BUDDY LISTS
by Jerry Michalski

Have you ever wanted to call a meeting of your project team NOW -- the instant you see a selling opportunity, for example -- but put it off because you couldn't get hold of them all quickly enough? That's one reason why brokerage firms spend a lot of money on phone "turrets" and "hoot 'n holler" public-address systems.

Most communications aren't that urgent or dramatic. Sometimes you want to chat with someone, but you don't want to interrupt. Plus, the time and trouble it will take to send e-mail, negotiate a convenient time and so on kills the urge to chat and makes the chat more formal than you ever intended. By the time you actually talk to them, you're tired and cranky.

In January's issue about conversation technologies on the Net, we described buddy lists, the diminutive on-screen windows that indicate when your colleagues and friends are online with you and allow you to communicate with them readily. These small, innocuous applications with the wimpy name can solve the kinds of problems we just described.

For example, you could call your team into a chat room or Web-based, multi-party audioconference by right-mouse-clicking on the group's label in your buddy list and selecting the appropriate function in the menu it provides. You could decide to send a colleague a short instant message based on the fact that her buddy-list icon shows she is not too busy to talk. She could then decide to postpone or escalate the chat to audio -- to "take the call," in more common parlance.

Under the radar

Amid the frenetic activity in browser standards (and bugs), push architectures and large-scale Web-development systems, a half-dozen companies have quietly developed buddy-list systems. Several of them have already attracted large, enthusiastic audiences.

Long run, buddy lists may be a more noticeable part of our everyday interactions with technology than browsers, which will meld into the OS; push, which will disappear into the infrastructure; and development systems, which already live in the back office.

INSIDE
BUDDY LISTS
Buddying 101.
Architectures and interoperability.
The dark side(s).
Some futures.
WHO'S WHO
AOL's Instant Messenger.
Mirabilis' ICQ.
Ubique's VP and Excite's PAL.
ichat's Pager.
OnLive!s LiveList.
Activerse's Ding!
PeopleLink.
Flash Communications.
Others.
RESOURCES
Names, phones, dates, URLs.

OUR HIGH-TECH FORUM: AMSTERDAM, OCT. 5-8
Push systems don’t really push, anyway: The client elements poll the servers periodically for updates, which makes it impossible to distribute breaking news right when it happens. To deploy buddy lists, companies have developed notification architectures that solve this problem and that have been missing in the Internet protocol suite. These architectures can do more than serve buddy lists, an idea we return to later in this section. For now, think of buddy lists as an improvement on the dial tone, touch-tone and busy-signal interface in use all the time today.

Ready? Set...

This issue of Release 1.0 examines the current state of buddy-list technology and explores its future role in the computing and communications infrastructure. The first section describes how buddy lists work, the thorny issues they raise and how various companies are approaching the market, which is crowding quickly with entrants. This issue’s second section covers those companies -- Activerse, AOL, Excite, Flash Communications, iChat, Mirabilis, OnLive!, PeopleLink, and Ubique -- though we cover them in order of market entry, not alphabetically.

Most of these companies’ products are still in beta; a few haven’t made it that far yet. All of them are free or bundled as part of a service, as in AOL’s case. A few companies plan eventually to charge for the client software. Given the $20 to $50 per person that they intend to charge, none of them is likely to collect exorbitant revenues, even factoring in some ad sales. So what’s the appeal? The revenue model here is subordinate to the feature’s high intrinsic value, which is the focus of this issue. The best buddy-list systems are likely to find their way into Microsoft, Netscape, AT&T and other companies who need the feature for their infrastructure.

Buddy lists are social. They offer explicit, general-purpose, wide-area support for the loosely linked relationship webs described by the Institute for Research on Learning’s Susan Stucky and Etienne Wenger (see last month’s issue); Inflow’s Valdis Krebs (see Release 1.0, 2-96); and Net Dev’s Duncan Work (see 11-96).

The functionality that buddy lists offer isn’t novel. It has existed in many guises within closed systems. Part of the fun of MUDding is seeing others in the virtual space and letting them see you; the same goes for other multi-player gaming environments (see Release 1.0, 6-93). The Internet commands "finger" and "who" return user profile information and login status; proprietary systems have had such capabilities, too. But polling for user status is much clumsier than seeing it maintained in real time. Also, the user base for those proprietary systems was usually small. The new buddy lists function across the open Internet.

What’s in a name?

The memorable term "buddy list" has caught on quickly, but it may not be the most appropriate one. From a marketing perspective, the word "buddy" exudes the kind of friendly playfulness that kept the perky original Macintosh from being considered seriously by many IT managers.

Other terms in use include "people browser," "pager," "personal access list," "presence systems," "awareness," "peripheral vision" and "notifica-
tion server." One could also try more mundane terms such as "contact list" or "people finder." Of this crop, we still prefer "buddy list," though the concepts of presence and awareness are key. We hope savvy IT managers scoot past any terminology biases they may have and jump right in.

Taking a functional perspective, perhaps "buddy list" doesn't evoke enough of the application's features. That is hard to say right now, because the feature set and its overall role in the interface are likely to evolve over the next few years. Moreover, buddy lists will probably touch so many other applications that their role may be hard to describe. It seems that the "buddy" part will always remain central. What to call buddy lists may become clearer as we explain what they do and where they may be headed. In the end, companies will probably have to find other terms for this feature. AOL has trademarked the term and plans to defend it.

On active desktops and social Interfaces

Tech weenies have created plenty of new language in the past few years trying to position their products favorably or to help frame what's going on (see Release 1.0, 4-97, which is on the Web at http://www.edventure.com/pods).

Our look into buddy lists shows how far off the mark Microsoft has been with its recent messages about "active desktops" and "social interfaces" (we'll mercifully ignore "information at your fingertips" and "where do you want to go today?"). In its most recent incarnation, Microsoft's active desktop (Internet Explorer 4) was in fact passive. Sitting in front of "channel" selectors and news tickers may involve on-screen motion, but it doesn't make the user do anything. Stuff moves around on TV screens, but we think of TV watchers as passive couch spuds.

Netscape's parallel efforts with Constellation, Mercury, Gemini and Compass are only marginally better. They include icons for tasks and contacts, but they don't yet make it easier to converse.

Then there's Microsoft Bob and the Office Assistant. Interacting with an on-screen character of limited intelligence may help you troubleshoot Office functions, but it's not social. Not by a long shot. Let someone know who else is around in a virtual space, and you have an active, even addictive social interface.

The PC is both an information processor and a conduit for contacts with others. Until recently, all attention was focused on the former function. Now the latter is coming into its own. Because NCs (Network Computers) can handle less local processing than full-fledged PCs, they may be especially well suited to collaborative communications. Most of the buddy-list systems described here will have Java versions; Activerse's is in Java only. Sounds like a good match.

The basics

The most basic function that buddy lists serve is letting people who are online see when other people they care about are online, too. People build
their buddy lists by adding other people's names to them (in the form of user IDs, user names or e-mail addresses). Typically, the people you want to "buddy" with have to give you permission to do so.

The invitation and authorization process is important. The best of them work roughly like this: After selecting "add buddy" from a menu or button, you type the e-mail address of someone you want to buddy with. The system checks to see if the person has already established an account. If yes, it sends that person a "request to buddy" on your behalf. If he approves the request, his name shows up on your buddy list.

