Everywhere messaging

by C. Schmandt
N. Marmasse
S. Marti
N. Sawhney
S. Wheeler

By "everywhere messaging" we refer to the ability to send and receive electronic communication at any time and through a variety of means, including wired and wireless computer networks, voice telephones, and pagers. Our goal is to design messaging systems in which the receiver is always "on" and available, and messages are correctly chosen for unintrusive delivery. But even in the office, and especially out of it, message arrival must compete in the real world with other activities that place demands on users' cognition and for which message alerting may itself be a distraction. In this paper we consider four experimental projects in terms of their ability to meet everywhere messaging requirements of minimizing interruption, adaptation to the user, location awareness, and undemanding user interfaces. These projects demonstrate message filtering, location-specific delivery, flexible auditory alerting, and operation in, and monitoring of, a heterogenous networking environment.

The last decade has seen e-mail become ubiquitous, following on the heels of voice mail from the previous decade. Wireless technology has become pervasive, at least in urban areas, both for text and voice communication. We have increasing means of being contacted and the pace of both personal and business communication is ever faster. Although the spread of wireless telephony clearly indicates a desire to communicate at any place and time, we complain of information overload, spam or junk mail, and continual interruptions from unwanted telephone calls. Some of us deliberately invoke strategies to minimize these interruptions, such as answering e-mail only once a day, but at a price of missing valuable communication and, in turn, spending more time playing phone or e-mail tag with correspondents who are no longer available by the time we get their messages.

Gone are the days when work took place at the office and family life at home. For some, work implies several work sites, as companies have grown to include multiple domestic and, more recently, international locations, with work teams spread across time zones. Telecommuters may have times during which they are explicitly "working at home," but for many of us work also impinges on evenings and weekends with family. Even workers with a permanent office with a single desktop computer are sometimes overwhelmed with e-mail and other communication tasks. But mobile messaging is particularly difficult, due to many factors that include intermittent radio coverage, the small display of portable devices, and the use of such devices in the midst of other activities of daily life.

In this paper we discuss "everywhere messaging," the ability to send and receive messages at any time. Everywhere messaging is ubiquitous and always "on." Technology to support voice or text communication almost anywhere on the planet is becoming very real; subscriber services and user interfaces lag further behind. In this paper we raise some of the challenges

 0018-8670/00/$5.00 © 2000 IBM

for everywhere messaging, and then discuss a series of research projects exploring these problem areas.

Challenges for everywhere messaging

The main problem for everywhere messaging is to provide enough value to subscribers while minimizing annoyance. This is easier in the office, where we are already at work, harder at home, where we desire a bit more peace and privacy, and hardest in mobile situations, because these divert our attention and intrude in our social interactions. The acceptance of true everywhere messaging hinges on providing messaging services that address the four challenges described below.

First, messaging services must minimize interruption while ensuring that important messages are delivered to users in a timely manner. Active use of new communication technologies makes users vulnerable to undesirable interruptions. An observational study[1] evaluated the effect of interruptions on the activity of mobile professionals in their workplace. On average, subjects spent nearly 10 minutes per hour on interruptions. Although a majority of the interruptions occurred in a face-to-face setting, 20 percent were due to telephone calls (no e-mail or pager activity was analyzed in this study). In 64 percent of the interruptions, the recipient received some benefit from the interaction. This suggests that a blanket approach to preventing interruptions, such as holding all calls at certain times of the day, would prevent beneficial interactions from occurring. However, in 41 percent of the interruptions, the recipients did not resume the work they were doing prior to the interruption. It appears that in order to prevent even greater disruptiveness, there is a need for improved filtering.

Second, communication services must adapt to changes in user behavior throughout the day. This can range from suppressed beeping when the subscriber is engaged, to a personalized user interface conforming to communication patterns. Adaptation allows devices and their user interfaces to be responsive as a user's activity changes during the course of the day, delivering service appropriate to the subscriber's level of availability.

Third, the term "everywhere messaging" almost begs for location awareness. Universal connectivity means that communication can occur at any time, while the communicating devices can be located anywhere in the world. Location awareness could include location-based services, or delivering messages appropriate to activity characterized by location, such as "the commute home" or "walking in the park." Location awareness can be a particular form of adaptation, in that the behavior of a device can change according to the locale (being more attention-getting in a private place, for example).

Fourth, the user interface for everywhere messaging must operate in concert with the competing demands for the user's attention. Everywhere messaging places particularly severe requirements on the user interfaces of portable devices. Displays are likely to be small and hard to read, often used under poor lighting conditions. Speech interfaces may be more suited to mobile use but in turn place a higher cognitive load on the user, whether it be to understand speech synthesis, or keep track of a multiple choice menu or even which messages have been responded to. By definition we are busy when interrupted, and user interfaces must be sympathetic to the activities we are engaged in.

In this paper we describe four projects that provide solutions toward these challenges. *Clues* is an e-mail filtering agent that attempts to identify "timely" messages by analyzing the information found on a subscriber's desktop computer and the message transaction history. *Active Messenger* manages multiple communication channels, prioritizing the messages to be delivered to the appropriate channel, and determining the preferred channel for a mobile subscriber. *Nomadic Radio* focuses on a purely auditory user interface for a wearable computer, attempting to match the incoming message delivery to the subscriber's attentive state. *comMotion* focuses on "right-time right-place" messaging by tracking physical location via the Global Positioning System (GPS) and by learning travel patterns. As shown in tabular form in Figure 1, each of these projects offers different solutions to the four challenges for everywhere messaging.

