Meta-mail: A Medium for Meaning

BY ESTHER DYSON

First story: My inbox is overflowing. I have 3158 messages in it, dating back to the last general cleanup, January 2004. I also have a folder called Memorial Day, which contains 1825 messages dating back to spring of 2002 and before: These are all the messages I was planning to handle over the 2002 Memorial Day weekend, but never got around to. They still sit there, a reminder of human frailty. I know that I can find them, even with Eudora’s relatively slow search. But I want to know more: Which ones of the 3158 new ones should I be paying attention to and looking for?

Second story: A couple of weeks ago, analysts following Omnicom Group noted that the company plans to spend an extra $50 to 60 million in audit fees and internal costs (mostly IT, we assume) to comply with the new Sarbanes-Oxley requirements. Presumably it has all the data, but now it needs to make the processes explicit: Who’s in the chain of command? Who made the decision to pay the bill? It wasn’t made by the programmer who wrote the code. (Vendors, take notice! This could be bigger than Y2k!)

Two stories, one theme: Getting control of business processes, not business data.

Indeed, data is relatively easy, and we have good tools for it: the calculator, the spreadsheet, and the giant financial number-crunching application. The spreadsheet gave users a tool not just to calculate, but to build complex models and, in fact, to do many things that previously could be done only by IT high priests. Better yet, the spreadsheet allowed them to build models that were intelligible to normal people. So-called power users could build the models, while
other users could reuse or modify them, plugging in their own data and coefficients. Complementary graphing and other tools made the data more visible and meaningful to ordinary people who could not pick trends out of a sea of numbers. We also have the database, which acts as a back-end to those corporate applications and to the spreadsheets, allowing for easier sharing of data across applications and even among enterprises.

The first successful spreadsheet was called VisiCalc; where is VisiProcess? We have the equivalent of the calculator for business interaction: It’s e-mail (though the analogy does not hold perfectly). And we have giant, corporate workflow applications, along with business-process design tools, project-planning languages and the like. But we have no spreadsheet for process: a lightweight, user-friendly tool that lets some people design processes and other people understand and reuse them. Nor do we have a back-end metadatabase that can store and manage the details of all those processes. (Note however that processes aren’t like numbers; they are more like proteins, which fold and unfold and interact according to genetic instructions. More on this starting on page 25.)

But that’s beginning to change. The dream of defining and implementing business processes is an old one: Henry Ford was noted not just for cars but for the concept of the assembly line; later that century management theorists studied operations research and academicians studied computer-supported collaborative work (CSCW). CSCW focused on computer scientists collaborating in the lab. We wrote about it extensively in the late 80s and early 90s, covering many of the people in this issue (Irene Greif, Mitch Kapor, Tom Malone, Ray Ozzie, Chuck Digate), but now CSCW has lost its academic name and can be applied in the real world. Business is being conducted online: Decisions are being communicated by e-mail; corporate processes are carried out or delegated to others via user dialogues with corporate applications. Thus, the metadata to manage and record business processes can be created electronically as part of the activity. (Everyone who has ever collected expense receipts knows how painful it is to collect and record data after the fact. Now who was that lunch with?)
All this online activity creates both opportunity and urgency. On one side, users overloaded with e-mail and other incoming data are trying to manage their inboxes, where many messages and files are proxies for appointments, commitments, to-dos, requests and other business activities. On the other, corporations responding to the requirements of Sarbanes-Oxley in particular and corporate governance in general are imposing new demands for auditability and visibility on the business processes carried out by and through their IT systems.

This issue highlights some of the tools and applications that are beginning to appear in this middle space, where users are finally getting tools to define and control processes for themselves, for teams and even in part for enterprises. There’s no single concept quite as compelling as the spreadsheet, but a collection of tools and approaches that will serve to make business processes easier to model and manage. Underneath it all is metadata – information about the processes – as opposed to data, which is what the processes often manage. And off in the future is Longhorn, Microsoft’s database-oriented operating and file system that should do for metadata and processes what databases do for data.

To get widespread adoption, these tools must integrate with e-mail and other ad hoc communications functions. (Only involuntary users – i.e., people tied to a single well-defined application – will suffer the constraints of an isolated system and stay within its boundaries.) For teams, it’s new tools for tracking interaction and defining work status. And for corporations, it’s a new programming approach, treating processes like objects, as easy to modify as spreadsheet formulas even as a background system keeps track of the modifications and keeps the meaning intact. There are also functions that will make it easier to detect, generate and manage metadata, lowering the barriers to adoption – too much work for too little immediate return.

Mail is indeed fundamental, since it implements the vast majority of human interactions online, although other media such as instant messaging, blogs, RSS feeds and the like are crowding in. But there are other antecedents too, most notably the PIMs (for Personal Information Managers) of the past.

Some members of the team that launched Lotus 1-2-3 later developed Agenda, a cross between a spreadsheet and a database that was one of the early PIMs. (Agenda itself was an evolutionary dead-end, even though it presaged much of what’s happening today, and its development manager, Rich Landsman, has re-surfaced at Mailblocks, page 35.) It helped users define and manage tasks and it had strong cal-
endar features, but it and the other PIMs were too personal: They had little or no
communications capabilities (like today’s PDAs, now becoming smart phones).

Then of course there was Lotus Notes, both structured and unstructured, but isolat-
ed in its own environment apart from the Windows environment that was becoming
standard even then. It was a roach motel: You could put things in, and they would
never come out. To get its benefits – though they were many – you had to stay inside.
But Notes has continued to evolve, becoming more open and garnering 100-million-
plus users and a thriving development community. IBM has continued to extend the
product and its Notes-supporting/absorbing Workplace Client Technology (see page
15) will be one of the key players in the new process-management world.

IBM’s and others’ user-centric process-management tools may become so user-
definable as to be almost indistinguishable from one another, but in fact they will be
made up of specialized, highly engineered components that interoperate, and that
reflect defaults and templates defined by the vendor. They must also support cooper-
ation and task management for multiple users and most applications, since almost
every process occurs between people or at least is asked for by or delivered to some-
one else. Technically it may come from somewhere else, but from the user perspec-
tive, most of what we describe below will feel like an extension of mail: time-
sensitive and alert/attention-driven, focused on human interaction, but with new
kinds of input and new kinds of structure and visualization. So, for want of a better
word, we’ll call the tool suite Meta-mail.

**Meta-mail: A medium for meaning**

Like a spreadsheet’s formulas, Meta-mail’s process elements will be user-definable,
inspectable, reusable and above all personal. Meta-mail will operate at any scale
from a single formula (organizing a two-person meeting, say) to a large shared
model of next quarter’s project launch or an operating division. It will have both
power users and casual users: people who build models and processes, and those
who use them. It will support people who simply want to manage their own tasks
better, and people who want to design multi-person processes in which they them-
selves are just one player. And, like a good spreadsheet, it will talk to a variety of data
sources. But it will be more than just a tool, because it will come with a host of
handy functions and features built in and around it.

If a spreadsheet enables a user to manage data, in a user-centric, accessible way
rather than like a database, Meta-mail does the same for metadata and processes.
However, because metadata is harder to define and collect, Meta-mail has to be much smarter than a spreadsheet. Specifically, it has to automate or at least support the easy generation of metadata, or users will never use it. (Some early examples include blog tools that can auto-create RSS feeds and XML labels, and other tools that can detect inherently structured information in plain text.) That has been the downfall of so many organizational/project/workflow tools in the past. Either they are either too rigid and corporate, or they require too much effort to customize or even just implement for too little return...or both.

On the corporate side, the Meta-mail tools will be similar to very high-level languages/application frameworks, all of which can build the same applications, but which nonetheless have different flavors. Power users or developers will be able to build robust, full-featured corporate workflow systems such as new-hire recruitment, asset-tracking, internal audit, employee-benefits management or any other process that involves human activity and decision-making. They also will be able to use modification tools to add their own functionality as well as their own data to existing processes. But the tools themselves will force these new or changed processes to use the system’s grammar, making the processes transparent and auditable. The key point is that the components are business processes rather than computer transactions: If you want to change a process you need to do so within the system. That is, a programmer can’t simply declare an extra payment to a foreign bank account. Any payment process needs to be approved, by an e-mail to some user with a defined role registered within the system, or an exception must be made and noted by someone else within the system. (Of course, a programmer with enough knowledge could get around the system; that’s a bug in reality, not just in Meta-mail.)

Although these process tools will be uniquely flexible, vendors face the danger of making them too flexible – leaving up to the user too many decisions that could be better made as defaults that can easily be changed by the user. What will define market success, then, is the ability of any particular vendor to make the right guesses and set the right defaults for any particular market or function, plus intelligent, easy-to-understand customization and modification tools. All the user should have to do is confirm the system’s guesses.