If the other party is not already in the system, it sends the person an e-mail invitation that contains a brief explanation, a URL from which to download the software and a personal note from you, if you add one.

Float like a butterfly

In use, buddy lists put a small window on screen that can float above other applications. Minimized, buddy lists leave a small status icon on your desktop (in the System Tray for Win95; on the Menu Bar for Macintosh). Buddy lists are usually configured to load when you start your computer, so they are active all the time, assuming you have a full-time Net connection.

Each buddy-list system has its own way of showing when others are online and what their status is. Some show only the people online, hiding the names of people not online (AOL's AIM). Others show the full list but turn names of people online bold (Activerse's Ding!), add an asterisk or move them into different subsections of the buddy list window (Mirabilis' ICQ). Systems that offer only basic status settings -- online, offline, do not disturb -- usually show them by making names bold or changing their color (Ding!). AIM doesn't even have "do not disturb." Systems with more status choices use either icons in front of the names (ICQ) or user-supplied text descriptions after (Ding!, OnLive!'s LiveList).

Knowing someone's online is only the start. You can then use the buddy list to do many different things with others, ranging from simple instant messages to passing files or URLs and even running separate applications such as Microsoft's NetMeeting. You can invite several people to a chat. The buddy list resolves names and addresses.

A glance backward

Buddy lists aren't perfect, but well-implemented ones can collapse many functions of a phone call, potentially making communications much simpler and more fluid. They also allow users to do many things that an ordinary phone call can't do, as noted in the examples just given.

To make a phone call, you pick up the receiver, dial a number that ranges between two digits (an internal extension) to 35 or more (an international call, using your calling card with an alternate long-distance carrier). You have to think before you dial, because what you dial depends on where you are. Then you wait for the phone system to set up the circuit and ring the phone -- or give you a busy signal or an interminable voice menu system. Sometimes you make phone calls just to see if the other party is available for conversation, but not actually to have the conversation itself. That's why the buddy list status settings are important.
All this assumes that the person is on your buddy list already. The effort required to get people on your buddy list is about the same as to locate and store phone numbers today, but there are many ways it will become easier.

**Echoes of "hello"**

Much of what is happening today with buddy lists and the Internet mirrors developments in the early telephone system. Before telephones became popular, Alexander Graham Bell and others spent considerable time choosing a word that people would use to answer the phone. One candidate word was "ahoy"; we ended up with "hello."

That's the stage buddy lists are in right now. The companies covered here are each setting vocabulary, defining feature sets and exploring the social protocols that make a social technology work.

Once phones (more importantly, phoning) became popular, the manual system of using operators to connect calls proved unworkable. It was slow and it didn't scale. Eventually, it was replaced with direct dialing, which turned every caller into an operator, and electromechanical (later fully electronic) switches.

Similarly, early Internet phones and groupware applications suffered because users had a rough time finding each other and often had to type in IP addresses to make the applications work. Because buddy lists store that information, they should be an excellent place for many different applications to rendezvous.

**BUDDY-LIST ARCHITECTURES**

Buddy lists' real-time, social nature and potential scale presents hefty development challenges in both the user interface and server architecture.

The usual need to design for convenience and ease-of-use is exacerbated by the fact that just about everything on a buddy list's user interface directly affects people's privacy and productivity. Subtle shades of meaning can change substantially a user's perceptions.

Making the myriad notifications, states, features and options clear within the application's small window and without diluting its usefulness is tough. The next section is about these design challenges and the social issues in which they are embedded.

This section looks under the hood, at the server side. Creating a basic buddy-list system for a small number of users is beguilingly simple. Creating an architecture that can keep millions of concurrent users notified of each other's status is a daunting technical challenge.

Alongside advanced routers, buddy-list servers could become the next generation of phone-system switches. Given that potential, extensions of small-scale systems are unlikely to survive. Ambitious planning and thoughtful execution should pay off.

**Release 1.0**

20 June 1997
Client/server or peer-to-peer?

If you were going to build a buddy-list system, would you design it with a single server, multiple communicating servers or peer-to-peer clients that make minimal use of servers? There are many tradeoffs, though we favor the highly distributed, peer-to-peer model. The systems that occupy the extreme ends of the architecture spectrum are Ubique’s VP Internet Buddy (single server complex; page 17) and Activerse’s Ding! (peer-to-peer; page 20).

Both extremes have strengths and weaknesses. In a monolithic, single-server system, the server knocks everyone off the air when it goes down. Such a system is also a target for hackers, who could try denial-of-service attacks (i.e., flooding the lone server with spurious client requests so it can’t service real ones). Also, corporations are unlikely to want to rely on a system that is hosted outside their walls.

At the other extreme, Activerse’s Ding! clients look at the servers only to make initial contact with other clients. After that, they deal directly with one another. If for some reason a machine doesn’t respond at its old IP address, the machine looking for it will query the servers again for updated information. Although it is intuitively appealing, the completely distributed model hasn’t been proven to scale yet for buddy-list applications. At the pace this industry is attracting users, such proof or dis-proof shouldn’t be far away.

Other systems lie between the two extremes. For example, Mirabilis’ ICQ relies heavily on a central server complex, but when ICQ clients communicate with each other, they do so directly. Pager, the system iChat is building, will support a federated server architecture, where servers communicate with each other and forward user status information as needed.

It is likely that hybrid systems will be the rule. Even then, designers face major architectural considerations. Buddy-list systems have to decentralize. The question is, how much, and for which functions? If people can register on different servers, either there has to be a common name space, or the servers have to communicate so that people on one server can buddy with people on another.

Show me the patterns!

The key to choosing architectures is predicting future usage patterns, which means understanding people’s needs, preferences and behaviors.

Unexamined assumptions can get in the way of thinking clearly. The phone system seems like the natural model for thinking about buddy lists, but is it? For example, you have to pay extra to have an unlisted phone number. By default you get listed in a directory available to all -- and now searchable online. This assumption is loaded with implications for people’s sense of privacy, as well as for how often they will be solicited by strangers, which increases the noise level and reduces these tools’ usefulness.

There are a few ways of examining this question. One perspective is narrowcast or 1:1 vs. broadcast. Do you believe buddy lists will mostly support small workgroups and personal ties, or will they become a transport for large-scale, real-time broadcasts? Do buddy lists create one big audience or many small ones?
Another perspective is inclusion vs. exclusion. Which pattern of buddy-list use will prevail: private conversations with trusted friends in which people selectively invite others to come in, or public conversations with friends and strangers, where participants block people out?

What is big?

Scale is relative. Sure, it's useful to be able to buddy with anyone in the world, which argues in favor of competition that leaves one surviving service provider or standards that allow all of them to interoperate. But for the members of relatively stable, small workgroups, scale may not matter at all. They may be able to live happily with a system that encompasses only them. The privacy and productivity they have may outweigh other benefits.

Of course, nobody is a member of only one community of any kind. Inevitably, different communities will adopt different buddy list systems, which again raises the issue of dominance vs. interoperability.

Maybe one company will come to dominate the broad market for visible buddy lists that include broadcast messages and several will carve out comfortable niches creating power tools for private communities of practice.

You there? You there?

The components of all buddy-list systems keep in regular touch, but each system has its own rules for how often and in what direction those communications happen. In client/server systems, the clients generally ping the servers every minute or two. In peer-to-peer systems, the clients ping one another at similar intervals.