Clues: Filtering for everywhere messaging environments

Clues uses the subscribers' personal information on desktop computers to prioritize messages for mobile access. By relying on information sources, such as a calendar or mail log, that change along with the user's plans and activities, Clues creates dynamic filters in order to identify timely messages.

Figure 1 Comparison of everywhere messaging strategies

— For position only —

	Clues	Active Messenger	Nomadic Radio	comMotion
Minimize interruption	Prioritization	Channel selection	Monitors conversation	Right time/place
Adaptive	Monitors activity	Monitors user activity over multiple devices	Responds to inactivity	Learns locations
Location awareness	Coarse, metro areas (calendar)	Virtual	Acoustic classification	Physical/ geographic
User interface (competing demands)	Prioritizes messages to save time	---	Scaled/alerting, voice control	Auditory user interface for mobility, voice control

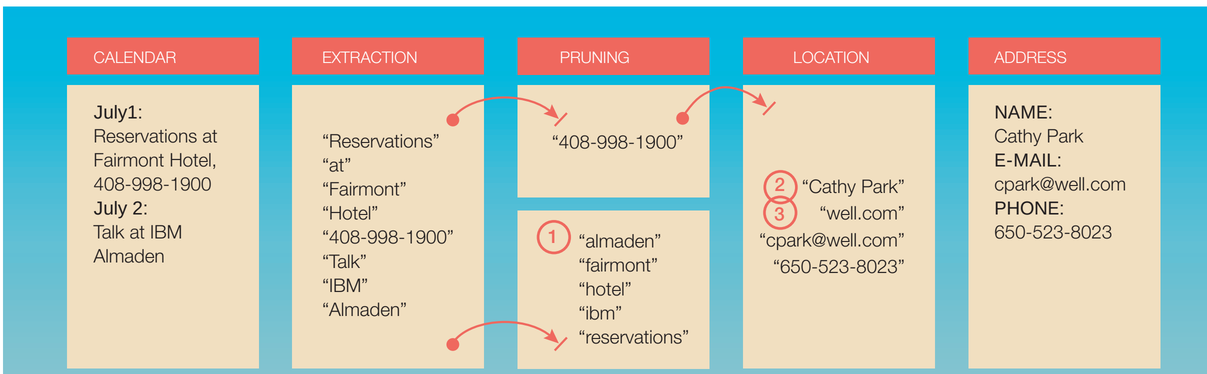
Clues was developed for use in a telephone-based system for e-mail access using speech recognition and synthesis,[2] in which a design goal was to deliver the higher priority messages first. Clues' approach to filtering messages for timeliness proved so successful that it was retrofitted into Phoneshell,[3] a touch-tone-based system that has been in continual use since the early 1980s. Clues was also used in asynchronous messaging systems where its role was to extend static filtering. Unlike most mail filtering systems, Clues contrasts short-term, timely information from long-term information that reflects stable user interests or social relationships. Some systems[4] allow users to specify rule-driven filters that classify and handle incoming messages. Other systems[5,6] dispense with user-authored rules by using knowledge of the user's past handling of messages. However, both suffer from drawbacks. Static rules are difficult to modify on the move. Users rarely remember to change the rules after returning to the office. Automatic rule creation requires a long ramp-up interval for creating an accurate user model. Clues divides the work of filtering, using both dynamic filters to capture short-term interests and a small number of user-authored rules to capture long-term interests, which are not dependent on time or place.

The processing steps in Clues are applied in stages. Information is first extracted from temporally organized data sources. Calendar entries are indexed by date and time, and a user-defined window of interest determines the granularity used in the analysis. Users often maintain to-do lists, and Clues assumes that such lists are, by nature, up-to-date and therefore relevant. Sent-mail logs supply names of frequent correspondents and subjects of interest, and thus are relevant as well, within a user-defined window of interest. Computer telephony applications supply data on outgoing calls, which may be associated with e-mail addresses via the address book. To implement this stage, a Perl script produces a list of unique words, proper noun phrases, names, e-mail addresses, and phone numbers contained within the appropriate time-window, from all information sources.

In the second stage, frequently occurring words such as "meeting" or "mail" are removed using "stop lists." Clues provides a rule-cancellation facility, which creates an additional short-term stop list. Words remain on this stop list for an amount of time that increases with subsequent cancellations. After a number of cancellations, the words are added to the permanent stop list. The final clue-gathering process is correlating information to generate additional clues from geographically organized data sources, primarily the user's Rolodex.** This process attempts to identify potential correspondents who are

Figure 2 Example of filter generation

— For position only —



RULE	EXPLANATION (PAGER)
1 ^Subject:,*[^a-zA-Z0-9]+Almaden([^a-zA-Z0-9]+)	The message matched on the string Almaden on the "from" line. The basis for this rule is a calendar entry for list item for Talk at IBM Almaden.
2 ^From:, Cathy Park	Messages from the San Francisco area are relevant, because you have a calendar entry that lists a phone number with the 408 area code. The person with this e-mail address has a telephone number in or near that area code.
3 ^From:, *@well.com	The area code 408 is relevant, because you have a calendar entry that lists a phone number with this area code or its surroundings. This domain name has been associated with this area from a Rolodex™ entry for Cathy Park.

geographically related to previously extracted phone numbers or domain names. After these steps, Clues generates rules to be used in the actual filtering process. The rules take the form of regular expressions matching targets in "From" and "Subject" header fields. The body of the message is not currently considered for rule matching. Finally, upon arrival of an incoming message, the rules are applied in sequence, to determine if the message is to be considered "timely."