Are the modification tools a pick list? A shortcut key? All of those, and more. They could also be a set of “scenarios.” That is, rather than make the modifications one by one, a user can simply specify “party” or “offsite” to get different versions of an event-scheduling module, one with selections such as “kids invited” and the other with a link to “agenda.”
**Reality check**

Below, we survey some early examples of Meta-mail tools: Some of them are research projects; some are full-scale applications, or features in them, that are already shipping. Some are full Meta-mail suites, and some have features that could enrich Meta-mail. One well-known brand name is key: IBM’s Workplace Client Technology, which promises to have many of the features we describe here and makes its initial release this month. (One of its capabilities, ironically, would be to parse this sentence to determine which month we mean: OK, it’s July 2004.) Groove, shipping since April 2001, already has an architecture/platform that supports multiple, disconnected users and many Meta-mail features, but lacks the explicit process support of full Meta-mail. We also cover mail tools such as Google’s Gmail, Stata Labs’ Bloomba and Mailblocks because they illuminate some of the features of Meta-mail. On the workflow side, tools range from the lightweight, flexible WikiPad to the still cumbersome, heavy-duty process-manager in beta from Resilient. We have arrayed the companies on the chart below in approximate indication of their positioning, although there is much overlap. Note that there is no “better” or “worse” position on the chart; the goal is simply to be good at what you do. This is not a complete survey, but a sampling. Other companies that could be mentioned include Intraspect/Vignette, Lombardi and ePeople.

At the end, we discuss a couple of approaches to automatic detection and management of calendar and contact data; these are only components of Meta-mail, but
they are the kinds of tools and capabilities that will make it successful, by making the
generation and capture of metadata easier for individuals. Indeed, as an individual
user, you can probably get an immediate ROI on this issue of **Release 1.0** by turning
to page 34, reading about Anagram and downloading it so that you can start gener-
ating your own metadata quickly and easily.

Although what we describe here is still just a vision, there are enough components of
it extant to give it credibility. But we’re early in the game: The products are generally
new or in beta. But they are something we look forward to. It’s not just efficiency.
Whether you’re an individual or a corporation, managing newsletter revisions or
merger negotiations, there’s some emotional, legal and financial security in being
sure you can track everything that’s going on.

**Some distinctions**
Meta-mail uses information and metadata management to support **attention management** for individuals and **process management** for teams or enterprises.
Although we’re still looking at things from the user outwards, the management of
one’s own attention to tasks and of interaction with other people – both through
messaging and by managing shared content – is key.

**Inbox vs. workspace**
Some people think there’s a conflict to resolve: Is this new Meta-mail user environ-
ment an inbox or a set of workspaces? In fact, it has to be both, offering the user a
shift in perspective from overview with alerts and alternatives, to a more focused
view of a single activity (the Manistique trip) or a single communication medium
(just my SMSes or just my RSS feeds).

Of course, each vendor will have a sweet spot: For example, Groove focuses on mul-
tiple shared workspaces, whereas IBM Workplace Client Technology helps a user
assess the sweep of a giant inbox that holds alerts, messages, activities and conversa-
tions, with visualizations that make it easy to explore each of them from a variety of
perspectives. It’s ideal for people with complicated, intermingled, distracted lives,
whereas Groove is targeted to people who work intensely with others on one or
more discrete projects.

**User-centric vs. process-centric**
And finally, there’s the distinction between the user client, which helps the user
manage his attention and commitments and relationships with others, and the
process-based workflow, which generally lives across machines and ties a number of
users into a single process, even though it may have been developed by a single user
for the group. At the extreme, process-centric Meta-mail tools develop and manage
processes that are owned by the enterprise, but Meta-mail will make them easier for
individuals to build as well as more transparent, auditable and modifiable (for better
or worse) than traditional corporate workflow applications. Users will be able to
customize not just their own data but their own processes and metadata; the
processes will be self-aware and addressable by users.

Search's Great: Less Filing

We begin with search precisely because it is not Meta-mail, but it makes it clear what
Meta-mail can bring to the party.

Two recent new mail services – Stata Labs’ Bloomba and Google’s Gmail – are raising
the profile of one feature of Meta-mail: It will use search and (user-specified) attrib-
utes or labels rather than folders as a default way to find things. This search func-
tionality may be subsumed over time by Longhorn, but these two mail clients
provide a hint of how simple and powerful search can be. In short, people will no
longer worry about finding things.

Bloomba: Change your outlook

We first covered Bloomba in March of this year. About a year before Google,
Bloomba launched a product and promoted the idea of search-based mail: Why use
folders when you can just use search? – and labels if you insist on it. Folders have the
limitation that a single document can be stored in only one place (along a retrieval
path, technically), whereas a document can have multiple labels. Of course, the
purists insist that most documents don’t even need labels: Classify things if you
want, but rest assured that even if you don’t you can find them later via pure search.

As noted, the real key to Meta-mail’s content management is not in user-generated
searching, but in customizability, allowing each user to deal with the elements of the
“flow” in his own way. Filtering, for example, is an application of search technology
that can help manage the incoming flow rather than find things thereafter. There’s
also saved searches, similar to filters – a way of defining “folders” that is more com-
plex than a single-level label.
A SHORT, CONTINUING HISTORY OF MAIL

Where we started is an accident of history, but where we will end up is destiny.

Mail began as a simple tool in two parts: the messaging service that moved things (files and messages) from A to B; and the user clients at A and B that received, managed and stored the messages and related files.

In modern times, mail has become ubiquitous, and people are demanding more functionality around it, even as other forms of messaging proliferate. “Mail” tools have added calendar and contact database modules (or at least data import and export tools) to the client, and they are expecting back-end messaging services to catch spam and viruses, control access and more. Marketers are developing “mail” that can take over your mailbox and show you a video, or that can surreptitiously track what you do with the mail that you receive. And while other vendors are promoting RSS (see Release 1.0, July 2003) as the cure for spam, it works more as a way to ensure that periodic, wanted messages or data get through, rather than as a spam deterrent.

Less visible are the more subtle, more fundamental architectural changes that will support Meta-mail, heralded by Groove, wikis and other collaborative tools. (See Release 1.0, April 2003, on social software.) At the high end, various workflow development tools are just beginning to have the flexibility and user modifiability we describe here. Some of them now allow their users to work in their “natural” environment of e-mail.

Now the back-end of e-mail is tending towards being a shared, perhaps distributed cloud in which much of the content is stored; only links, pointers, references, alerts and short messages are actually moved around. From the user point of view, the data feels both local and shared. Although the cloud may feel peer-to-peer to users, it will most likely be run by third-party services or sit on corporate servers, which themselves must integrate seamlessly. All this will require reliable access control, privacy assurances and identity management, along with seamless version control and management and transaction integrity. In this issue, we focus more on the client side, but complementary facilities in the cloud will be necessary. And of course, there’s always the challenge of managing a single user’s data even though he may enter, use or receive it on multiple devices and want different views of it on those devices. We will cover these questions in a future issue of Release 1.0.

Bloomba makes a point of its capabilities for managing subscriptions, whether e-mail lists or RSS feeds. It labels them automatically if the user wants, and then allows the user to specify for each subscription or category separately, how long the messages for that subscription should remain in the inbox before moving to the archive or being deleted. Thus, for example, you could have your Google news alerts stick around for a day before disappearing, while your mother’s blog would last for a week before being archived automatically. You could also do this with regular mail from a particular correspondent. . .if you dare.

Now how about a feature that marks Juan’s mail as URGENT if it’s more than a week old, and moves it to the head of our queue? By using the saved search feature, you could create a label for such urgent mail and keep that saved search on your desktop. This is an example of what we mean by programmability. The features and internal data of the program (such as dates and metrics about the content as well as within it) are exposed in a way that the user can customize the system to his needs without excessive kludginess.

..and more: low-carb conversations!

Finally, Bloomba is developing an enticing new function (planned for sometime next year) it calls “flattening” that parses a message text’s quoting hierarchy, rese-
Says Bloomba founder Raymie Stata: “The problem with mail today is not spam. It’s the flow of legitimate messages. The inbox is used for so many different tasks—announcements, scheduling, collaborative document editing—that users have become inundated and don’t have adequate tools for dealing with that flow.” He neatly describes three main categories of incoming messages. Only one involves processes:

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Spam</strong></td>
<td>Needs detection</td>
</tr>
<tr>
<td><strong>Subscriptions</strong></td>
<td>(read-only – RSS, lists, etc.) – need filing/labeling/threading/filtering...and scanning for important content if possible</td>
</tr>
<tr>
<td><strong>Correspondence</strong></td>
<td>(including to-dos, actions, replies, etc.) – needs attention...and perhaps privacy protection</td>
</tr>
</tbody>
</table>

sequences it chronologically and removes the redundancies. How? First, says Stata, “We track down the thread. On that score, we just do what everybody does. We follow replies and we remove [i.e. hide] ‘bottom quoting’ – the inclusion of an entire previous message at the bottom of a response.” This is not a unique capability, but Stata is committed to making it more intelligent over time, both in avoiding false positives—as when someone simply replies to an old message on an unrelated topic instead of re-entering a correspondent’s address—and in finding occasions when someone initiates a seemingly new message that is actually part of a thread—as when someone can’t find the original message but remembers it and initiates a fresh reply. E.g. “Can’t see you Friday after all.”