A few systems send messages only when the client's status changes. While this reduces network traffic substantially, it also leads to inaccurate information when one party's link is down or its machine crashes, since it couldn't have sent out a message that it is off the air.

Most buddy-list systems communicate mainly over UDP, the User Datagram Protocol (often using a UDP "heartbeat," which sends a small packet with status information on a pre-set rhythm). Unfortunately, IT security experts don't like UDP, because it opens an uncontrolled real-time stream between two points on the Net. One way around this problem is to use HTTP, the standard Web protocol, which is much more likely to make it through corporate firewalls but is considerably slower.

A few systems use proprietary TCP/IP protocols, which probably won't make it through firewalls and seem eminently hackable. On top of everything else, the fact that buddy lists are so new means they are ripe for attacks, because their inevitable loopholes haven't been found and plugged yet.

Identity management

Most of the buddy-list systems ask users to pick a unique user name when they first register. If the name is already taken, they must try again, until they have found one for themselves. They can't change the name afterwards without creating a new account and having to re-enter and re-authorize all their buddies.
That's a problem with the AOL Instant Messenger, as well as one of its main virtues: You can communicate with any AOL member. That means when you create your user account, you have to find a unique user name among the eight million registered users and the millions who have left. Plus, of course, each user is allowed up to five user names.

ICQ assigns each new member a unique ID number that the member need not memorize, publish or type in anywhere. You can call yourself anything you like, and anyone buddying with you can change your name from their perspective. If you want to name your stockbroker "shifty," he need never know about it. This seems like the best configuration of all.

It would be useful if buddy lists coordinated with a computer's certificate system and other identity attributes, both local and across the Web. This would make it possible for someone to keep her identity consistent while switching applications.

Wanted: New servers

Most directory servers can't function as buddy-list servers. They are designed to hold static information that doesn't need to be updated that frequently. Buddy-list status might change all the time, especially if it offers details such as when your screen saver is on or you're on the phone.

It is likely that buddy list development will lead to a new server -- call it the notification server -- that drops into a software slot next to the Web, directory, e-mail, certificate, FTP and other servers in Netscape's SuiteSpot or Microsoft's Normandy. Microsoft has submitted to Internet standards bodies extensions to LDAP (the Lightweight Directory Access Protocol) that add dynamic event information.

Offline use

Buddy lists are principally about real-time communications, but it would be helpful if they could send messages to people who are offline. However, few do. ICQ can hold messages and deliver them at the next available opportunity. The messages are stored in the sending machine's client software, not on the ICQ server.

Activerse solves this problem a different way. If you use Ding! to send an instant message to someone who isn't online or goes offline while you're sending, Ding! automatically rolls the message to e-mail. Of course, this changes the tenor of the message. Perhaps the buddy list could check the e-mail server when it logs in, fetch any converted messages and present them as it normally would have, making use of an existing, reliable store-and-forward system.

Mobile use

Mobile use of buddy lists isn't about use while in motion, though that is not as far-fetched as it may sound, but rather about use from different physical places. Why should you have to retype all your contacts when you're working from a different desk?

Seamless "roaming" is a function of where the information is kept (on a server, not the client) and how the system views new installations of its
client software. If all the information is on the server and the client is a Java applet, it's much easier to sit down at an arbitrary Net-connected computer and have your buddy list active right away.

Just to make things more interesting, it would also be nice to use your buddy list when you are offline, which favors storing information on the client. The best answer is probably a graceful and transparent synchronizing of buddy list information between client and server.

Interoperability

It should come as no surprise that none of the buddy lists covered here interoperate. That is, a user of OnLive!'s LiveList can't interact with a user of Excite's PAL or Mirabilis' ICQ in any way other than cut-and-paste. The biggest players have little incentive to cooperate. They want to build the most market presence as quickly as possible. Interoperability may be the key competitive lever for late entrants and second-tier players.

In the meantime, things could get ugly fast. Witness our plight this moment: Most of our buddies are in ICQ; we need AOL's Instant Messenger to see our friends inside AOL and PAL to see our friends on Macs. Each list is growing. It's entertaining, but only for a short while.

Activerse and PeopleLink are working to define some interoperability protocols for buddy lists now. Though neither company has put its system in public use yet, their work is promising.

Jeff Bone, Activerse's co-founder and cto, makes his case for agreeing on standards soon. "The buddy-list market today is like the e-mail market in 1990," he says. "It's balkanized. There's no interoperability. As long as this is the case, the adoption curves will be much flatter than they could be." One of the first standards that Activerse has proposed is WhoDP.

WhoDP?

The Who Datagram Protocol is a lightweight "presence" protocol that allows entities -- in the case of Ding!, people -- to publish dynamic information about their online presence to subscribers. WhoDP adopts a naming and addressing scheme based on URLs (e.g., whodp://ding.activerse.com/john@doe.com). WhoDP also specifies lightweight "I'm still interested" packets that clients send one another as often as is warranted by network performance and application requirements, currently twice a minute.

The fact that WhoDP follows URL conventions opens many possibilities. For example, WhoDP could be part of your e-mail signature file, giving people immediate access to your presence server. Or, more likely, giving them access to a WhoDP address that discloses only as much about you as you wish.

Activerse expects some directory services to extend their data models to include WhoDP-style addresses, potentially precluding the need for a Ding!-specific centralized directory.

More protocols

The Lotus groupware team, led by Irene Greif, has been working on what it refers to as "awareness systems." In the process, it developed an Internet
protocol for client/server coordination called the Notification Service Transport Protocol (NSTP). The group presented NSTP at the last Computer Supported Cooperative Work conference and has published the specification. Curiously, it has not implemented the protocol by publishing an application that uses it.

Microsoft hasn’t jumped into the buddy list/notification standards game yet, though it has chewed at the edges of this problem with various efforts, including its Internet Location Service (formerly the User Location Service), push Channel Definition Format, Microsoft Internet Relay Chat protocol, Net-Meeting work-sharing application, various directory efforts and DirectPlay multi-player game lobbies. As we mentioned earlier, the company is proposing LDAP extensions that carry state information useful to buddy lists.

Netscape hasn’t jumped in, either.

There is a danger in locking in protocols and APIs before knowing what the base feature set and optimal architecture for buddy lists should be. We are cautiously optimistic that competition and voluntary agreements will counteract each other, leading to long-term interoperability.

DESIGN AND SOCIAL IMPLICATIONS

Faxes. Voicemail. E-mail. Now buddy lists. Are we just running away from communication media as soon as they become overcrowded? Are buddy lists exciting because they’re new and full of only the people we most want to be in touch with?

Here’s another good question that has a similar answer. Are buddy lists just souped-up e-mail? They have a lot in common: a list of names, personal messages, attachments...

Turbo stealth e-mail?

Buddy lists are different from e-mail in important ways that change the nature of communication. Buddy lists are more immediate, casual and transient than e-mail. We will use buddy lists with fewer people, in much smaller circles. We can’t help receiving mail from strangers once they discover our e-mail address (OK, we could write a new filter for each person, deleting his mail automatically, but what a bother!). It’s easy to block, be invisible to or just not authorize people you don’t want on your buddy list.