Figure 2 illustrates the process described above, for hypothetical calendar and address listings. Assuming that the calendar entries are within a user-definable time window, Clues would generate a total of 19 filtering rules from this data set. Three of these rules, shown in Figure 1, illustrate the use of regular expressions for pattern matching. The first rule matches any incoming e-mail containing the word "Almaden" in a subject line. Explanations are generated at the time of rule creation, since intermediate data structures will be lost after the filtering rules are written to disk. As shown in Figure 1, the explanation includes the data source associated with the rule. The last two rules refer to correspondents and their e-mail addresses, geographically associated with the calendar entries. The explanations indicate the data source, as well as the rationale used to link the address and calendar entries.

Minimizing interruption. Clues does not specify how messages are to be handled. A timely message is relevant to short-term contexts, in which the benefit of delivery outweighs the cost of interruption. Systems such as Active Messenger and Nomadic Radio use this information in order to minimize user interruptions. Using this short-term relevance captured by Clues, applications can determine with more confidence when an interruption is appropriate.

Adaptation. Clues attempts to address two drawbacks of desktop-based e-mail filtering systems for mobile messaging systems. First, e-mail filtering rules are difficult to create or modify while traveling. Desktop users must typically either create rules using a filter "wizard," or create them by hand. Clues generates all dynamic filter rules automatically, without user intervention. Furthermore, the user may apply rule cancellation in order to refine the rule set. Second, in a mobile environment, static rules must be changed more frequently in order to remain current since, by definition, the mobile user's geographical context is changing.

E-mail filtering in Clues adapts to changing user contexts by using desktop information sources, such as a calendar or a to-do list, as a basis for rule creation. These data sources represent a partial model of user context. However, if a user forgets to update a calendar entry, or an item on the user's to-do list becomes irrelevant, Clues may falsely mark some messages as timely. Clues does not attempt to infer change in contexts from data sources where timeliness may be implicit. Other projects[7] have used memory-based reasoning to anticipate the user's actions for a given message. Adaptive approaches may seem attractive for lightweight rule creation, but the large number of training examples they require makes them unsuitable for capturing short-term interest. For instance, I am interested in company X when I have a scheduled meeting this afternoon with Mr. A from X, or when I have a message from Mr. A, a reply to an earlier message of my own.

Location awareness. In Clues, some filters are generated directly from temporally organized data sources. Other filters are created by searching through geographically organized data sources for items related to temporal clues.

Unlike other systems, which infer user location from GPS or network data, Clues looks for a special user-authored entry in the calendar to determine a user's location. Area codes are used as coarse locators in this system for information relevant for metropolitan geographical areas. The system includes a database of related area codes, including a place-name table for explanatory purposes. When Clues finds a location-entry in the calendar, it checks the address book for people who share that area code, or who are in an associated area. Clues looks for domain names associated with the area code; this allows mail from new correspondents who are most likely associated with an area to be marked timely as well. For example, if a calendar entry for today reads "fax: 415-506-9987" Clues will note that the area code is 415 and that the geographic area is "San Francisco," and will look for address entries in that area. The system finds Cathy Park in area code 650 and infers that cpark@well.com is important. In fact, the system will automatically mark messages from well.com as timely, since the area code is associated with the domain through the matching address entry. To prevent confusion over geographically distributed domain names, a stop list of domain names is used, and the cancellation facility provides an extra mechanism to alter the system behavior.

User interfaces for filtering rationale. Since filters are generated automatically by Clues, no user interface is necessary for rule creation. For the domain of e-mail filtering at least, Clues embodies the notion that the best user interface is no user interface. But users still need to be able to trust these invisible rules running in the background, and feel in control of the system.

For this reason, a recent enhancement to the system generates natural language explanations of rule sets. Explanations in Clues relate a timely message to the information source, as well as the process of location correlation. Explanations are generated at the same time a rule is generated, since all intermediate data structures would be otherwise lost after the program execution. Two explanations are generated, using a simple template; one for text paging systems and consoles, and another for speech output. Explanations are generated from a user-configurable template file. The following template, for example, is used to create an explanation for a filter rule generated by correlating phone numbers and locations.

> Messages from the [LOCALITY] area are relevant, because you have a [SRC_TYPE] entry which lists a phone number with the [AREA_CODE] area code. The person with this email address has a telephone number in or near that area code.

Items in brackets are replaced by the actual values when the template is used. The explanations enable users to understand why and how rules were created by the system. When a rule is not performing well, it can be cancelled by the user. Even if Clues delivers an error, users are more accepting of the system as a whole if they can understand why a particular error was made. User interaction with the system is confined to pruning automatically generated rule sets, rather than rule creation. For mobile users, the benefit of this approach is that far less effort is required to interact with the system.

Usage and summary. Three users rely on Clues for day-to-day e-mail filtering on several systems. Although no work has been done to gather accuracy and recall statistics, users find that Clues provides a useful "first pass" for selecting potentially useful messages. In a typical three-week period, User 1 receives about 1300 messages. User 1 relies on static filter rules to capture long-term relevance. These rules cover about 25 percent of incoming messages. Clues would identify 50 percent of messages flagged by these rules as timely. Of the remaining messages, 14 percent are marked timely. User 2 receives about 1200 messages, 12 percent of which are considered timely. User 2 does not use static filter rules, but relies on short-term Clues filter rules only.

Clues provides the means to prioritize incoming messages based on short-term relevance. We characterize these messages as timely. Rules are generated by examining sources of information on the desktop which may provide "clues" to the user's current context, such as calendar entries, a to-do list, or recent outgoing mail. After removing stop words, we correlate temporally relevant information with location-organized data sources. Clues enables systems with more demanding user interfaces to minimize interruption. Recent work on Clues has focused on automatic generation of explanations and user-controlled rule cancellation. Clues has been in active use for more than three years, and its initial success led to its deployment in additional systems, as described below.