Overall, Stata is more optimistic than many mail vendors, perhaps based on his success in getting people to “change their Outlook” in order to try Bloomba. (He had expected that users of “minor brands” such as Eudora would be more likely to move to another brand, but he discovered that users of minor-brand e-mail clients do so on purpose, whereas some significant number of Outlook users do so only by default.) He won’t be specific: “From my perspective the numbers are quite good, but attitudes vary... We were so convinced that Outlook users would be hands-off that we didn’t even have an Outlook importer. We were inundated with people who said, ‘I’d like an alternative, but I really need a calendar.” So now Bloomba has an iCal-supporting calendar...and an importer for Outlook as well as for Outlook Express, Eudora, Netscape and Mozilla.

More significantly, Stata notes, “We design client software under the assumption that the user won’t put in much current effort for the sake of future benefits. But users are going to have to put some effort into managing their activities. So the good news is this: Historically, folders have been the place where users have put in manual effort to keep themselves organized. As search gains hold, we see folders fading. This frees up exactly the right kind of cycles—’I need to get organized’ cycles—to work on one’s
personal workflow. If you’re going to spend 15 minutes a day providing metadata, associating messages with activities or setting deadlines is more useful than filing things into backward-looking topic categories.” In short, a search tool can figure out what the topic was, but only a human can determine what he needs to do about it.

**Gmail: Gee, you’ve got mail!**

Because of Gmail’s role as an instigator of excitement over e-mail, and because of Google’s role in the market overall, we give a little more attention to Gmail than it might merit strictly as a Meta-mail product. In many ways, Gmail is not even a product: It’s closer to a working demo of a concept, which is search-based mail.

The launch of Google’s Gmail, coinciding roughly with news of the company’s IPO, captured the imagination of the press not so much for its one gigabyte of storage and its search capabilities, as for the perceived privacy risks (see box, next page). In fact, Gmail has simply made visible risks that already exist, especially for Web-based mail such as Hotmail and Yahoo! mail, and which have more to do with government policies concerning stored mail than with unique attributes of Gmail. But yes, one uncomfortable fact remains: “Mail” enjoys special privacy protections under US law that do not apply to “stored data,” especially when it is stored by a third party. That is something that will need to be resolved for everyone and especially for all third-party-hosted mail services, not just for Gmail. We will not address it further here.

Commercially, Gmail beat out the other notable Webmail services with its out-of-the-gate offer of a gigabyte of storage. Although we doubt that Gmail is causing any kind of slide in Yahoo!’s or Hotmail’s customer numbers, it is messing up their business model and causing Yahoo! and Hotmail at least to raise the ante on what they offer for free and for fee.

Gmail amply illustrates the benefits of search-based mail. It’s sleek, fast, capacious and Google-like. Moreover, it does let users “label” their mail with an arbitrary number of labels to define subtleties of categorization that can’t be caught by text search/filtering alone. Indeed, Gmail’s topic-catching capabilities are strong. You could say that the ads (or “related pages”) express Google’s view of the topics of the mail. Do your friend Juan’s messages come next to ads for disk drives and mice? Perhaps he needs a broader view of life. Do your loved one’s messages arrive festooned with ads for exotic vacations or fancy restaurants? Maybe you should take the hint.
Searching goes beyond finding documents, however. Gmail, like most client mail tools nowadays but unlike other Webmail, indexes recipient names and auto-completes them as a user types. But it also automatically triggers the creation of a contact database that works for sending mail even if you never fill it in (and presumably it could later serve as a whitelist to filter incoming mail). In Google character, Gmail uses clever algorithms to order the list, based on frequency and last-use metrics. But as a commercial proposition, Gmail is still deservedly in beta. What keeps customers loyal to Yahoo! or Hotmail is not storage space or price; it’s what each user has built at Yahoo! or MSN -personalization and links to other features such as calendar, financial services and the rest of the “MyPortal” offerings. Gmail is sleek and austere, with much of the lightning appeal of Google itself. But we’re not sure mail is supposed to be sleek and austere; it’s supposed to be personal. Studies show that users like Google’s uncluttered interface, but that’s because most other websites are cluttered with the website’s “information,” rather than the user’s information. Users can stand or even like a much higher level of information density if it’s their own information, arranged in a way meaningful to them. (See the rest of this issue, pages 13 to 39.)

Thus Gmail’s sleekness is not necessarily a benefit, as it is for Google the search engine. There are few ways to customize it, beyond labels and the use of stars. It has powerful advanced-search features and a solid capability for threading conversations, but it collects the threads into a single, hard-to-scan conversation rather than displaying them in the elegant way IBM’s Remail does (See page 15).

But there’s little point in criticizing Gmail for not being what it is not trying to be. It’s an elegant example of the power of search.
**Beyond Search Lies Structure**

Right now, there’s a lot of excitement over search in mail. Find anything you want, fast! That’s true, up to a point, and better search will make mail incomparably better. There’s no way that faster and more comprehensive (i.e. across all folders – if you’re still old-fashioned enough to *have* folders, and across all fields and including attachments) can’t be better than slow, limited search.

The use of search rather than folders in mail is part of the general trend away from clumsy file structures and towards document attributes and content rather than (sniff of scorn) an “arbitrary storage structure.” However, vendors can go too far in this direction. If people start using search and filters broadly, they’ll find they have the same old problem of too *many* messages matching, and search rankings that may or may not produce the desired messages first. For some people, this problem never arises. Maybe they don’t *have* a lot of mail, or they just happen to like search and feel comfortable using it to the exclusion of other mechanisms. Moreover, says Groove founder Ray Ozzie: “The death of folders is greatly exaggerated. Our brains like a combination of explicit hierarchical structure *and* metadata tagging *and* free search. Some people are foldering nuts, and some never folder but use some custom views, and everyone uses full-text query.”

And finding things isn’t the key to Meta-mail anyway. Users will still need and want to classify items in order to make sure they re-emerge at the right time. In other words, no, I don’t want to look for that message about an appointment; I want it to pop up and remind me the day before. . .or if I attempt to schedule something else at the same time.

That leads to labels – or user-generated metadata – which allow a user to put things into multiple folders, and to create a hierarchy and metadata that is also searchable. Search doesn’t have to be “pure,” and having the user classify things in a way that the system cannot, whether into a folder or with a “date stamp,” is a way to enrich search rather than an a sign of failure.

In fact, with search *and* metadata users will be able to look for and manage processes, activities, transaction threads and the like by what often matters most: links between events and activities, timing, state of doneness or dueness, attributes that have to do with the state of interactions among objects and people. (In a different context, we covered this tension between search and structure in Release 1.0, January 2003.) It comes down to search as the ability to find a desired set of items, perhaps
ranking them by relevance, versus the ability of other tools and of humans to rank or classify things in some other way and to present a visualization that reveals context or structure in addition to whatever search can do. Moreover, many users surround formal applications with mail and other messages – which corporations now want to capture and structure: “Juan, please fill in this form I have linked to, and by the way, you should know that on line 43, it’s OK if you just enter last week’s date.” Juan may want to find that voice mail later. Or Alice may want to delete it...

In short, text search can catch topics (or nouns, what something’s about), but it can’t catch the implicit “transactions” (or verbs, such as commitments, deadlines and decisions).

**Finding or creating structure**

So how can we find and represent this deeper, active meaning? Some of that can happen automatically with the right tools; other cases will require some user input. Making that user input easy and natural is a design challenge at least partly met by some of the tools we describe below.

Of course, there are different options for structure: There’s metadata around each message or data item – when created, by whom, to whom, and so forth. There’s metadata in the links between messages – threads, filters, all messages from or to someone and the like, and the structures and sequences formed by collections of messages. There’s implicit structured data within messages, such as dates and times and people. And finally, there’s the message “meaning” (the action verbs as opposed to the topic nouns), such as delegate a task, request a meeting, fulfill a commitment, agree to a deadline. A message about picking up an RV might belong to the “summer vacation” activity, even though there is no such indication in the text. Sometimes it’s just up to the user to tell the system.

The result will be information that’s more self-aware and queriable, so you can search – or make the system surface information – not just by content or classifica-
tion, but by any other characteristic: three-day-old mails that haven’t been answered, documents related to a certain project or referring to appointments on a particular date, and so forth. The supporting tools and functions will allow users to visualize the characteristics of messages or the state of any process; will automatically recognize structured data such as contacts, dates and times for easy integration with schedules and contact/identity management; and will use presence management and identity management in order to route work and queries to roles rather than people; and will interoperate with mail on the one hand and with formal workflow and database applications on the other.

**IBM/Lotus: Reinventing mail**

Although Microsoft is the clear winner in mail clients, Lotus Notes is a leader in the high-end enterprise space with more than 100 million users. In many ways it has been ahead of its time, isolated from the mainstream in its own environment, but that is gradually changing. IBM’s offerings in this space are squarely focused on the business market, but we can imagine many of these functions slipping over into consumer products/services over time.

When IBM acquired Lotus in 1995, it acquired Lotus’s leading-edge collaborative-software research team, since integrated into IBM Research. In the next year or two from now, much of Lotus’s work on a project called “Reinventing e-mail: Remail” will find its way into the market, as part of IBM’s Workplace Client Technology. The team has been led by Irene Greif (see **RELEASE 1.0, MARCH 2004**), one of the pioneers of computer-supported cooperative work. Using their own tools, members of the team work in the Almaden, CA and Watson Labs in the US; Haifa, Israel; and Beijing, China; as well as at Lotus’s facilities in Cambridge, MA.