E-mail will certainly survive buddy lists. The two forms of communication will simply form part of a hierarchy of media that ranges from instant, lightweight messages and calls (via buddy lists) through longer and longer-lasting messages we refer to often and forward to others (e-mail), and to more polished documents we either publish or present (Web pages and presentations).

Buddy lists also provide a needed mix of privacy and spontaneity that chat rooms and e-mail don’t offer. Private chat rooms are easy to create, but the public perception of chat is of chaotic rooms in which nobody is saying anything worth listening to. E-mail is convenient, but it’s still cumbersome. You have to find or remember a person’s nickname or e-mail address,
then tab through a subject heading and type a message body. If you want to suppress your signature file to make the message less formal, that’s three more mouse clicks.

**Lasting changes**

Buddy systems are the ultimate word-of-mouth applications; they spread like spores in a strong breeze. It doubtless helps that all of them are free right now.

There is already a little life-cycle to buddy-list usage. Many first-time users are enthralled. They use the systems intensely and enthusiastically evangelize their friends, family and colleagues. Eventually, the novelty wears off, but for many the effects are long-lasting. It’s hard to use the word "permanent" when buddy lists have been around such a short while, but we believe the long-term effects will be profound.

"To have people online all the time has revolutionized the way I work," says Web developer and ICQ user Emily Davidow. "When something is urgent, I use the buddy list. If the message is long or includes an attachment or has to go to people not on the [buddy] list, I use e-mail." Davidow continues, "It’s great for file transfers, too. You receive a request and an acknowledgment; it leaves you a full written record. I use ICQ more frequently than I would have used e-mail. It’s more convenient for those three-word messages. It has definitely cut down on my phone calls."

Buddy lists offer the ultimate in call completion -- all for the low cost of your Internet connection. The phone companies have to be listening.

**New social protocols**

Getting good at using buddy lists will be challenging, too. How do you keep all the beeping and messaging from interrupting your work?

What happens when you get a request to buddy from someone who you don’t want knowing when you’re available, much less what you’re up to? You could simply turn them down, but that would be a pretty explicit put-down to some people. You’re saying, in effect, "You’re not my buddy." Some buddy lists offer individual settings, so you could accept the invitation, then go "invisible" on the person permanently.

When you’ve just had several messages with one person, then set your "do not disturb" sign, is that a subtle insult to the person you were just speaking to? We haven’t yet developed common expectations about buddy lists.

**Presence, privacy and disclosure**

Often you are online, but you’re not at your PC. Most buddy lists can’t indicate the difference (though ICQ can be tied to your screen saver). In principle, a ubiquitous computing-style proximity sensor could provide that information, as could a cordless or cellular phone. It would also be useful if the system could tell others when you are on the phone (though the slow pace of computer-telephone integration may keep that from happening for a while). Officemates often look at their phones to see which lines are tied up before ringing one another.

**Release 1.0**

20 June 1997
Unless you like living life in a bubble that's visible to all, you don't really want everyone knowing where you are or what you are doing. There are a few people whom you would love to offer more detail, perhaps including a query to your calendar so they can see whom you are meeting with or a real-time reading of your whereabouts. The ability to disclose selectively such information on a person by person basis is critical.

"Buddy lists are a collection of bilateral treaties. The group exists only implicitly and may not know all its own members."

-- John Patterson, Lotus researcher

The dark side(s)

There's a good reason that AOL's original buddy-list feature was nicknamed the "stalker list" by some. Its default setting allows people to add other AOLers to their buddy list without the other party's knowledge. If you dig through the interface a bit and read the text carefully, you can turn the setting around, so that only people you want to buddy with can watch you. Many unsuspecting AOLers have found out about the default the hard way (including our Mom, who was puzzled when suddenly people started to Instant Message her as soon as she logged on, in an effort to keep her from posting to the widely read political boards).

AOL has learned from its early efforts. AIM (AOL Instant Messenger), the service's new, external buddy list, has a feature that allows people to play traffic cop for one another. Any subscriber who is subjected to abusive or unwanted messages can add warning points to the offenders' account IDs by clicking on AIM's "Warn" button. The complaint information is stored on the server, where its owner can't change it. The total shows up as a cryptic rating on incoming Instant Messages: "Warning level 20%." You can't yet avoid all messages from people with warning levels over x percent, but it's clearly a desirable feature.

All techno-gadgets have an effect on the pace and quality of our lives. All too often, it's negative. Remember the days when your business card was sleek and uncluttered with pointers? Remember when you could vacation without getting contracts and presentations faxed to your hotel; when you could watch good theater without your cell phone or pager going off? As Michael Crichton warned at the PC Forum last March, time spent in front of keyboards and on cell phones is not quality time with your family.

Buddy-list service providers will have to be explicit about what they do with data they collect (see Release 1.0, 2-97). By their very nature, buddy lists can be privy to a lot of sensitive personal information. It's one thing to get a copy of someone's address book; it's quite another to know exactly with whom and when an individual communicates.

SOME FUTURES

To support practice, not merely chat, buddy lists need to be linked to more and more applications and to one another. As the initial wave of fascina-
tion with buddy lists passes, developers and users are realizing that buddy lists need to touch many other technologies. The list is long and includes address books and directory services (see box), all kinds of messaging (e.g., voicemail, e-mail, video), calendaring and scheduling, personal information management, identity management, Internet telephony, computer-telephony integration, multicasting, groupware, PDAs and Web technology in general.

One thing all these industries have to get right is the transitions between all the features and applications. How people include additional members in a conversation or take a conversation from text chat to telephony will matter a great deal. Social norms and legal issues abound, too. When and how people can record conversations -- which is illegal on the phone in the US if you don't notify the other party -- and whether it is OK to publish a transcript are all undefined territory.

Three tiers

Here's a three-tier approach that positions buddy lists relative to the other electronic places where we find people's names.

Your buddy list holds the names of people you interact with all the time and with whom you have trusted relationships. It's not a big group; you won't allow just anyone on your buddy list. Most likely, your list will include family members and members of your immediate work or project team. It needn't stop there. It may well include your clients or suppliers, or other peers in your industry worldwide.

You will be constantly negotiating with yourself whether to put people on the buddy list or not. You may keep people there for only a few days, during the peak portion of a project. People may pop onto your buddy list for only the time it takes to do a task -- say, shop online.

The second tier is your address book or PIM, which is the place you put all the names you harvest from daily communications.

The third tier is the collection of larger directories, internal and external, that speak LDAP. When you need to find an address for someone whose name you know, you'll search these directories. You might also search them to find out who occupies a specific position at a company, such as who does customer service for your accounting package or who sells parts for your motorcycle.

Why should the buddy list be separate from the e-mail client? It makes all the sense in the world for the two to be one application. The buddy list is the almost-minimized version of the full address book. Checkmarks in certain address-book columns indicate which people go on your buddy list. There's no duplication of effort.

Other devices, other uses

One can imagine a buddy list kept active on the small screen of a cellular phone or PDA (finally! an application that's the size of a PDA's screen!).

Release 1.0 20 June 1997
The data streams needed to know who is online are only marginally bigger than those needed to know your phone is on and where you are located -- which is useful added information, by the way.