Active Messenger: Message delivery in heterogeneous communication environments

Active Messenger (AM) is a server-based agent process that monitors a user's incoming e-mail messages, prioritizes them using Clues (described earlier), and forwards them to the available communication channels, e.g., pagers, fax machines, and phones. The basic forwarding rules are specified in a user preference file, but can be modified by the agent to adjust to the user's current situation. After having sent a message to the first channel, it checks the status of each message and channel, and waits for possible user reactions. If the user has not read the message after a certain time, the agent sends it to the next appropriate channel that is available, and so forth.

The agent process is created by executing a single Perl script, and consists of several steps. First, the agent loads all new messages into memory. Then it tries to determine the user's location with the help of caller identifier (ID) logs and UNIX *finger*[8] information. Subsequently, it checks whether each message has been read by the user, e.g., by parsing the user's mail spool file. Then it checks the status of the paging systems that are involved, e.g., if the last message sent has been received. In the next step, it sends all scheduled and queued messages for a channel, and schedules the messages for successive channels. Finally, the agent writes the log files, which allows the developer to track all program activities. Within the main script, there are also modules that handle the sending of messages to each channel. This includes a generic e-mail-to-fax transcoder, a Short Message System module for two-way text communication on cellular telephones, an e-mail-to-phone transcoder using text to speech, and specific modules for Canard[9] (a Media Lab proprietary two-way paging system), Skytel**, Iridium**, Pocketmail**, and a voice pager. Adding modules for new channels is a relatively easy task even for novice Perl programmers. A configuration file that the user edits during setup, and referred to as the "preference file," controls the actual operation of the agent. This simple ASCII text file contains all necessary information about the user's communication infrastructure and preferences: all e-mail addresses, phone and fax numbers, pin codes, as well as information specifying when a message of a certain type should be sent where. This ASCII file could be easily updated via a Web-based interface, if desired.

There are several projects and products that address the problem of message delivery in heterogeneous communication environments. The Mobile People Architecture[10] (MPA) is an advanced framework for connecting people, instead of their devices. However, not many channels are implemented yet, and the MPA does not include a scheme for trying multiple channels and gathering data on user reactions in order to guarantee the delivery of a message. iPulse[11] by Oz.com is a commercial proprietary system that is directed more toward service providers. Some of our own systems use Clues to filter and prioritize messages. However, given multiple possible delivery channels, these systems are not effective in deciding where to send a message. In Active Messenger instead, forwarding messages includes not only routing a message to the appropriate channels, but also using several channels sequentially over time, avoiding redundant messages, reducing information overload, and minimizing user interruption (as described next).

Minimizing interruption: channel selection. The agent process tries to minimize user interruption with a sophisticated channel selection mechanism. In the preference file, the user can define a channel sequence: rules that map each class of message (according to Clues) to a set of devices or delivery mechanisms. This allows the subscriber to control the number of messages as a function of how "reachable" he or she is. Messages from fellow workers may be routinely delivered while within range of the wireless network at the workplace, but unwanted for delivery to home by telephone, for example.

Additionally, the agent can modify these rules, bypassing channels that seem to be inactive or have not been used recently, and therefore minimizing user interruption. For example, AM does not forward e-mail messages to any other channel if the user is currently logged in to a computer and reading e-mail with a mail client. In another example, AM skips a pager service specified in a channel sequence if the last two messages sent there did not arrive. It can check the status of each message by logging in to the Web site of the service provider and looking up the relevant information. Once the agent has inferred that the user has read a message, it stops forwarding it to any more channels. An example of the channel selection process is illustrated in Figure 3.