IBM’s first version of this development/operating environment was browser-only, and was not a formal product; it was used by about 500 customers. This summer, IBM will ship to customers an actual revenue product (about $30 per seat) called Workplace Client Technology 2.0, which right now is complementary to Notes. It will eventually replace some of Notes’ functionality, though it will continue to support Notes applications.

Along with Groove (see page 20), Workplace Client Technology is the product closest to realizing the capabilities of Meta-mail. Some of the features we describe below will be available this summer; others won’t come until versions 2.5 and 3.0, next winter and next spring or summer respectively.
The challenge IBM meets is not to manage the user’s content, but to manage her attention. As shown in the Remail “federated inbox” below, that requires a variety of tools, just as a spreadsheet comprises graphs, back-solving and other techniques. (For more detail, you can check an unusually open and comprehensive set of papers on IBM’s work in progress; see Resources, page 39.)

Remail has its own name for the things-formerly-known-as-folders – collections. The point is to distinguish them from folders because they are closer to search labels: They classify a message but don’t necessarily put it away somewhere. Thus, you can have in-sight and out-of-sight collections, and a particular message may be in-sight

Remail offers the viewer a customizable, comprehensive overview of his tasks. Here we see two views of the calendar (1) and the main message inbox at the top center. The messages are default-sorted by date and tagged by origin/medium (mail, SMS, RSS feeds, etc.). One message is highlighted; it is part of a conversation thread which pops up (2), showing senders’ names and the first few words of each message (since the subject line will tend to be the same for all the messages). Over time, users may learn to put key thoughts into their first line. . .though that may be hoping for too much! Below, a single message from the thread is shown (3), along with its place in the message thread – the squiggle (3a) in the corner of that pane. There also is a list of the user’s collections (4), akin to folders, and a message map (5), allowing the user to “see” a large number of messages in the inbox or in a selected collection, highlighted with color and other markings according to user-selected criteria. For example, you can highlight a specific message, and see all the other messages from the same sender.
even though its collection is out-of-sight. That is, you can put a collection out-of-sight, but you may still want one particular message that you need to answer to remain in-sight.

In short, the desktop metaphor was a really handy one when we were just getting started with mail and other desktop functions. But now it has become limiting: Why should virtual messages be subject to the limitations of physical ones? Why can’t a message be part of multiple collections, even though you can’t easily put a single memo into multiple real-world folders? Why can’t your calendar serve as an index to your messages, along with a more conventional hierarchical, subject-based index? Or a map serve as an index to store locations? (see Release 1.0, April 2004.)

The collection of features in Remail is stunning. The simple ones should be out this year, including collections, source-tagging, the message thread arcs and some of the calendaring functionality, as part of Lotus Workplace 2.5. Others will come later.

Activity Explorer
In addition, Remail will support what IBM calls activity threads – a way of focusing the work around an activity such as a project, rather than around a function or application such as mail, chat or for that matter a single spreadsheet. “Right now, collaboration is tool-centric,” says Werner Geyer, IBM Research staff member and inventor of Activity Explorer. “We have e-mail, instant messaging, discussion databases, document repositories, shared workspaces, and so on, but they’re all disconnected. We are trying to make sense of things by organizing everything into an activity. You can begin an activity in the most natural way: You get in touch with someone –through a communication medium such as e-mail or IM – and you share some information – usually a file or a folder.”

The Activity Explorer will be part of WCT 2.0 – but as a “technology preview,” or a beta component of a shipping product suite. With charming diffidence, IBM is offering it to customers with the subtext: “We know it’s imperfect and it’s new, but tell us how you like it and how to improve it.” From the demo we saw, it looks like a winner, because it clarifies rather than adds complexity.

As shown in the screen shot on the next page, an activity may comprise a variety of communications media, a variety of people and a variety of work products (spreadsheets, documents, Powerpoints, pdfs). And of course it could include a formal project plan...but it needn’t.
However, for now the Activity Explorer tracks only “shared messages” – its own version of e-mail – rather than “regular” messages created in a mail system. The major constituents are the people and the work objects involved. The objects could be files such as spreadsheets, Word docs, a data file; folders that contain several such objects; or related conversation threads in – for now – any of three media – mail, “persistent chat” and shared screens. We could imagine adding voice to the mix as well. Other than an implicit “before” and “after” based on the hierarchical structure of the activities, there’s no explicit time component to the activities. Over time, assuming the users like the prototype, it will become better integrated with such functions as mail and calendar, and it will handle version management of the shared objects (which is currently not implemented).

**Message thread arcs**

Two other compelling and useful visualizations in Remail deserve specific mention. In the future, but not in the current (July) release, you will be able to see an e-mail conversation as “message thread arcs,” as shown in the illustration on the next page. IBM has tested them extensively on actual users, but doesn’t feel they are quite ready.
yet (unfortunately!). The idea is fairly simple (but before you ask, we can’t find out its patent status). When you click on a message in the inbox, you see a “thread” view of the e-mail and all the e-mails in that conversation thread. As shown, the thread arc represents each message as a node in a straight, time-ordered sequence, and then draws arcs from node to node representing a message-response link. A message that generates lots of responses results in a bushy thread, whereas a sequence of messages each replying to the previous one in turn is called “narrow.” The nodes can be colored to represent different senders (assuming a small-sized group), or whether messages are answered or unanswered, and so forth.

It’s easy to think of use cases where this approach could get messy: really large conversations, or ones with more correspondents than there are colors. But the whole point is to develop a useful, practical tool that makes it easy to visualize typical work conversations, and here the thread arcs succeed. They are specifically designed not for large, bulletin-board/discussion group-style conversations, but rather collaborative conversations where a small group works on a problem. Note also that these thread arcs show the specific point of view of a single participant; they aren’t displays of public postings (though of course there is some overlap).

**Mapping your inbox**

The correspondents view will look familiar to people who have seen Smart Money’s Map of the Market (see [release](#) 1.0, September 2002). The designer of that visualization, Martin Wattenberg, joined the Remail project in 2002 and immediately applied the same treemap methods to the inbox. It may not be elegant (reflecting life’s own messiness), but it reveals useful information by displaying the messages in any collection, grouped by sender. The senders in turn are grouped by domain, a rough proxy for an institution. (This reflects WCT’s business orientation, for many private users might find most of their correspondents grouped under aol.com, hotmail.com or, some day, gmail.com.) The display lets you see who writes the most, and also analyzes your own responsiveness.
**Groove: Outside the box, out of the box**

“For the first time in seven years,” says Groove’s Ray Ozzie, “I finally feel we have the product we were imagining from the start, supporting personal and team content management and interaction. Groove builds virtual workspaces that bring together people, information and tools to get things done.”

Indeed, it all goes back even farther than seven years, to when Ozzie first thought of what became Lotus Notes back in 1982. Groove extends those original ideas, but over the years Notes and Groove have diverged some, though their common heritage is evident. Groove, now in Version 3, is a multi-user collaborative environment; that is, it provides the plumbing and some process templates for group work in a separate, protected environment that allows users to collaborate in a single virtual space, synchronously or asynchronously, even if they are on different networks, behind different firewalls and on different kinds of machines. (Some people use Groove to “collaborate” with themselves as they move from device to device and want to keep their own work in sync.) Although the original focus was on document management and sharing of content, including maintenance of document or application data integrity as different users changed things at different times, the new version has an enhanced Web services interface that makes it easier to integrate with other systems.

Its strength is in dealing with all the plumbing to make the collaboration seamless – which is much harder than it looks. The difference from, say, Notes and IBM’s WCT is subtle, but it means that Groove assumes you want your tools and teams first, and your workbench overview is secondary. Also, IBM/Notes assume that you are part of an enterprise and that your coworkers are on the same system. Groove is more oriented to the needs of individual users, small businesses and teams working with people *outside* their own organizations. Of Groove’s revenues (undisclosed), about 40 percent comes from corporations, 40 percent from public-sector institutions (governments, especially secure and military units, and NGOs), and 20 percent from individuals and “IT-less” organizations of fewer than 50 employees.

Says Ozzie, “I believe strongly in a UI designed specifically around management of complexity, information overload and awareness; hence the IM-style launchbar with presence management and the ‘Workspace Explorer’ paradigms we use. Although Groove integrates deeply with Outlook because it’s the primary [mail tool] out there, Groove isn’t a mail program, It treats mail as something you need to deal with to work with other people, but it’s something you’d rather spend *less* time in, not *more*.” The user can also check the status of a workspace: when the user last checked it or when it was last changed and by whom. The system also includes a discussion tool.
Certainly, Microsoft is thinking about these things too, but it doesn't seem to feel quite as much pressure. At the back end, it's working on Longhorn, the new database-based OS/file manager that won't be out for several years.