If you generalize the buddy list capability, you see that it’s a great notification mechanism -- a feature that the Internet has been missing all along. Of the companies profiled in the next section, only Flash Communications has seized on that idea.
WHO'S WHO

Here is today's crop of buddy list creators, listed roughly in order of market entry, with a miscellaneous category at the end. Almost all of their products are in beta testing, so our comments about usability should be taken with a grain of salt. Half the companies -- AOL, iChat, Ubique and OnLive! -- have appeared in *Release 1.0* before, but in other contexts. The others are upstarts with good ideas: Activerse, Flash Communications, Mirabilis and PeopleLink.

Unless stated otherwise, you can assume that each system has the basic functions you would expect from a buddy list, including simple status settings (and we do mean simple), user-defined groups, floating windows and small status icons for when the application is minimized.

Few of them offer offline messaging, easy file and URL exchange or links to third-party applications, features that will be increasingly important to have. Mirabilis is the feature leader; it was also the first buddy list for the open Internet.

Buddy-list developers come from many markets, including Website software, Internet telephony, online conferencing and chat systems. Each sees the problems and opportunities differently.

The companies are using a broad variety of revenue models. A few will rely on banner ads (PeopleLink). Some will sell client licenses (Activerse); some servers (iChat); and some services (Ubique). Some may sell souped-up "pro" versions of their client software, with a basic client available for free. AOL will treat its external buddy list as a way of both giving outsiders a taste of AOL technology and connecting its members to Internet users.

Finally, we should note how dominant Windows 95 implementations are in this market. Almost all the developers promise Windows 3.1 and Macintosh versions, but there is an astonishing lack of support for Macintosh today. Only Virtual Places/Excite and iChat have Mac versions.

AOL: ANOTHER SMART MOVE

As we wrote in January 1997, AOL put buddy lists in the public eye when it introduced the feature with its 3.0 software release. Over 6.5 million of its 8.5 million subscribers have already adopted the internal buddy list, making AOL the market leader. The adoption rate is spectacular. Just in January, we reported 4.5 million buddy-list users.

More recently, AOL did another smart thing: It released a free buddy-list program called AOL Instant Messenger (AIM) that works on the open Internet. Now all AOL members can buddy with outsiders, and outsiders can see better inside AOL -- as long as they use AOL's buddy list. Outsiders can also use AIM with each other, paying no regard at all to AOL.

For people who have friends on AOL but don't have AOL accounts (or have them and don't use them much), this is a great feature. They can sit outside AOL, track when their friends log in and out, and communicate with them. It
works the other way around, too: AOLers can see when their external buddies get on and off the Internet.

Special features

The Instant Message interface in AIM is the same as the excellent one inside AOL. (In fact, to someone accustomed to AOL’s environment, the experience of using a small piece of AOL outside AOL can be a little eerie.) There are a few small but significant differences -- features added to protect users from unwanted intrusions.

The first extra feature, called "knock-knock," is a dialog box displayed before an incoming message is presented to request permission to display it. The second feature is the cumulative warning-point system described above, which puts a "warning level" note next to the incoming messager’s name. We should note that AOL intends to enhance the version of AIM in current use quickly, probably with features such as better user states (e.g., away, busy) and support for clubs and associations.

But...

All that said, AIM is confusing to use, especially to start building your collection of buddies. It is hard to tell when or even whether you have invited someone to "buddy" successfully. Nothing in the setup procedure or software interface explains where to start. The intuitive first action, simply typing in user names, offers no feedback. This uncertainty will cause some people to balk and quit trying.

AIM has fewer features than the other systems described here. Users can’t pass files or invoke arbitrary third-party applications such as Internet telephony or Microsoft’s NetMeeting. They can embed Internet URLs in IM text, but, perhaps more importantly, they can’t embed URL equivalents to places inside AOL (imagine a URL format aol://times/nationalnews/12:34 that goes to a specific post in a bulletin board inside AOL -- or to a signup or day-rate page if you’re not a member). If AIM is supposed to drive traffic back into AOL, it must be easy to do.

As mentioned before, new users have to pick a unique name from the many already consumed by AOL subscribers (see page 8). This will further deplete the already shallow pool of names for both AOL and AIM users. Also, because it works similarly to the internal AOL buddy list, AIM has the same weak invitation and authorization protocol. The knock-knock and warning features don’t repair or offset this weakness.

Once you get AIM working, it is solid and elegant. The fact that a huge service such as AOL is pioneering such a leading-edge application bodes well for AOL users. AOL hasn’t decided yet whether AIM will remain free permanently, but for now, it’s the best cheap online entertainment available. Now if AIM had better privacy and a few more features....

--------

1 Like a phone company re-using phone numbers, AOL recycles screen names six months after accounts are cancelled.
UBIQUE: THE KISS PRINCIPLE

Ubique is a veteran developer of multi-party online technology. We first reported on its Active Mail project in February 1994, as the company was being formed. That August, we described Ubique’s novel Virtual Places system, which features chat windows that dock around your Web browser and avatars you can customize and use to move around Web pages and interact with others. No other company has implemented Ubique’s idea that the entire Net is the environment.

Two years ago, AOL bought the company, which went silent for a while. Although it is still a part of AOL, Ubique is functionally independent. AOL’s internal and external buddy lists don’t use Ubique’s technology.

Now Ubique is slowly re-emerging. Its Virtual Places technology is much improved. More importantly, the company has developed a subset of that functionality into a complete buddy-list system, called VP Internet Buddy, which runs on the same server. The Buddy is significant in two ways: It’s the only buddy list being private-labeled today (see box), and it’s the only one that has good support for both Windows 95 and Macintosh.

**Excite puts Ubique’s buddy list to work**

Ubique’s first major licensee is Excite, which recently announced Excite PAL: the Personal Access List. Ubique did the customization, but Excite designed its interface. One of the things Excite can do as it customizes VP Buddy is put a search button on the interface, where it is always at hand.

This is an interesting move for a search-engine company. Excite was already a licensee of Ubique’s Virtual Places technology and had put a private-branded chat service based on VP on its Website for general use. That system is complete with avatars, gestures and all the bells and whistles that VP offers. With PAL, Excite continues to move into applications that generate personal interaction. Look for many more Websites to add such features.

Ubique’s design philosophy is to keep the system simple. Instead of loading VP Buddy with functions, it will invoke other applications, as well as its own component architecture.

**Similarities and directions**

The first versions of Ubique’s buddy list have thin invitation/approval procedures and loose support for privacy. Like AOL’s internal buddy list, Ubique’s VP Buddy lets users create an "allow" list that lets only certain people buddy them. But the default setting is full visibility -- anyone can put you on his buddy list. You can also be invisible to people selectively, but if you don’t know the person has you on his buddy list, you don’t know to stop them from seeing you.

Ubique’s licensing arrangements are per concurrent user. As we described earlier, Ubique has the most centralized buddy-list architecture of all.

**Release 1.0** 20 June 1997
Everything passes through the server complex, which is made up of a VPserver and several VPmultiplexers. Ubique has done stress testing on the complex to simulate 100,000 concurrent users with no problems. Given the growth rate in this market, the question of how Ubique's system will scale to large numbers of users in real life should be answered in the next year or two.

MIRABILIS: SEEKING EVERYONE

Mirabilis' ICQ buddy list, officially still in beta, has attracted over 1.5 million registered users, making it the second largest, behind AOL's internal buddy list, and the largest that works over the open Internet. The system regularly hosts 30,000 simultaneous users, with peaks of over 90,000. We have been using it quite happily for four months to work on the PC Forum and our Website with remote colleagues.