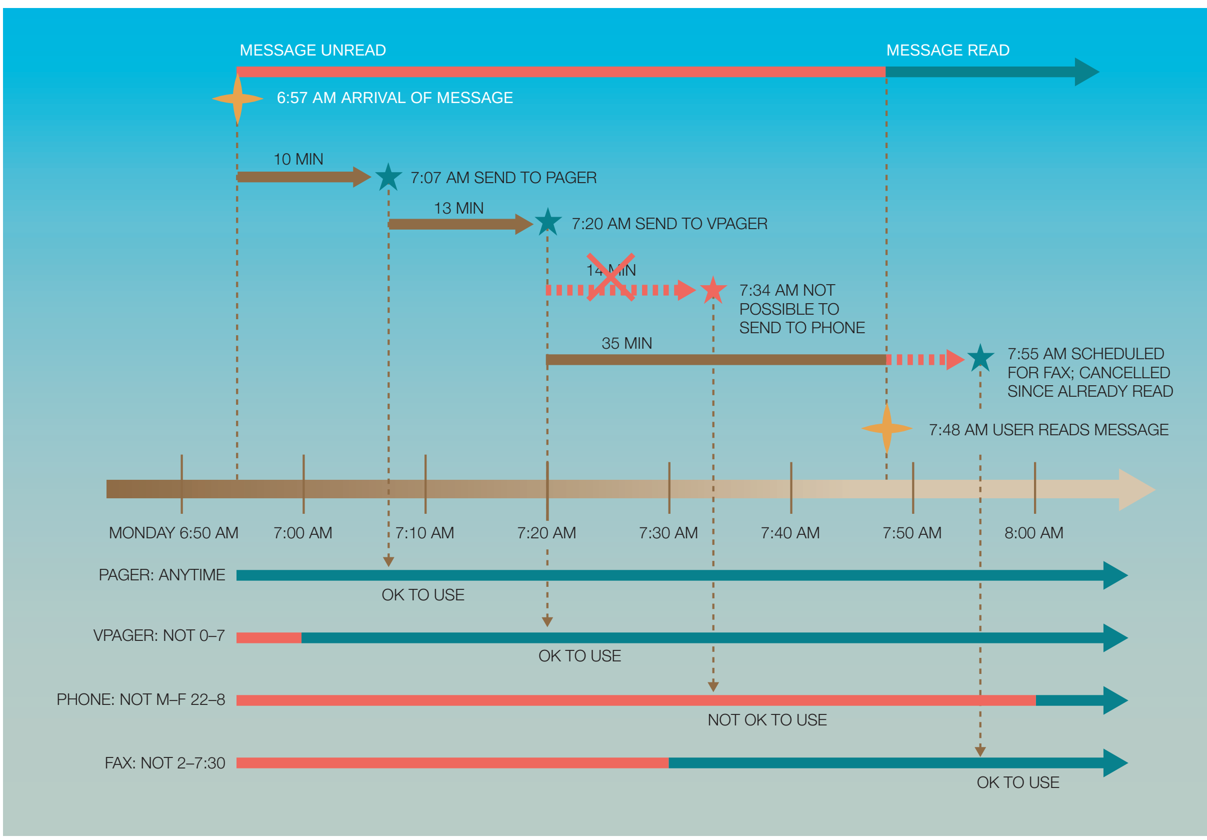
Adaptation by monitoring user activity. Active Messenger adapts to the user by monitoring the user's communication activities. For each channel, it keeps track of which message was sent there, which one has arrived, and when the last message was sent from this channel. Using this information, it modifies its forwarding behavior. For example, the user can specify a delay of ten minutes until a new message is sent to his or her pager. This delay makes sure that the user has enough time to read a new message if the user is at her desktop computer, without receiving the message on the pager shortly after. However, if the user is not logged into the computer, such a delay is unnecessary and reduces the immediacy of the channel. Therefore, the agent reduces this delay gradually, depending on the idle time of the user. If the user is idle for at least one hour, the agent sends the message immediately.

Location awareness through assessing channel usage. None of the devices currently in use is aware of its actual geographical location, but AM associates some delivery channels with virtual locations, such as "home," "work," or "out of town." In the preference file, the user can associate telephone numbers and computer names with virtual locations. Active Messenger checks regularly if the user is logged in to computers that are familiar to the agent, using the UNIX finger command and Activity Server,[12] a location server developed by the Speech Interface Group at MIT. When subscribers phone in to retrieve voice mail or use other telephone-based services of Phoneshell, we may detect a caller-ID number. Either a known Internet Protocol (IP) address or telephone number allows AM to modify its forwarding behavior accordingly, e.g., by sending a fax to the home number or office number.

Messages from the devices themselves, such as replies to e-mails, are another source of location information. Such traffic is known to AM for two reasons. First, although many of our devices have their own e-mail addresses, we use AM as a proxy that repackages replies so they all come from the same canonical (and public) mail addresses. Second, while traveling, the user accesses personal and external information by using the paging system Knothole.[13] Knothole uses two-way pagers to provide a bi-directional link to the subscriber's normal e-mail, with additional access to personal information on the subscriber's computer, such as address book and calendar, and external information via the Web and other sources, such as news, weather, and traffic reports. With Knothole, information is requested via short commands embedded in an e-mail message to the user's own e-mail account. Knothole thus supplies implicit hints about the user's whereabouts.

Figure 3 Example of channel selection. It is based on the user-specified channel sequence Important = Pager, Voicepager (13), Phone (14), Fax (35). The numbers in parentheses specify the delay in minutes before the channel is used. This channel sequence applies for important messages. Note that the channel Phone is skipped because at the scheduled time, the user does not allow the use of the channel Phone. Note also that the sending to the Fax machine is cancelled because the user has read the message already.