On the attention/activity side, Microsoft is working on extending Office, mostly via Outlook, and also improving its Sharepoint collaboration suite. “What would help [for these process-management tasks]?” asks Microsoft CTO David Vaskevitch. “Think of it as a version of Outlook. You would still have an inbox or a task list, but you would also have a flowchart. Or actually, you would have a whole bunch of them. Instead of giving you a task out of context, it would give you the broader view, including what other people are doing. And it would help you with ‘correlation’ [cf. activity threads]: figuring out and managing which payment matches which invoice – it could be several payments – or which document corresponds to which e-mailed request for a proposal. If someone gets an order to approve, they may not just approve the order; they may decide to split it into several orders because something is back-ordered. How do you represent that?” Good question!

Vaskevitch sees the world in three tiers: “There’s a database at the bottom, so you can ask company-wide queries. At the top, there’s a document- and mail-oriented layer. The thing in the middle – which we don’t really have yet – is the repository for all the organization’s business processes, a business-process server.” This is orthogonal to the layering from personal to departmental to corporate-wide. The middle repository can support everything from individual schedules to departmental collaboration to company-wide workflows (analogous to everything from spreadsheets to databases).

“In terms of where this crosses our product line right now, you can see the start of it in Biztalk and Sharepoint services. Each new release will have fairly major steps in this direction [of supporting the middle tier of processes]. Sharepoint supports collaboration, whereas Biztalk is a more traditional application server that supports structured, IT-programmed workflow applications built around a database and data transactions.”

In Longhorn, says Vaskevitch, Microsoft will redo the underlying database and file structure: “One main thing will be shifting the basis of the OS from a file system to a database so that you can have standardized schemas and store schedules, plans and documents. But to support business processes, you really need a suite of products – the equivalent of applications for an application server.”

Outlook, of course, is notable for its Exchange server-based calendar, which has become a de facto standard. It includes a variety of multi-user scheduling features and automated invitations and confirmations. It integrates with Outlook’s contact manager in the sense that you can drag contacts registered on the same Exchange server into an appointment and automatically invite them. Many third-party products attempt to “improve” on features of Outlook mail, but almost all of them rely on Outlook/Exchange’s calendar as a default.

Vaskevitch concludes: “Just consider, when you read about Bill Gates giving a speech, you’d think we’d have some kind of great collaborative environment around that: developing the ideas, organizing demos and Power-points. But in fact, mostly we just use phone calls and e-mail.”

(Groove’s archivable version of IM). It is more of a platform than a development environment, but it does support forms and lightweight multi-user process applications. Reflecting its four years on the market, Groove includes 13 mostly form-based templates complete with customized tools to support common use cases including project management, opportunity tracking, customer reference, relationship management, document review, status reporting, asset request and issue tracking. “It was out-of-the-box templates that made Notes catch on with VARs,” notes Ozzie.

As in Notes, much of the work other than sharing documents and application files is accomplished through Groove forms; they benefit from Groove services such as synchronization, offline use and security. They can also synchronize with external data sources, eliminating duplicate copies of data and the need to re-enter information.
Open Source Applications Foundation: Back to the future

While a variety of vendors are trying to build Meta-mail tools for profit, the Open Source Applications Foundation is building its suite as a non-profit venture. It scratches both the user itch and the philanthropic itch of founder Mitch Kapor, who co-designed Agenda and who supported the creation and launch of Notes while he was founder/CEO at Lotus. And it raises the interesting question of whether open source is the “right” way to develop applications as opposed to infrastructure. But it won’t really answer the question. Any such project has too many specifics…

The idea behind OSAF’s forthcoming product Chandler, named after detective novelist Raymond Chandler, is to build precisely the kind of flexible attention-management, personal-workflow tool we describe in this issue, with a focus on the individual user. So far, the project is between Release 0.3 and 0.4. Code is freely available for download either for users, who are advised not to keep unbacked-up data in Chandler, or for developers, who can contribute to the main project or build modifications for themselves.

Because the product is still far from completion, it’s impossible to be specific about what it will actually contain at release 1.0, but it will invite people to add or fix things rather than complain about them. Unsurprisingly, the project is taking longer than expected, and Kapor has appointed himself development manager. You can read about that on his blog (see resources, page 39).

The product suite itself sounds promising. Rather than some abstract software developer’s notion of how people can be efficient, it takes its design philosophy from the work of Dave Allen, author of “Getting Things Done” and recently featured in an article by longtime Lotus Agenda fan James Fallows in the Atlantic Monthly. Allen has a solid practical understanding about how people can organize their personal workflow. The ideas are not a secret,” Kapor says. “They go beyond mere efficiency to the notion that if you record all your tasks so that you feel secure you haven’t forgotten anything, you can get into the flow and move from one thing to the next without feeling distracted. We want our tool to reflect and support that – with separate modes for seeing the overview and then following the sequence of tasks.”

Chandler supports the kind of triage Allen suggests – handle, defer or file away, ideally with some label or other. “Good search is even more valuable if you have things in approximately the right piles,” says Kapor. “Very often you get a piece of mail with information about an event or task. You need a quick and seamless way to say this is an event and stick it into the calendar. And you can add a headline for how it would
appear in the calendar." Chandler will support iCal, the standard behind both Outlook’s calendar and the Mac’s iLife. “We’ll be very standards-compliant!” says Kapor with the irony of a disappointed idealist. “But unfortunately the existing calendar programs aren’t that consistent. So we’ll do whatever a single party can do....”

As for timing, says Kapor, “We’re hell-bent on making dog food, stuff that we can use internally, this year. And 2005 is the year of dog food for everyone.”

Managing Processes

The tools we describe above are focused on the individual user experience and tasks, although some of them, especially Groove, support the creation of lightweight multi-user workflows. This section focuses on tools where the process rather than the individual user is central. It starts with the application wiki, an informal group tool, and then moves on to Resilient for industrial-strength, tightly “managed” workflows. Whereas most of the wiki applications are basically forms-based apps, Resilient manages formal, auditable processes. Finally, Phios has a taxonomy for keeping track of all these processes.

The basic benefits compared to most existing workflow applications are flexibility and transparency. You can modify a process just as you can modify a formula in a spreadsheet, and you can inspect the new formula and see the impact propagate through other processes. Of course, plain code is very flexible too, but it lacks the granularity that, say, Resilient gets from its object-oriented approach and it has to be reverse-engineered to be understood. Meta-mail comprises a high-level grammar/language for making changes so that the system can “see” what has been done to it.

Project WikiPad: Application wiki

Wikis manage shared spaces that are extremely flexible and forgiving; any member can edit them or create new pages, but anyone else can roll the edits back at will. (Group blogs have some of the same flavor, but without the group editability. see RELEASE 1.0, MAY 2003.) CEO Joe Kraus and Graham Spencer, who together co-founded Excite in 1993, co-founded Project WikiPad last year and hope to launch later this year. The WikiPad is designed to be a flexible, multi-user development/execution environment that lets users build, modify and use lightweight data-based applica-
tions almost as easily as they can construct documents in a wiki. They call their tool an application wiki, distinguishing it from a classical “document” wiki.

“People think using wikis as an application development platform is kind of weird, but wikis have many capabilities we need,” says Kraus. “They work as a better way to build lightweight workgroup applications. We like them because they collect a ton of energy; there are many cooks making each application better.”

WikiPad is not really focused on process management, but we expect many of the applications built using it will be process- rather than data-intensive. The idea is to use the flexibility and forgivingness of a wiki to design workgroup applications. Says Kraus: “Most people don’t know how they want things to be structured at ‘the end.’ Wikis allow you to see how people work together, what organizing structures emerge and then allow you to modify the wiki to match the emerging behavior. Apps shouldn’t need to be defined entirely up front. Instead they should allow people great flexibility; once habits and patterns emerge, then encoding or more structure can be applied.”

When Cotton Mather started Harvard centuries ago, says Kraus, “He said he would pave the campus where people have walked. In the same way, we want structures to emerge from people’s activities. Users build the forms and define the data they need, and then the structure emerges. The design process never ends.”

Of course, you have to hope that all the users use the same term for person: If Juan calls them “users” and Alice calls them “customers,” they may never pool their contact information. Says Spencer: “We expect that people will come up with different names for the same things, so in keeping with the wiki philosophy, we make it easy to make changes as you go. You can always add or remove properties for any object in the system, so both ‘users’ and ‘customers’ can have a name property; then you can search, sort, or filter by the property that they share. You can also rename properties if necessary.” This will probably work fine as long as no one expects the WikiPad systems to scale up or to pass a corporate audit test.

Underlying WikiPad is a flexible, extensible database that can handle whatever the users throw at it. A form, says Kraus, is like a mask or a view into the database. Each user can create a new page – i.e. a form that either creates a new field or uses ones already in the system. One page of any team’s application wiki is a list of all the
already-created forms and data elements. “You don’t need to pre-formulate the schema,” says Kraus.

Will it actually be simple enough when it ships later this year? “Our target is the guy who can write an Excel macro,” says Kraus. He cites one application the WikiPad team built – in four hours – to track job applicants for a venture capitalist. Previously, the VC e-mailed around an Excel spreadsheet.

