than in previous years by framing the discussion each day around a single product -- with ample input from competing products and alternative approaches, both from competitors and from resellers and users, who give the ultimate verdict.

But rather than a series of speeches about, say, "information management" on Monday, we'll have a centered discussion, a sort of compare-and-contrast that should provoke concrete, lively discussion. Products/approaches considered will include e-mail filters, Reach/Coopers & Lybrand's Knowledge Network, and the American Information Exchange.

On Tuesday, we'll address "legacies" not just as data, but as the applications and business rules and procedures that user companies want to represent (and execute) in software. Who controls them? How can MIS and pc users work together? Is it better to reverse-engineer, or to start over? Does it make sense simply to encapsulate mainframe applications (the Sybase Open Server for CICS approach; Release 1.0, 90-9), or to rebuild from scratch? What's the relation between servers and objects (see page 4)? Is object-oriented programming a solution to software complexity, or another cause of it? For vendors attempting to sell to large customers, these are key questions -- akin to the current fascination with providing connectivity between spreadsheets and mainframe databases, but on a broader, more technically challenging scale. Products and approaches will include object-oriented programming, reverse-engineering tools, and a new transaction-oriented development tool.

Finally, on Wednesday we'll discuss notepads. But just as networks are important not only as hardware but as a medium for information-sharing, so are notepads important not only as mobile computers but as the medium for a new paradigm of computing. They promise the possible emergence of a new operating environment; contenders include GO and Active Book, with desktop competition from Patriot Partners, NeXT and others.

Release 1.0
30 November 1990
Featured speeches will include Bellcore/Wang's Ken Knowlton on visualization of qualitative data such as text, priorities, workflows, data structures and program sequences; Ken Oshman on "Wiring the world from the bottom up;" and Intel's Andy Grove on "Power in platforms -- what for?" In addition, "Bionomics" author Michael Rothschild will provide a new perspective on competition and economic evolution.

Please plan to stay all three days, through the discussion of notepads and new operating environments on Wednesday. The conference will end Wednesday at noon with a speech by Electronic Frontier Foundation co-founder (with Mitch Kapor) and Grateful Dead lyricist John Perry Barlow, who will both close the conference and kick off a limited-space tour of the Biosphere, the experimental self-sufficient ecosystem, on Wednesday afternoon.

GENERAL SESSION SPEAKERS

| John Perry Barlow                  | Electronic Frontier Foundation |
| Bob Epstein                      | Sybase                         |
| Bill Gates                       | Microsoft                      |
| Andy Grove                       | Intel                          |
| Hermann Hauser                   | Active Book                    |
| Philippe Kahn                    | Borland                        |
| Jerry Kaplan                     | GO Corporation                 |
| John Landry                      | Agility Systems                |
| Dave Liddle                      | Metaphor                       |
| Jim Manzi                        | Lotus Development              |
| Dennis McEvoy                    | Cooperative Solutions          |
| Ken Oshman                       | Echelon                        |
| Ray Ozzie                        | Iris                           |
| Vern Raburn                      | Slate                          |
| Michael Rothschild               | Cambridge Meridian             |
| Phil Salin                       | American Information Exchange  |
| Mark Tebbe                       | Lante                          |
| Tom White                        | Action Technologies            |
| Andy Zimmerman                  | Coopers & Lybrand              |
| Mike Zisman                      | Soft•Switch                    |

...and others to be added.

Afternoon company demo sessions will include many of the products mentioned above, as well as others from AT&T (Rhapsody), NCR (Cooperation), Action Technologies (The Coordinator and beyond), Systems in Concert (Calypso), MCC (gIBIS), Coopers & Lybrand/Reach, Artificial Linguistics Inc., Eden Group, Clarity, Echelon, cc:Mail, Beyond, Agility, Individual Inc. (First!), Desktop Data (NewsEDGE), Hunter Systems, Keyfile, GRiD, GO Corporation, Slate Corporation, PanSoft, DB Software, ParaGraph and GRiD.
# Release 1.0 Calendar