— For position only —



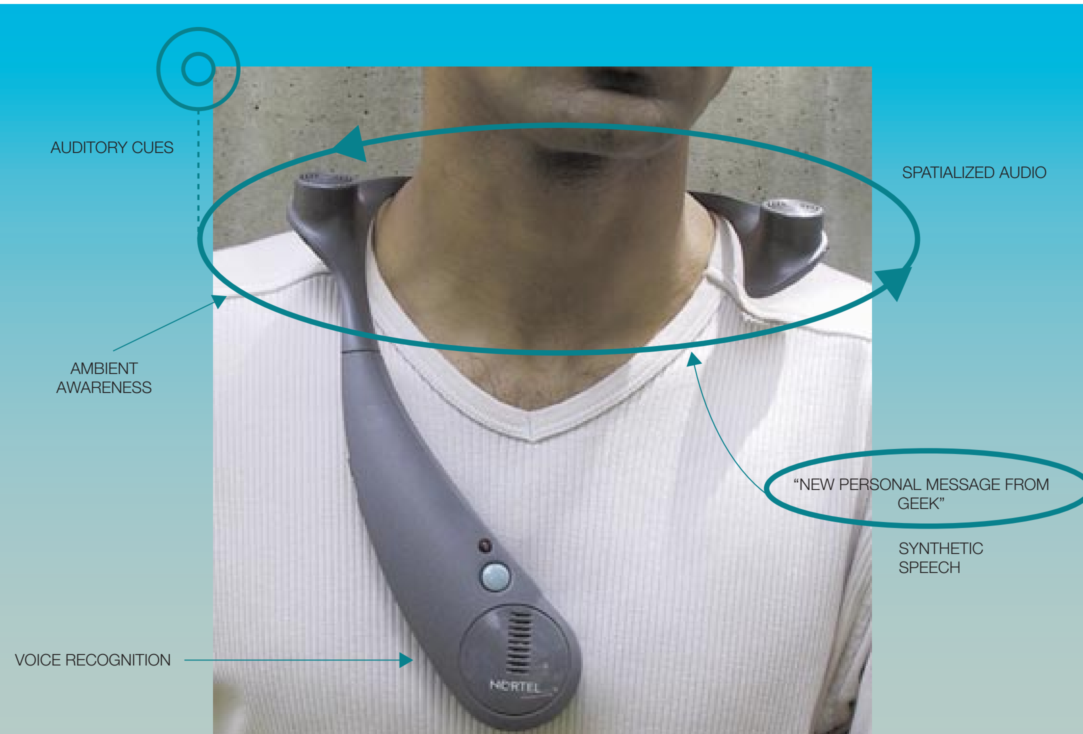
User interface. Active Messenger is a software agent that acts autonomously on behalf of the user. Before building AM, we relied on previously prepared filter files and scripts to invoke them from our devices, or over Phoneshell. Because AM is autonomous, no such direct interaction between the user and the agent is required. As an example consider the requirement to adjust the e-mail forwarding rules to a different location, e.g., from office to home. The user had to switch between different sets of forwarding rules manually, at the right time, and while within range of the right device, a task difficult to remember, and awkward to execute on the move. Instead, Active Messenger does this automatically, exploiting back channel information as described above. This allows the subscriber to worry about message content, not whether or not messages are received.

The agent also keeps track of messages that were sent to a channel but never arrived because the device, e.g., a two-way pager, was out of range. Once the pager comes back in range, the agent automatically resends the messages that were lost, unless the user has already read them on another channel. Using prior systems, the user had to request a list of recent messages, and then instruct the system manually to resend the missing ones. The agent now performs all these functions autonomously, without the user having to interact with it on an ongoing basis.

Usage and summary. Two people have used the agent described in this section continuously for approximately 12 months and find it an essential part of their everyday communication infrastructure. The two users have different filtering settings. User A,

Figure 4 The primary wearable audio device for interoffice use, the Soundbeam Neckset, with directional speakers and microphone*

— For position only —



* From N. Sawhney and C. Schmandt, "Nomadic Radio: Scaleable and Contextual Notification for Wearable Audio Messaging." Copyright 1999 Association for Computing Machinery (ACM). Reprinted with permission from *Proceedings of the CHI 99 Conference on Human Factors in Computing Systems: The CHI Is the Limit*, May 15–20, 1999, Pittsburgh, PA.

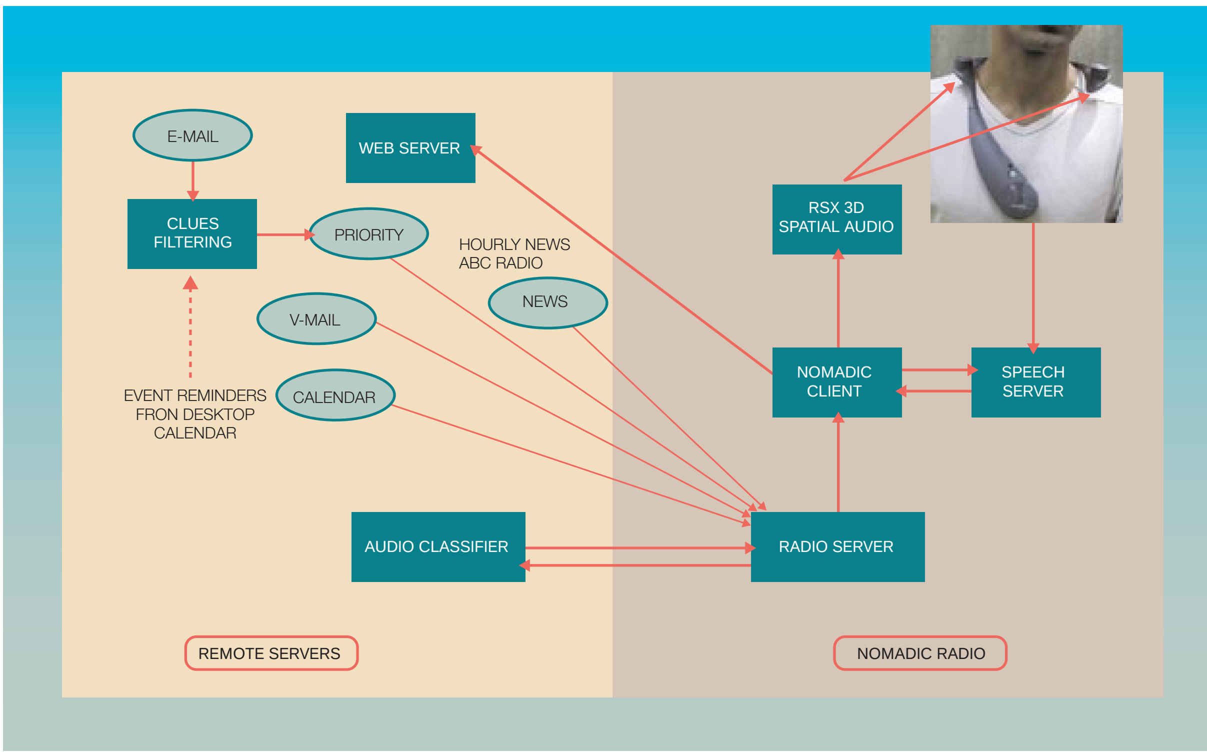
who gets on average 53 messages per day, lets the agent process almost 90 percent of these messages. User B gets on average 132 messages per day and lets the agent process 38 percent of them. Processing nearly 20000 messages per user over a duration of 12 months puts a high responsibility on the agent. These two users have depended on AM for over a year and have found that it significantly impacts their communication habits; they grow very frustrated when AM is not operational. While this endorsement is of limited value as these users are the system designers, AM must be providing value if they rely on it for handling their mail in work and social environments. Of course, we cannot be overly confident of how well the configuration and timing algorithms work in general, due to the limited user population. We plan to increase the number of users to get statistically significant answers to user evaluation questions.