Among the nice wiki-like features of WikiPad, adds Kraus, is that “You can e-mail information to the application and it does the right thing, as long as you use a reasonable approximation of its format. If you send a data-entry form to an application, it can scrape the relevant data into its database.”

Resilient: Making processes into objects

Many companies have foundered trying to automate corporate processes, warn our VC friends. So we’re not yet ready to declare Resilient a success, despite enthusiastic response to pilot deployments focused on governance, management and compliance at organizations ranging from a mid-sized startup to a global Fortune 100 manufacturing firm and a large US government agency. It plans to ship at the end of this year.

Resilient is the third company founded by Jonathan Hare; each of them aspired to enable non-technical business users to define and execute continuously evolving business processes on the fly. This vision, while compelling to customers, investors and analysts, proved easier said than done. His first try, Evolve Software, was launched in 1995 to enable a radical new generation of “adaptive applications,” but evolved into a more traditional enterprise application focused on professional-services automation. It went public in 2000 and later was acquired by Primavera. The second, Consilient, closed down shortly after 9/11. The third, Resilient, reflects what Hare says he learned at the first two: “Pursuing the Holy Grail of enterprise software is much easier if you use Microsoft infrastructure and tools instead of building your own.” More than that, Resilient’s basic user interface is the user’s own e-mail, allowing each user to stay in mostly in his or her specific, traditional environment and still benefit from – or be guided by – the system’s processes.

Technically, the system treats processes like objects: Its software-embodied processes can be combined, copied and specialized according to the system’s own rules to
build arbitrarily complex processes. Moreover, they can be modified on the fly. The software starts with a small set of primitives – entity, thread, message, content, script, link, and tag – which can then be combined and concatenated to build processes a user would recognize, such as a budget approval, a document review cycle, a new-employee hire and so forth. As with Notes, much of the company’s success will hinge on a usable set of templates and enthusiastic developers, both in-house at customers and at ISVs.

Of course, users have to agree to play the game and reply via the system, but Resilient makes it easier to do so than not. (It’s assumed that somewhere there’s an edict saying: “It’s not an approval unless you do it through the system.”)

How does it actually work? Let’s say Juan starts a process, by clicking on “budget approval.” If he isn’t simply reusing the budget approval from last quarter, he will designate the file containing the budget, list the users who need to approve it (specifying who must and who may see it) and set some deadlines if he wants. (This new process can inherit most of the features of the old one or reuse other processes whole.)

When Alice gets the message asking for her approval, it arrives as a simple HTML e-mail (no Java scripts). Although this thread message has a familiar appearance with a subject line, message text and attachments, it also includes smart HTML buttons for structured responses such as approve or decline, delegate, acknowledge, defer and abstain. Alice clicks on the one she wants, and ends up in a mail-like browser window where she completes the transaction and is invited to send an accompanying message. She may also have other options for follow-up, depending on the process involved. (The system generally assumes people are online, although it has facilities for offline users.)

The system can be hosted either on a corporate server or by an ASP operated by Resilient or some third party. Behind the scenes, each button actually links to a unique URL, one for each message and recipient response. The process threads automatically collate and summarize such responses to determine, for example, that a specific request was approved based upon the responses of five recipients, notify them that approval was granted, and trigger other messages or actions as a result. The initiator or anyone else with access can check (via a thread browser) the status of the process.

Each thread is made up of pages (or strands) containing all of the messages, attachments, related threads and links to individuals that are part of or linked to the
thread. A thread can be based upon a template that includes scripts or triggers to automate workflow, customize appearance, enforce policies or invoke external applications or Web services.

A power user can learn to create new processes with minimal training. . .though, like spreadsheet models, process threads can get arbitrarily complex. We did this ourselves, as shown in the screen shot, above. We found it surprisingly easy, if still a little slow and cumbersome – which should be fixable. It gives one the feeling: “Well of course it’s easy! This is how it should be done” – but of course it usually isn’t.

Here we are in the middle of creating a process thread to review this section of **Release 1.0** with Resilient, called “Release 1.0 editorial review.” Christina has selected Mathew Spolin to participate. She’s got one thing for him to review, and one question (“request”) for him to answer. She can set deadlines if she wants, and below there are buttons (not visible) to specify who must see and who may see the information she is sending. He may want to delegate part of the process, but he can’t forward the information to a third party without permission.
Users can interact with threads entirely through standard, unmodified e-mail clients or Web browsers. Regular users can install add-ins so that threads appear as a native feature of their e-mail client (e.g., Outlook), supporting a more familiar user interface and getting the performance advantages of a local application. In future releases, Resilient plans native support for RSS feeds and add-ins for standard desktop applications such as Microsoft Office, so threads can automatically keep track of changes to external data.

All this may seem somewhat heavyweight, but it enables Resilient to keep track of what’s going on in the background, while keeping a familiar environment for users in the foreground.

If the system is so easy to modify, how does anyone control the proliferation of slightly modified processes and figure out what is really going on? As with any large-scale organization, at some point someone may feel the need for a global process dictionary, akin to a data dictionary. (See page 29, on this page.) You either need some way to minimize variation and force everyone into the same patterns, or you need a way of managing the complexity. What happens when you have built hundreds of processes and now you’re trying to figure out what happened?

Fortunately, Resilient’s structure, where everything is built as a modification of something else, means that the system should be self-aware enough to usefully answer such questions. It will still need human intelligence to figure out what to ask, but you will be able to ask the system questions such as: Find every instance in which the approval process was changed by an exception to a rule, and by whom.

Resilient is trying to solve what CTO Mathew Spolin calls “the impedance mismatch between how developers write programs, and how people actually work and use the programs.” By contrast, there was a positive match between Spolin, who had started a BBS when he was ten (24 years ago), and Hare, who was trying to combine the ideas of self-organization and complex adaptive systems with the demands of corporate processes. Spolin got started in the corporate world as the first software developer at Human Genome Sciences when it still had only 25 employees. They were mostly scientists, working together and trying to match sequences of As, Cs, Gs and Ts with corresponding data about characteristics of individuals, family trees and other kinds of data. “There was no definitive process for making discoveries,” recalls Spolin, “and there was no predictability to the discoveries. Yet if you got granular
enough there were patterns that were repeated” both in the gene sequences and in the processes used to discover them.

Later on, as Human Genome grew, Spolin designed a workflow system for the patent application process almost as unpredictable as the scientific process, he says. That experience underlies his approach to Resilient’s software, which is the equivalent of a genome for business processes, he says.

**Phios and the Process Handbook**

If Resilient has a system that can build business processes from genes, Phios, a commercial company that grew out of the Center for Coordination Science at MIT, has a comprehensive “business genome map,” composed of descriptions of elemental business processes that combine and iterate in complex ways, instantiating the structure and activities of a business.

Phios (PHios) was co-founded by Tom Malone, a long-time researcher in CSCW and MIT professor and most recently author of the book “The Future of Work.” He spoke at PC Forum in 1992 and was also covered in Release 1.0 frequently during the late 1980s and early 1990s. Over the past decade, the CCS has produced what it calls the “Process Handbook,” an extensive taxonomy of business activities, building up from a small set of primitives: create, preserve, modify, destroy, combine, separate, decide, manage and “unclassified.” Any other (business) process is either a specialization or combination of these; the Handbook lists 5000 of them, in a carefully laid-out framework designed to help users or software developers define and organize their own business processes.

The software tools for extending the taxonomy allow users to specify “families” of related processes in which specialized versions of a process automatically “inherit” the characteristics of the more general processes. This is similar to inheritance in traditional object-oriented programming, but as Malone points out: “Traditional object-oriented programming inherits down a hierarchy of nouns; the Process Handbook inherits down a hierarchy of verbs.” This verb-oriented hierarchy, therefore, could provide the kind of meta-model that many of the other systems described in this issue will need to organize their repositories of activities and business processes. However, Phios’s tools are for developers to use in managing their code; they don’t build working implementations of the processes they describe. That’s up to companies such as Resilient (which has no formal relationship with Phios at this point, though they have a common vision from different perspectives).
Under license from MIT, Phios offers the Process Handbook and related software, content, and consulting services to its customers. For example, says CEO Patrick Flynn, a large health-care company has reduced cycle times and costs with the help of the Process Handbook. He says, “It created a ‘gold standard’ process for managing hospitals, and then created a different specialization of this process for each partner hospital. This helped them improve quality through ensuring consistency, providing uniform access to information and materials for training and education, and reducing handoffs through electronic links and XML exports. The net impact was to reduce the time (and cost) to evaluate a hospital, make recommendations, share data, and implement solutions.”

Phios reflects Malone’s conclusion that his ideas are more likely to be useful and visible with a commercial vehicle behind them, ideally being used by many of the companies described here...and fostering at least some interoperability among them.

In that spirit, the company is planning to distribute its wares under a new, intended-to-be-copied hybrid license being designed in conjunction with Creative Commons (see resources, page 39). The idea is to offer the software, patents, and database to two kinds of users: commercial operations who would pay Phios royalties, and to all others who would get use of the material for free. The idea of two kinds of licenses is not new, but the precise terms are intriguing. The proprietary part operates similar to a patent: It gives the creators a short period in which to reap rewards, and then it requires them to open-source their enhancements after a specified period, probably a year or two. Thus it rewards the inventors but also encourages the spread of their work. And yes, the license itself will be freely copiable.