ICQ is well designed and full of features. It has a robust and wizard-aided invitation/acknowledgement procedure, a wide range of features accessible with a right-click of the mouse, links to many third-party applications and good aesthetics that blend tongue-in-cheek humor with economical use of screen real estate.

For example, a flower in your system tray indicates whether you're connected to the ICQ server or not. (This flower becomes a barometer for your Internet connection as well as the ICQ server. When the flower is red and flashing, either the server is down or your link has crashed. When it is green, all is well in the world.)

Detailing

ICQ offers a nice range of degrees of presence, from offline/disconnected to privacy/invisible (you're online but can't be seen), do not disturb, away (which hangs a little note over your icon) and online/connected. The system also lets you customize your settings with individuals, so you can disappear to one person while remaining visible to the rest.

One feature the system is missing is groups, which are promised in an upcoming version. It is unclear how the current interface will support them, though, since it currently separates buddies into those who are online, offline and waiting for authorization. Adding groups may require substantial redesign of the interface.

Mirabilis offers a unique function it calls paging. Every ICQ user automatically gets a Web page from which anyone can send notes to them that show up as ICQ messages. This is a handy feature, but it bypasses the otherwise strong privacy protections built into ICQ. Worse, pages show up in the same place where system messages live, making them seem official. Despite safeguards to prevent sequential or automated downloads of information, the paging feature, combined with sequential IDs and the white-pages directory on Mirabilis' site, are prime targets for spammers.

Already, Mirabilis has had to counter several rumors in its user base, most notably one that claimed it was time to pay for ICQ. The rumors spread as word-of-mouth through ICQ's forwarding feature.

Release 1.0

20 June 1997
One for all

Mirabilis is the strongest proponent of a single server. Its system uses one finely tuned server complex to maintain state information. Clients ping the server every minute to ascertain their status. As its user base grows, Mirabilis' engineers continue to tweak and tune the hardware to get more performance.

The ICQ server isn't needed for all communications. When clients send things to one another, they do so directly. Nevertheless, if buddy lists become nearly as popular as telephones, we can't imagine that one server will do. The phone system has many huge switches, and people use their telephones only a small proportion of the time; buddy lists need to pass information much more frequently.

Even though ICQ is the most full-featured and businesslike of current buddy lists, its current centralized, service-only architecture may limit its appeal to businesses. In July, Mirabilis plans to announce an intranet server for corporations.

The company

Mirabilis, an Israeli company run by young entrepreneurs, is backed by private investors. Arik Vardi is its ceo and chief server developer, Sefi Vigiser is the president and chief creative director, and Yair Goldfinger is the vp of R&D. The company had only four employees four months ago; now it has 25.

Mirabilis is racing to build market share, without concern for short-term revenues or interoperability with other players. It figures that first-movers will have a big advantage in this market. With an estimated 40 million connect hours for this month, Mirabilis believes ICQ will be one of the most-used online applications. Mirabilis has not made definite choices yet about pursuing either an ad-sponsorship model or having two versions, a useful free one and a more powerful paid one.

ICQ is available now for Windows 95 or NT; Java and Macintosh versions are due next month. There are also special versions that link to Microsoft NetMeeting, V-Chat and standard chat.

ICHAT: PROMISING, BUT EARLY

Austin-based iChat is another familiar face. We wrote about it in January, as a developer of conversation technology for the Net, and in March, as a company presenter at the PC Forum. Andrew Busey, iChat's president, has ambitious plans for his products.

The company started as a vendor of chat systems built on its flexible Rooms server and featuring a variety of clients: plug-in, Java and plain HTML. Recently it made an early beta release of Pager, its buddy-list software, which is available on the Web for both Windows 95/NT and Mac. The Pager server is separate from the Rooms server.

Unfortunately, the early Pager interface isn't ready for prime time. Its features are muddy. For example, the invitation/acknowledgement process is
hard to figure out. Even after you have invited people, you can't tell when you've connected with them until you actually chat. The privacy and visibility model is relatively weak. Anyone can add you; you have to "ignore" them. Pager has a distracting rotating logo, but it does display message threads nicely.

Short-term interface nits needn't be terminal flaws, of course. The next release of the client promises to be much more usable. iChat's engineers are also working hard to turn Rooms' and Pager's back-ends into federated, communicating servers so buddy-list queries can find their way around a network of distributed servers. The company also has a strong market presence already. There are over 800 Rooms licensees who will doubtless be interested in Pager. Look for many of its current customers to launch branded buddy-list services using Pager.

**ONLIVE!: HELPING GROUPS TALK**

One of 1995's most memorable products was OnLive!'s Traveler, which lets multiple participants interact with one another in a 3D space using lip-synching avatar heads (see Release 1.0, 11-95 and 3-96). The following year, OnLive! launched Talker, another offering built around Traveler's multi-party Internet audio capability. One of Talker's features is an occupants list, so participants know who is in a "room" with them as well as who is speaking at any given moment. That feature, separated from Talker and redesigned, is now a buddy-list system called LiveList that is available on the Web for free download. (Again, LiveList is available for Windows 95/NT only.)

**Let's talk!**

LiveList nicely complements the company's other two offerings, which are "destination" products -- they are based on specific virtual places. LiveList is only about people. The program makes heavy use of peer-to-peer communications, much as Activerse does (see below), and can host group text chats of up to 30 people. It is also firewall-friendly.

Long-run, LiveList's great differentiating feature is its ability to do multi-party audio by invoking Talker. For now, parallel to its development of Talker, OnLive! has created a more traditional, standards-based audioconferencing server, the Community Server 2.0. The company will soon release large-scale audio and data conferencing enhancements to Microsoft's NetMeeting called LiveMeeting (with support for T.120 at first and H.323 later). It also plans to deliver a corporate directory server based on LDAP v3.

Although OnLive! hasn't set pricing on its new products yet, it expects the servers to follow the lead of e-mail system pricing, which averages about $50-75 per seat.

**ACTIVERSE: A RUSH OF GOOD IDEAS**

Although it is still in product development, Activerse is full of innovative ideas and clever details about buddy-list technology. It is also a leader.
in developing buddy-list interoperability standards. Austin, TX-based Activerse's primary focus is on managing network presence. Personal communication of many kinds is a natural result of shared presence. Activerse's attention to such communications is reflected in its architecture.

As we mentioned earlier, Activerse's Ding! minimizes its use of centralized servers by having clients communicate directly with clients whenever possible. Ding! clients check a server once using WhoDP2 to establish contact with others. The server gives them each other's IP addresses. After that, every time Ding! starts, it tries the old IP addresses. Only if that fails does it try the server again to get new ones.

Organizations will be able to create their own Ding! directories by using Activerse's server, Ding! Switchboard, which maps dynamic information to more static directory information, which is available through LDAP. The company will add features to the system through helper applications it calls Dinglets.

Nice details

The early Ding! interface makes it hard to distinguish between users who are offline and those who have chosen "do not disturb," but Ding! has another way to indicate status. A "Doing" field -- as in, "what I am doing now" -- lets people offer a little detail about their activities or whereabouts. For example, the Doing remark under your name might read "Out to lunch," "Alpha proposal" or "In meetings all day."