**Happy holidays!**

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>January 6-10</td>
<td>Sybase user meeting &amp; training conference - San Francisco.</td>
<td>Call Loretta Taylor, (508) 470-3880 or (800) 767-2356.</td>
</tr>
<tr>
<td>January 18</td>
<td>Massachusetts Computer Software Council's annual meeting - Newton, MA.</td>
<td>Call Joyce Plotkin, (617) 437-0600.</td>
</tr>
<tr>
<td>January 24-25</td>
<td>Software support conference - Atlanta. Sponsored by the Institute for International Research.</td>
<td>Call Audrey Wu, (800) 345-8016 or (212) 826-1260.</td>
</tr>
<tr>
<td>Feb 23-March 1</td>
<td>*Graphicon '91 - Moscow, USSR. Sponsor: USSR Academy of Sciences, with SIGGRAPH.</td>
<td>Call Kathleen Nilles, (312) 644-6610.</td>
</tr>
<tr>
<td>February 24-28</td>
<td>IEEE conference on artificial intelligence applications - Miami Beach, FL.</td>
<td>Call Anne Marie Kelly, (202) 371-0101.</td>
</tr>
<tr>
<td>March 5-7</td>
<td>Windows &amp; OS/2 conference - San Jose. Sponsored by CM Ventures.</td>
<td>Call John Bourgein, (415) 601-5000.</td>
</tr>
<tr>
<td>March 9-12</td>
<td>The fifth computer game developers conference - San Jose.</td>
<td>Call Brenda Laurel, (408) 741-5865.</td>
</tr>
<tr>
<td>March 15-17</td>
<td>Networks for the 90's - Tyngsboro, MA. Sponsored by Wang Institute, Boston U.</td>
<td>Call Andree Fontaine, (508) 649-9731.</td>
</tr>
<tr>
<td>March 17-20</td>
<td>*Software Publishers Association spring symposium - San Francisco.</td>
<td>Call Ken Wasch, (202) 452-1600.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Intended &quot;to build bridges between the villages on the electronic frontier,&quot; this is a multi-ethnic event, with people from the FBI and government agencies as well as public and private lawyers, hackers, free spirits, and intellectual property rightists and leftists (whichever is which). Keynote address by Professor Laurence Tribe of Harvard Law School. Call Jim Warren, (415) 851-7075 or fax (415) 851-2814.</td>
</tr>
</tbody>
</table>

30 November 1990
<table>
<thead>
<tr>
<th>Month</th>
<th>Event Description</th>
<th>Sponsor/Contact Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>March 26-28</td>
<td>Spring symposium series - Stanford University. Sponsored by The American Association for Artificial Intelligence. Call Carol Hamilton, (415) 328-3123.</td>
<td></td>
</tr>
<tr>
<td>May 20-23</td>
<td>Spring Comdex - Atlanta, GA. Sponsored by The Interface Group. Call Elizabeth Moody, (617) 449-6600.</td>
<td></td>
</tr>
<tr>
<td>June 3-7</td>
<td>*Object World - San Francisco. Co-sponsored by The Object Management Group and World Expo Corp. Businesspeople's answer to OOPSLA. Call Dave Bradway, (508) 820-8123.</td>
<td></td>
</tr>
<tr>
<td>June 17-21</td>
<td>*International Computer Forum - Moscow. Call Levon Amdilyan, 7 (095) 921-09-02, or Esther Dyson at 1 (212) 758-3434.</td>
<td></td>
</tr>
<tr>
<td>June 25-27</td>
<td>PC Expo - New York City. Sponsored by H.A. Bruno. Call Stephen Feher, (201) 569-8542 or (800) 444-EXPO.</td>
<td></td>
</tr>
<tr>
<td>July 14-19</td>
<td>*AAAAI conference - Anaheim. Sponsored by American Association for Artificial Intelligence. Also includes Innovative Applications of AI. Call Carol Hamilton, (415) 328-3123.</td>
<td></td>
</tr>
<tr>
<td>September 11-14</td>
<td>*Software Publishers Association annual conference - Orlando. Call Ken Wasch, (202) 452-1600.</td>
<td></td>
</tr>
</tbody>
</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.

Release 1.0 30 November 1990
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 rate on request. Satisfaction guaranteed or your money back.

Name ________________________________________________
Title ________________________________________________
Company ______________________________________________
Address ______________________________________________
City __________________________ State _______ Zip _______
Telephone ____________________________________________

How did you hear about Release 1.0? ____________________________

☐ Payment enclosed.
☐ American Express #________________________ Expires _______
Signature __________________________________________

☐ Please send me additional information on your multiple-copy rate.

Please fill in the information above and send to:

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

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

Daphne Kis
Associate Publisher