**Presence/Role Management**

Calendars and scheduling pose challenges that are well-understood if not yet completely addressed, as vendors and users converge around iCal. Even more interesting is the problem of real-time presence management. As with other kinds of information, the challenge will change from finding things out to deciding what you want to know: Which people’s presence are you interested in knowing about?
Or to tell: How “present” do you want to appear to others – specific others? How much of your own presence – now and in the future – do you want to reveal? In theory, strangers could know that you are online now – or that you will be in Palo Alto next week. If you don’t want to be present, can you automatically refer people to a stand-in? Can you specify a different stand-in depending on the context – i.e. send your spouse to Parents’ Day and your VP to the conference reception?

Here, Meta-mail has a lot to learn (or borrow) from current identity management tools on one side, and from instant messaging on the other. Many software vendors are building or using identity management tools, but few can handle the complexities of different roles as well as Convoq. Though for now it is focused on setting up online videoconferences or shared application/presentation sessions, its identity-management capabilities are much more broadly applicable.

**Convoq: Are you part of the critical mass?**

Finding the appropriate people in real time or for planned time is the province of ASAP from Convoq (see also **Release 1.0, March 2004**). The product launched in March (at PC Forum) and has picked up about 130 corporate accounts from small and medium businesses through telesales. The company has just started an enterprise direct-sales effort and is beginning to talk with potential partners for reselling and embedded OEM solutions.

As we outlined in March, the basic proposition is presence-based ad-hoc meeting management. ASAP uses a combination of instant-messaging and e-mail infrastructure to detect users’ presence and arrange meetings on demand “as soon as possible.” Says founder Chuck Digate (previously senior VP at Lotus, founder of BeyondMail, and chairman, president, and CEO of MathSoft): “It gives you the ability to publish your presence anywhere – on your blog, in e-mail. It could be a link back to you, or to a stand-in or a tech-support person.”

Currently ASAP operates as a stand-alone application delivered as a hosted service and focused on Web conferencing, but one could imagine its functionality used in Meta-mail to establish all kinds of synchronous (and asynchronous) communications for individuals and their roles and stand-ins. The basic technology is a system that sends invitations (by IM, e-mail or eventually other means) to an arbitrary number of users to set up an online meeting. If they’re available (according to their presence indicators), ASAP starts the meeting so they can join. If not, it monitors their availability, with permission, until the meeting can be convened. Registered
(and paid) users of the system have a client-side tool from Convoq that’s like a unified IM presence detector. It presents a contact list that aggregates the names and presence metadata from all the popular IM systems that a user may be running. Invitees simply click on a link in their IM chat window or their e-mail inbox to join the Flash-based meeting without having to download or install any software.

You can also put an ASAP link on a website, so that any passing customer can “con-voke” a meeting with a customer-support rep (or a salesperson masquerading as customer support). And in a feature currently in beta, a personal ASAP link embeddable in mail, for example, could include a presence indicator for the recipient to see. If the sender is available at the time the recipient responds to it, ASAP sets up the meeting – online, but it could be by phone or IM.

That part, though complicated to implement, is relatively simple to comprehend. What makes the tool interesting is its integration with presence and identity information. For example, ASAP can arrange multi-person meetings the moment a user-defined critical mass is present. The person calling the meeting can list, say, four individuals whose presence is necessary, two roles or “Lifelines” whose presence is necessary (e.g. the first available person from each of marketing and sales), and others who are invited but on whom the meeting does not depend.

In addition, ASAP has the concept of “stand-ins” – different people standing in for the same individual in different roles. For example, the CFO could stand in for the president in a meeting with investors, while the VP of sales would stand in for her with a potential client.

As a result, one could imagine using a combination of Lifelines and stand-ins to resolve an arbitrarily complex situation, such as “there must be an equal number of sales and marketing people, and either Juan or Alice but not both.” You would invite an equal number of Sales and Marketing Lifelines and Juan, whose stand-in is Alice.

For our purposes, Convoq’s online meetings are less interesting than its ability to mediate presence across multiple people for any kind of connection, whether by video-conference, SIP phone call or even face-to-face in a corporate lunchroom but convened via a set of Treos. Although Convoq’s special sauce is centered around presence, ASAP also supports scheduled meetings and integrates with the Outlook calendar. The company recognizes the importance of supporting multiple methods.
of convocation. Says CTO Chris Herot (also formerly with Lotus): “There is a con-
tinuum between ad hoc and scheduled, and between real-time and asynchronous. 
Hence, you can schedule a meeting, insert it in Outlook, but still start and end it 
when you wish, and invite additional people in real time. You can send someone an 
instant message but if the recipient is not available, have that instant message turn 
into a stored message. And when the recipient clicks on the stored message, if the 
sender is available, the conversation turns back into a real-time conversation.”

Its ID and presence management functionality fits nicely into Meta-mail, where doc-
uments and files or application data as well as e-mails could contain links to relevant 
individuals or roles whose presence information might be useful. For example, 
imagine a social-networking tool that could select a path to your desired target 
based on the presence of the individuals on that path, or you could quickly query 
someone on the path before making the approach. (Obviously, there are privacy 
issues here, as with all contact and presence management applications.)

Data Detection: This is a Title

There are many challenges beyond the architectural/communications issues of collabo-
rationate work, whether within the enterprise or across boundaries.

Among the notable features of IBM’s Remail (PAGE 15) is automatic detection of 
structured data such as dates and contact information. It won’t be available to out-
siders until Workplace Client Technology 3.0, sometime next year. But you can 
already get a taste of it from a tiny Cambridge, MA-based company called Textual. 
Los Altos, CA-based Mailblocks is working on a similar feature and demoed it pub-
licly earlier this year at the IDG Demo conference.

The idea is simple. A small background program looks at text and recognizes (or 
suspects) that some piece of text is in fact a date, time or contact information. Then 
it adds it to a calendar or contact database.

For now, however, the user typically has to select the relevant text (although auto-
matic recognition already exists in a variety of domain-specific applications such as 
resume scanners). And the user often has to help in figuring out the dates, especially 
those calculated from the source date, such as “tomorrow” or next week,” and 
matching them appropriately to events. For example, “May 10.” Is that a date, or part
of a sentence: “May 10 of my best friends [from Orkut] come to your birthday party next week?” In fact, the mail was sent on June 12 and the party was actually on June 20. A truly well-tuned recognizer could have detected that the party date was probably not May 10. . . . But none of the vendors can yet promise that, even though IBM’s Remail does look at verb tenses near the dates it recognizes.

**Textual/Anagram: Automating an annoying chore**

Although the functionality of Textual's Anagram tool is more limited than the IBM or Mailblocks demos, it is working and available now for download onto a Windows PC (with eventual payment of $19.95 per PC or $26.90 for two devices/one user). The company is also looking to license it on an OEM basis to other software vendors; it says it’s “exploring” versions for the Palm and RIM/Blackberry platforms, whether with those vendors or independently.

The tool is very simple: The user highlights a piece of text on a user’s screen, in any source document at all (mail, a Web page, a file of any kind) that represents a potential appointment, a contact or some other semi-structured text. When she presses control-C twice (or some other user-definable trigger), Anagram opens her Outlook or Palm calendar or contact database and creates a tentative new entry. (Or in theory the user could select some other destination – for example, the resources section of this newsletter.)

The user can confirm the proposed entry, modify it or reject it outright. There is an option to mail any incorrectly recognized text to Textual, which has given the company three years of valuable feedback, says founder-developer-CEO Nicholas Maddix, a former electrical engineer who comprises the company’s entire full-time workforce. Among other places, he formerly worked at Ideo’s Lexington, MA, office, where, he says, “I got a great education in UI design.” Maddix did most of the programming himself, with the help of a computational linguist and a few foreign-language experts. Some of the integration with Outlook 2003 was done by Arcadia Software of St. Petersburg, Russia. There’s also a part-time business development manager, but so far no other staff. Says Maddix: “All our users have come from word-of-mouth; Anagram promotes itself. We’re tiny, grass-roots really.” But he has noticed a pick-up in interest lately, and plans to start looking for funding soon.
Precisely because Anagram is dealing with the vagaries of human expression user testing is key. You can figure out a lot about words in a development environment, but in the end you need to test with real messages. Over three years, the accuracy has improved from “just functional to remarkably accurate,” says Maddix. Particularly difficult, it turns out, is distinguishing between a person’s name and a company name. “That’s where my linguist friend spent most of his time,” says Maddix. While getting dates automatically can be a challenge, the fact that the user selects the text means that the user can supply it manually, if needed.

Textual is currently working on enhancements to the current functionality, including a version that will enter data into the new version of Palm’s desktop software organizer as well as Salesforce.com and ACT! organizer.

In addition, says Maddix, he has built a capability to recognize relevant data in context, rather than merely to parse what a user highlights. “It’s not a problem for an individual user to select a piece of text,” he says, “but if you’re a recruiter going through hundreds of resumes for example, then you want it automated. Another company wants to build it into a toolbar.” And, we note, even some individual users are lazy enough to want such functionality.