Ding! also has a feature called "the watcher list" that allows users to flip the lens -- to see who is buddying them at any moment. Another nice detail is that the native format for Ding! instant messages is HTML, which means they can contain anything that Web pages (or HTML e-mail) can contain. It also makes it easier to use the notes elsewhere on the Net.

Slow-perking Java

Ding! is written in Java, but as a local application, not an applet. That makes it harder to install, because the download file includes a Java virtual machine, making it 2.5 Mb instead of the more typical 500-800 Kb of other buddy lists. Java also limits the things Ding! can do. In fact, until the most recent version of the Java environment shipped, Ding! development was held up because many important features one takes for granted in the native Windows and Mac environments were unavailable.

2 A note for techies: This address allows a Ding! client to find a Ding! Switchboard for some other entity. The Switchboard is queried for the location of the requested entity and responds with a current IP address for that entity if it's online. If the requested entity chooses to publish its presence information to the requesting client, the two then begin to communicate via a UDP messages defined in the WhoDP protocol. Once a publisher-subscriber relationship is established, clients publish information about state changes (i.e., user "closes his door") to subscribers as needed, and clients "ping" each other on a configurable or dynamic basis.
Nevertheless, Activerse is happy with its choice of language and platform; it expects to have beta products available on the Net at the end of June. Once the product is finished, the company will follow Netscape's business model: Ding! will be free for non-commercial use and $30 per user otherwise, with volume discounts. Corporate accounts that buy many licenses will get the Ding! Switchboard free. Individuals will use public servers.

Choosing a new path

Activerse was formed in October 1996 by the merger of Active Paper, which developed Internet e-mail and Web browser software for Magic Cap, and In-traverse, whose founders had created a SmallTalk-based visual Web application system for ParcPlace/Digitalk called VisualWave. Before the merger, Active Paper sold its applications to General Magic.

From 1989 to 1992, Jeff Bone was a developer of large-scale systems at Sun. During this time, Bone became involved in the early development of MUDs, developing distributed server technology based on object migration and the first GUI MUD client, Mudtool. This experience led him to recognize the importance of online, real-time communication and the notion of presence. MUDs offer a sense of presence through their "who list," which tells you who is online in the MUD whenever you type the command "@who."

Activerse's first prototype was a MUD-like system called Venue. Ding! is one of its features. Bone and his colleagues believe strongly in the metaphor of virtual places and will continue to develop them, but they feel that buddy lists offer a simpler starting point for ad hoc communications, because they don't introduce unnecessary navigation through spaces.

PEOPLELINK: BRANDS & ADS

PeopleLink has adopted a business model different from most of the companies profiled here. It will give away its software -- clients and servers -- and share the ad revenues, creating a multitude of ongoing revenue streams for itself. President and ceo Steve Glenn feels this model will at least guarantee the company some revenues right away.

The company's strength is in channels. Glenn has been criss-crossing the USA negotiating deals with ISPs and highly visited Websites to use PeopleLink as their buddy-list offering. Glenn can offer them a customized interface and a share of the ad revenues.

PeopleLink, an IdeaLab! company, is also pursuing associations, schools and other organizations that can benefit from helping their members or alumni find each other and interact more easily. PeopleLink can provide the organizations a secure space within which people can find others who went to the same university and are interested in the same hobbies, for example. In this effort Glenn can benefit from the groundwork of another IdeaLab! company, CitySearch, which has been approaching such organizations systematically by the thousands.

---------

3 MUDs -- Multi-User Dungeons or Dimensions -- are text-based worlds generally used for fantasy or role-playing games (see Release 1.0, 6-93).]
Behaviors and expectations

Glenn is quite realistic about what he’s doing. "Buddy lists may not fetch the same prices from advertisers as chat," he says, "because the application is minimized or hidden most of the time. Chat tends to stay open a long while, with plenty of time for impressions." But he thinks ad performance will be better on buddy lists, since clicking on ads won't mean leaving the chat (the ads send your Web browser somewhere and leave the chat intact), the chat is likely to be less controversial (since the conversations are consensual, not free-for-all) and PeopleLink can target its users better (because it knows more demographic information about them).

Privacy is high on Glenn’s list of priorities. He wants to ensure that PeopleLink communications are consensual, so he is building in various software safeguards. PeopleLink won't publish an open directory so people can't use it to send abusive messages. It will, however, offer people the option of listing themselves in interest groups for open discussion.

PeopleLink has some good ideas of its own, but it isn’t counting on being the technology leader. It is working with Activerse, eShare and iChat to develop interoperability standards in order to grow the market.

One of those good ideas is the BuddyBot, a sort of virtual automatic call distributor that companies can install to send "callers" to open representatives. Imagine getting to a Website and having it know you’re there and automatically put a new buddy-list name up for sales or customer service.

Getting here

When we first wrote about PeopleLink, it had two products: the PeopleLink buddy list and ChatCall (see Release 1.0, 1-97). The latter was a pay-per-call, anonymous party line service. One person would pay for both ends of the call; the phone-number information wouldn’t cross parties. The company recently sold ChatCall to another startup, freeing itself to focus on a single business.

The near term has been harder. Late in the development process of its first prototype, PeopleLink realized that the release of Java it was using didn’t support many user-interface features it needed such as text styling and cut-and-paste. To get a product in the market, PeopleLink licensed technology and incorporated it with its own design and technology. The beta version should be out by the end of the month. A second buddy-list system will be PeopleLink’s own code, written in Java. That’s when the company’s engineers will be able to show their stuff.

FLASH: AN MIT REUNION

At the core of Flash Communications are eight MIT electrical engineering and computer science graduates from the same undergraduate dorm a decade ago. They each took different paths through the decade, but regrouped in early 1996 to build a scaleable and firewall-friendly buddy-list system. They began product development in January 1997, after almost a year of debate and design. They hope to have an internal product by the end of summer and a shipping product by the end of the year.

Release 1.0

20 June 1997
Although Flash’s engineers have taken considerable trouble to ensure their system will make it through firewalls, the company’s biggest insight and point of differentiation is that it treats the buddy list as a two-way, general-purpose transport for more than information about people’s status. More simply put, Flash is building an instant notification service that could just as easily carry urgent stock quotes with action requests as information on a colleague’s logoff.

Because this brings it into competition with companies such as Diffusion, Intermind, Wayfarer and BackWeb, Flash has described what it does as the next generation of push, a statement we found disconcerting until we understood its intent. Flash is being designed so it can also provide dynamically subscribable groups that could easily contain information feeds.

Francis de Souza, Flash’s co-founder, grew up in more countries than Madeleine Albright sees in a good week abroad, including Ethiopia, Greece and the United Arab Emirates. De Souza sees five markets that should be interested in the kind of system Flash is building: international corporate communications, custom applications, brokers of high-value transactions, advertisers and tailored third-party services. Flash plans to put a free version of its system for use on the open Net, and sell it to corporations.

Like Jeff Bone MUDding at Sun, Flash’s founders had tasted a buddy-list-like system before. Theirs was an MIT program called Zephyr, which started as a weekend hack and eventually became part of the Athena research program. Because Athena also incorporates Kerberos security, Zephyr included relatively good authentication and identity management.