Nomadic Radio: Everywhere messaging on the body

Nomadic Radio[14] is an audio-based wearable platform that unifies a range of personal information services, such as voice mail, e-mail, news broadcasts and calendar events for informing and alerting users on the move. To provide a hands-free and unobtrusive interface to a nomadic user, our goal was to use an audio-only wearable device. The SoundBeam Neckset, a research prototype patented by Nortel for use in hands-free telephony, was adapted as the primary wearable device. The system uses voice recognition and synthesis for interaction. It consists of two directional speakers mounted on the user's shoulders, and a directional microphone placed on the chest (Figure 4). The volume on the Neckset can be adjusted, so that audio output is heard primarily by the

Figure 5 Architecture of Nomadic Radio with the Audio Classifier module

— For position only —



user, while still being within conversational distance from others.

Nomadic Radio is a fully-functional prototype that consists of Java** clients running on a wearable (personal computer) PC and remote server components running on Sun SPARCstations**. Figure 5 shows the architecture of Nomadic Radio, including the communication paths between the Nomadic client and various remote servers. On the server messages are extracted, filtered using Clues, and periodically downloaded to the wearable device. A range of scaled auditory techniques (described in detail later) is used to present messages to the user. Users browse these messages and control the interface using synthetic speech feedback and voice commands (based on AT&T's Watson** recognizer). Speech interaction is coupled with button input for situations when speech input is socially intrusive.

Spatial audio is a technique by which the characteristics of sound sources are perceptually modeled so that a listener hears them at distinct locations around the head. In Nomadic Radio, spatial audio is rendered using the RSX three-dimensional audio application programming interface (API) developed by Intel, which is based on a model of head-related transfer (HRTF) measurements. Spatial audio is used to present multiple audio sources simultaneously, allowing listeners to segregate background and foreground information. The timing of spatial audio streams is synchronized with speech and auditory cues to deliver a coherent and well-paced presentation.

Minimizing interruption by contextual notification. In Nomadic Radio, whether messages are presented to the subscriber depends on the interruptability of the listener, the interval since the most recent interaction with the device, the message priority, and the auditory environment. These parameters, further described below, when weighted appropriately, constitute a notification model that estimates the value of information delivered, relative to the cost of interruption. The message priority provided by Clues-based filtering is used to weight messages proportionally to their timeliness. The usage level is determined based on the user's last interaction with the device (the more recent interactions are weighted higher). This assumes the users interacting with the device are more inclined to hear new notifications. The likelihood of conversation in the environment allows the system to infer whether the user is in a social context where he or she is less interruptible. An auditory classifier using hidden Markov models (HMMs)[15] is trained to detect if the user is speaking, if multiple speakers participate in a conversation, or if primarily room noise is heard. The classifier, which runs in real-time on a remote server, detects known classes with over 90 percent accuracy, and communicates with the wireless wearable device. When not connected to the server, Nomadic Radio continues to utilize the usage and priority cues while setting a higher threshold for likelihood of conversation. The notification model uses these contextual cues to dynamically scale the notification level (described later in this section) and the presentation latency of incoming messages. Greater latency allows listeners sufficient time to interrupt and deactivate a message before it begins playing.

Adaptation to user actions. The notification model described here is continuously adjusted to user actions, to gradually make the system more responsive and less intrusive. The user's prior responses to incoming messages are thus taken into consideration. After hearing an auditory cue for an incoming message, if a user takes no action the message is presented based on its urgency (as a summary or preview). However, if the user either activates or aborts the current message, notification for future messages is weighted accordingly. The system hence permits a form of positive or negative reinforcement of the notification weights within a narrow time window, by monitoring the user's actions during notifications. If aborted, all weights are reduced by a fixed percentage (default is 5 percent). If the user activates the message within 60 seconds after the notification, the playback scale selected by the user is used to increase all weights. If the message is ignored, no change is made to the weights, but the message remains active for 60 seconds during which the user's actions can continue to influence its notification. Hence the system continuously adapts message notification to both the ongoing context as well as user's actions within a narrow time window. While currently the user's actions primarily impact weights for subsequent messages, effective global reinforcement learning requires a model that generalizes a notification policy that maximizes some long-term measure of reinforcement.[16]

Location awareness via auditory context. comMotion, the subject of the next section, provides relevant spoken reminders to users based on their physical location. Although Nomadic Radio is not aware of the user's physical coordinates, it classifies location by the type of acoustic environment. Auditory context is modeled by extracting classes of sounds from various settings, such as people speaking, train station, hallways, indoor office, and the outdoors. Our pattern-recognition experiments with 2–3 hours of labeled recordings from outdoor sounds[17] shows that such an approach for inferring location context is feasible. For the purpose of scaled notification, Nomadic Radio only distinguishes among three classes: user speaking, sounds of people talking, and background sounds. Automatic classification of the auditory environment, however, can provide a nonintrusive alternative to physical sensors for determining the "kind" of place rather than its exact location, which may be more meaningful for certain mobile applications.