**Mailblocks: Not just a spam-blocker**

Mailblocks is working on similar functionality, code-named Maxwell (after Maxwell Smart, the fictional secret agent) for the next release of Mailblocks, hitherto touted mostly for its challenge-and-response/white-list spam-blocking. (In fact, it’s a useful all-around tool, especially for aggregating multiple e-mail accounts in one inbox.)

Mailblocks was founded in July 2002 by WebTV cofounder Phil Goldman, who died late in 2003. “Phil was passionate about e-mail,” says CTO and acting CEO Rich Landsman. “He was excited about offering a clean inbox, spam-free. He wanted to make your inbox the place you want to be during the day. But once you get rid of the garbage, you still have information glut from all the good stuff – not just e-mail, but any information. For example, if you receive voice mail or faxes in your inbox, you should have specific features that help you manage it; e-mail is just one example of the online information we all have to deal with.”

Landsman previously worked on Lotus Agenda 1.0 in 1987 to 1988 and then joined Irene Greif’s team at Lotus for a while. He acknowledges, “We have to take care not to
try to boil the ocean. So we started [in Maxwell] with a few easy things that seem really obvious to us.”

All depend on recognition technology harking back to Landsman’s days working on Agenda. The tool can recognize (mostly!) signature blocks, addresses, phone numbers, dates and times. Once it finds what it thinks is contact information, says Landsman, “We look at your current contact list to see if it matches, and then we look to see if we have found anything new. Thus it can either create or update entries in Outlook, the Mailblocks Web calendar and anything synched with them such as a smart phone or pocket PC. It’s not a product yet, but in its full form it could work through an existing e-mail corpus to build a contact database from a user’s old e-mails.”

It performs a similar function for appointments and meetings. “Even people using Outlook and Exchange seldom do formal meeting management,” says Landsman. “People who are calendar-managed – they look at their calendar in the morning to find out what they’re supposed to do – they often miss e-mails [about meetings] that should have ended up in their calendars.” Once Maxwell recognizes a date – e.g. “next Tuesday,” “4/30” (but not 1/100), “tomorrow” and the like; and 2:30, 4 o’clock, 14.30, 8 pm – it highlights it and suggests opening that day and time on the calendar.

Maxwell can also help generate distribution lists. It recognizes e-mail addresses wherever they are in an e-mail – in a quoted section, for example – and collects them all into a proposed distribution list. The user can select them all, or only the particular subset she wants. And finally, it allows you to type something into what it calls a Max note – an address, an appointment, and so on – and then suggests where to put it.

All this should be available by the end of this year, says Landsman. That is, by December 2004.

**Meta-mail Meets Social Networking**

What will Meta-mail mean for all of us? Perhaps the best way to explain it is to consider how it might affect a category of software that hasn’t yet frozen – social networking tools and platforms. (See RELEASE 1.0, NOVEMBER AND DECEMBER 2003.)
Right now, the social network platforms are generally monolithic server workflow applications: They manage interactions among users, such as declarations of a link, introductions of one person to another by a third, searches leading to double-blind introductions by the system, and the like. They also allow users to share information in a common but selectively visible space—everything from resumes and job-postings to photos of themselves or their cats, their personality profiles (true, fake or “aspirational”), and links to their “friends,” according to various privacy strictures. The ratio of transactions (or workflow) to shared content varies, as does the ambience, but the basic formats are the same.

We’re not sure exactly how these environments are constructed, but they feel pretty inflexible to the average user. Moreover, they are group-centric rather than user-centric. Users can’t define or modify their own social-network processes, so they are stuck with “Will you be my friend?” Hundreds of people (according to our own unscientific surveys at industry gatherings) feel uncomfortable disappointing or insulting the people who invite them over these networks.

When someone asks, “Are you a member of LinkedIn?” they’re wrongly assuming that LinkedIn is a social network, when in fact it’s a hosting service for its users’ social networks, some of which overlap. The proper question would be: “Are you a member of Juan’s social network on LinkedIn?” (It’s a little different for Friendster or Tribe, where part of the point is to create/support a subculture—and an audience for ads.)

So what should LinkedIn be offering? (We don’t mean to pick on LinkedIn in particular; it’s just one that’s good enough to be worth criticizing and improving. And in fact, LinkedIn is about to add some client-side functionality of the sort we describe below, but unfortunately only for Outlook users, through plug-ins.) It should be offering a user toolset—a suite of process templates—whereby I can manage my own social network locally, from myself outwards. It should acknowledge that each person is a member of a variety of networks, of which they are the center. (Networks overlap through individuals, not as groups.) And then, yes, it should manage a secure, selective-access database of links among its users.

Let’s close with another true story (fake names). I recently had lunch with Bill and decided to introduce him to another friend, Will. Like many people, Will has changed jobs recently, so I got his latest e-mail address from the cc list of a message about a board meeting we both attended recently (forget the address book!). Now that’s a social network—a small one, but a real one. I wouldn’t declare it to the world, and I haven’t asked all the people on that board to be my friends. The fact is, most of my
social networks happen in the context of communications about something or other; they are activity-based, and they happen in my regular mail, not via some social network platform. With many of my contacts, I share several activities, seamlessly.

But it would be handy to have some SN processes in my regular environment, for example, to select two people – one from my calendar and another from my address book – and generate a customizable e-mail inviting them to meet. Of course, that mail could be an IM, or it might contain some tool that would allow the recipients to declare their availability over some temporary IM-like presence manager.

I do this all the time, without a template. But with such a template, I could automatically create a record of the introduction I have made, so that someday I could search on either of those two names and a process such as “intro” and find the e-mail with which I did the deed. That record would belong to me, not to LinkedIn. I could easily find all the introductions I had made for a particular company, the reference checks with contacts I had made on a potential new hire, and so forth. I could also use and modify templates to define my own kinds of connections (business contact, acquaintance, secret admirer). I could develop my own templates for polite rejections (“Unfortunately, I don’t have time to give this the attention it deserves”). Meta-mail might ask if I want to designate someone as a good contact after I have sent them five e-mails in three days, or whatever metrics I set.

**Render unto LinkedIn. . .**

Where does that leave LinkedIn and its ilk? First of all, they would have to win my business by making its templates easier to use than doing things for myself, rather than by holding my contacts hostage.

But they would have a role. When I ask Juan whether he wants to be in my list of searchable contacts, his reply would get registered in my own contact database; if he said yes, a notification would go to LinkedIn, which is where the heavy-duty searching and matching and the link-crawling would occur. LinkedIn would find its proper role – connecting me and my contacts to the social networks of other people, while giving me the tools to handle my own for myself.

But I would maintain my own social networks locally; I would use LinkedIn only when I was trying to reach someone not directly linked to me (on behalf of myself or...
someone I wanted to help). LinkedIn would compete on the basis of the quality of its templates and its modification tools. In the long run, the various social networks might interlink directly – but only once proper privacy assurances and social protocol standards are in place. There would be complexities about indirect links – one person’s strong link is another’s “met once” – but that reflects real life: Many more people “know” Bill Gates than Bill Gates knows back. Perhaps the law of networks – the strength of a tie degrades by the square of the number of links – would become more apparent. . .and perhaps that would be a good thing.

We are not sure how good that is as a business model, but it works as a social model.
Release 1.0 Subscription Form

Complete this form and join the other industry executives who regularly rely on Release 1.0 to stay ahead of the headlines. Or if you wish, you can also subscribe online at www.release1-0.com.

Your annual Release 1.0 subscription costs $795 per year ($850 outside the US, Canada and Mexico), and includes both the print and electronic versions of 11 monthly issues; 25% off the cover price when you order from our online archives; a Release 1.0 binder; the bound transcript of this year’s PC Forum (a $300 value) and an invitation to next year’s PC Forum.

NAME _____________________________________________________________
TITLE _______________________________ COMPANY _________________________
ADDRESS ___________________________________________________________
CITY ___________________________ STATE _______ ZIP ________ COUNTRY _______
TELEPHONE ___________________________ FAX ___________________________
E-MAIL* ________________________________________ URL ___________________
*personal e-mail address required for electronic access.

☐ My colleagues should read Release 1.0, too!
Send me information about multiple copy subscriptions and electronic site licenses.

☐ Check enclosed ☐ Charge my (circle one): AMERICAN EXPRESS MASTER CARD VISA

CARD NUMBER ________________________________ EXPIRATION DATE ________________
NAME AND BILLING ADDRESS ____________________________________________________
SIGNATURE ________________________________________________________________

Please fax this form to Brodie Crawford at 1 (212) 924-0240.

Payment must be included with this form. Your satisfaction is guaranteed or your money back.

If you wish to pay by check, please mail this form with payment to: EDventure Holdings, 104 Fifth Avenue, 20th Floor, New York, NY 10011, USA. If you have any questions, please call us at 1 (212) 924-8800; e-mail us@edventure.com; www.release1-0.com.

Get our RSS feeds: We now have RSS feeds for Release 1.0 and Esther’s weblog. Subscribe to them at http://www.edventure.com/rss