Early funding for Flash came from private investors, including Mort Meyerson, chairman of Perot Systems. The company is based in Kendall Square in Cambridge, MA -- where else would MIT alumni pitch their tent?

OTHERS

There are several other significant players in the buddy list field. While it is not our intent to be exhaustive in this issue, here are some snapshots of their efforts.

When you go to Firefly, it shows you the user names of the people who logged in most recently. It’s a simple, useful feature (disclosure: Esther Dyson is an investor in Firefly). Why don’t all Web servers have a "who’s here?" function?

Commack, NY-based eShare has such software. Its business model is most similar to iChat’s. eShare sells several products, mostly standalone chat

--------

4 Zephyr is a text-driven Unix program that uses commands such as zlocate and zwrite to let participants find one another (as long as they’re logged into the same server) and exchange short messages. For more information, see http://web.mit.edu/olh/Zephyr/Zephyr.html.
and threaded discussion systems for Web hosts. From smallest to largest, its products are Reunion, Connections and Expressions. The latter product includes a Website-level buddy-list feature. eShare was originally Interactive Marketing Services, a database marketing consulting and services firm specializing in financial services and travel.

Manhattan-based EarthWeb has been developing a small constellation of Java applications and servers that include constructs for chat, real-time notifications and much more. Early prototypes include a native Java object-oriented database and a moderated chat system with separate clients for moderators, speakers and participants that will be debuted this month. EarthWeb is also working on buddy-list capabilities.

The big guns

Then come four heavyweights that none of the other entrants should ignore: Lotus, Novell, Netscape and Microsoft. If Netscape or Microsoft were to include buddy lists in their browsers, for example, it would twist the market considerably.

Lotus has done considerable research in buddy lists and related fields, but remarkably little product development. Long a dominant force in asynchronous groupware, Lotus has to wake up more quickly and get in the real-time collaboration field with products of its own, not awkwardly appended systems such as Intel's ProShare videoconferencing system.

Buddy lists needn't be boring lists of names. In one Lotus mockup of call center teammates, the buddy list uses small pictures of participants in different poses. When a person is on the phone, the picture is, too. When he's away, the picture goes gray. The team members can send messages to one another or post more important ones, such as "Baltimore is experiencing major outages today" to a department-wide ticker. The communication elements in this example are built into the page, as embedded ActiveX or Java components.

One product that would seem absolutely natural for Lotus would be a buddy list that does for Notes what Instant Messenger does for AOL: create connections into the host system without having to run the full client software. With it, Notes users in the field could know when their colleagues were on their servers and interact with them.

The "N"s

Novell has an opportunity to alter its course substantially by jumping into the buddy list market. Standard directories, including Novell's NDS, can't act as buddy-list servers because they aren't set to manage dynamic status information.

Netscape hasn't made any visible gestures toward buddy lists, though such features would make its Constellation more useful and appealing -- never mind the browser itself.

The big "M"

Finally, Microsoft has been grooming NetMeeting to be the group-work interface of choice. In fact, some of its new features have buddy list-like
qualities. However, NetMeeting's user interface still leaves a lot to be desired, and it's not clear that it will be more appealing than a buddy list, whether it comes bundled with the OS or not.

Should Microsoft design a new buddy list client from scratch, a great feature would be for one client to support multiple back ends. That is, the client would know how to log in to various buddy list servers, then consolidate the results.

Steve Liffick develops things like buddy lists at Microsoft. "Overall," Liffick says, "centralized architectures are superior for these applications." One reason he cites is that peer-to-peer systems hand out IP addresses too freely. Another is the need for a central naming authority. Liffick is pessimistic about interoperability efforts, especially around sharing user name information. He expects that service providers will guard their name lists ferociously. (Activerse separates the presence information from the names directory, solving this problem.)

It's tricky to build a server complex for real-time notification; you really need different servers with different roles, including a name server, a login processor, a translation server for cross-platform file transfers and so forth. Luckily for the rest of the industry, Microsoft can't just jam IRC and Exchange together and have a workable solution.
RESOURCES & PHONE NUMBERS

Steve Vandegrift, Jeff Bone, Activerse, (512) 708-1255; (512) 708-1293; steve@activerse.com, jbone@activerse.com
David Gang, AOL, (703) 453-5947; dgang@aol.com
Udi Shapiro, AOL/Ubique, (310) 770-4898; udishapiro@aol.com
Nova Spivak, EarthWeb, (212) 725-6550; fax, (212) 725-6559; nova@earthweb.com
James Tito, eShare, (516) 864-4700
Francis de Souza, Flash Communications, (617) 864-1471; fax, (617) 577-1209; fdesouza@iflash.com
Andrew Busey, iChat, (512) 425-2200 x11; (512) 349-0005; busey@ichat.com
Steve Liffick, Microsoft, (206) 936-4179; (206) 936-6399; stevel@microsoft.com
Yossi Vardi, Mirabilis, (212) 358-4000; fax, (212) 358-4099; vardi@ibm.net
Bill Owens, OnLive!, (408) 617-3514; fax, (408) 617-7010; bill@onlive.com.
Steve Glenn, PeopleLink, (310) 581-4299; fax, (310) 581-0020; steveg@peoplmlink.com

COMING SOON

- Identity management.
- Online governance.
- Handling the info-flood.
- Market-based security.
- And much more... (If you know of any good examples of the categories listed above, please let us know.)

Release 1.0 is published monthly, except for a combined July/August issue, by EDventure Holdings, 104 Fifth Ave., New York, NY 10011-6901; (212) 924-8800; fax, (212) 924-0240. It covers PCs, software, the Internet, computer-telephone integration, online services, groupware, text management, connectivity, messaging, wireless communications, intellectual property law and other unpredictable topics. Editor: Esther Dyson (edyson@edventure.com); publisher: Daphne Kis (daphne@edventure.com); managing editor: Jerry Michalski (spiff@edventure.com); promotion manager: Robyn Sturm (robyn@edventure.com); circulation manager: Scott Giering (scott@edventure.com); executive assistant: Donna Rapisarda (donna@edventure.com); office manager: Helen Martin (helen@edventure.com); editorial consultant: William M. Kutik (kutik@edventure.com). Copyright 1997, EDventure Holdings Inc. All rights reserved. No material in this publication may be reproduced without written permission; however, we gladly arrange for reprints or bulk purchases. Subscriptions cost $695 per year, $750 overseas.

Release 1.0 20 June 1997
Please enter my subscription to Release 1.0 at the rate of $695 per year in the U.S. and Canada. Overseas subscriptions are $750, airmail postage included. Payment must be enclosed. Satisfaction guaranteed or your money back.

Name______________________________
Title______________________________
Company____________________________
Address____________________________
City________________ State________ Zip_____
Country____________________________

Telephone________________ Fax________
E-mail________________ URL________

☐ Check enclosed.
☐ Charge my
  ☐ American Express    ☐ Mastercard    ☐ Visa
  Card Number_________________________ Expiration Date_______
  Name on Card_______________________ Signature________________

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

Please fill in the information above and send to:

EDVENTURE HOLDINGS INC.
104 FIFTH AVENUE, 20TH FLOOR
NEW YORK, NY 10011

If you have any questions, please contact us at 1 (212) 924-8800;
fax 1 (212) 924-0240; e-mail us@edventure.com; www.edventure.com.

Daphne Kis
Publisher