Scaled auditory user interface. Nomadic Radio enables a graceful audio experience for listeners by providing a range of techniques for awareness and notification of incoming messages (Figure 6). The motivation for such a scaled model is to reduce cognitive overhead for the user as he or she wears an "always-on" device everywhere. Ambient audio continuously heard in the background (sound of flowing water) reveals the operational state of the system and the status of incoming messages. The pitch is increased as messages are being downloaded to the device. This provides peripheral awareness and prepares the listener before the message is heard. When a new message arrives, priority cues alert the listener to its urgency inferred from Clues. A similar approach[18] used "e-mail glances" (sounds) to indicate category, sender, and content flags in messages. Finally, to indicate the identity of senders, e-mail messages are accompanied by VoiceCues, short segments of the sender's voice, extracted from prior voice mail and associated with their e-mail ID. Shortly after alerting the user to an incoming message, a spoken description summarizes the message header to indicate attributes such as sender, subject, category, priority and duration. If needed, a preview of the message extracts a few lines of the e-mail message or plays the first 2.5 seconds of a voice message. This is consistent with studies on voicemail usage,[19] which indicate that users generally listen to the first few seconds of a message to determine if

Figure 6 Dynamic scaling of an incoming voice message during its life cycle based on the interruptability of the listener. The message is presented at varying levels: from a subtle auditory cue to foreground presentation*

— For position only —

* From N. Sawhney and C. Schmandt, "Nomadic Radio: Scaleable and Contextual Notification for Wearable Audio Messaging." *Proceedings of the CHI 99 Conference on Human Factors in Computing Systems: The CHI Is the Limit*, May 15–20, 1999, Pittsburgh, PA.

it requires immediate action. An important message is played to the listener entirely, in the foreground of the spatial audio space such that it is heard predominantly over existing background audio streams. Such a scaled auditory scheme provides on-going awareness, fluid transitions between incoming messages and currently playing audio content, while allowing the listener sufficient time to attend and respond to incoming messages.

Usage and summary. Although the authors have been using and refining these techniques during system development, a preliminary evaluation was conducted involving two novice users, who used the system for 2–3 hours over a period of three days. Interviews were conducted and informal observations recorded, primarily focusing on alerting, rather than browsing, tasks. Even during limited use, we found the users were in general able to listen to the scaled notifications while attending to other tasks such as reading, typing or engaging in casual conversations. If, however, a message was *spoken* while the user was listening to others, this sometimes resulted in lost concentration. Here contextual notification, which was not activated in some cases, seemed more effective. In contrast to speech-only feedback, the users were more willing to listen to ambient and auditory cues while engaged in other tasks. They initially reported the overall auditory scheme to be, however, somewhat complex, especially when not familiarized with the cues. VoiceCues were perceived easily and users wished to hear ambient audio at all times to remain reassured that the system was still operational. To minimize errors in continuous speech recognition, we incorporated a form of "contextual recognition." Here the system briefly listens for voice commands when it expects some user interaction, e.g., during an incoming message. For additional details regarding the evaluation see Reference 14. We must emphasize that Nomadic Radio remains an exploratory prototype and is not used on a daily basis, unlike the other systems discussed in this paper. Hence additional evaluation work is needed in order to further validate our preliminary observations. The system has served, however, as a useful platform for exploring a variety of interaction and notification techniques, that can be applied to wearable devices, and mobile devices in general.

comMotion: On-the-go context-aware messaging

comMotion is a context-aware communication system for a mobile or wearable platform. The emphasis is on the mobility of the user, as opposed to the portability of the computer. As the user goes about daily routines, a location-learning agent, using GPS, monitors the user's travel patterns and learns the locations visited. The more the system learns about the user, the more it can adapt to him or her, delivering relevant information at the right place. Context-relevant reminders, to-do lists, content information and messages are triggered by, and sent to the user at his or her current location. Since mobility often implies being in a hands/eyes busy situation, the core functions of the system can all be accessed using speech.

The system consists of a client application running on the mobile/wearable computer. The client communicates wirelessly, over the Internet, to remote servers. There are servers for specific tasks, such as forwarding relevant e-mails and reminders, transferring Web content information and maps, querying and relaying position information to/from authorized users. The human-computer interface, on the client side, has both a speech component and a graphical one. The speech interface includes speech recognition and text-to-speech synthesis, and was developed using AT&T's Watson Software Developers Kit (SDK).

Most previous location-aware applications have used predefined content and/or predefined locations. C-Map,[20] for example, provides information to exhibition visitors based on location and individual interests. CyberGuide,[21] a collection of intelligent hand-held tour guides, provides information to tourists based on their position and orientation. CityGuide[22] enables a user to see his or her position on a map and request restaurant and hotel information. Augment-able Reality[23] allows users to dynamically attach digital information, such as voice notes or photographs, to the physical environment. Although comMotion can also provide predefined content (such as maps with locale information), its main feature is the delivery of dynamic user-defined content (such as the updated shopping list) delivered when the user is at the relevant learned location (near the grocery store). Hence neither the content, nor the locations, are predefined.

Minimizing interruption by just-in-time delivery. Reminders, to-do lists, and Web content can be delivered wherever the user is located. The delivery is triggered when the user enters the appropriate context: physical location, date, and time. A to-do list is associated with each learned location and may be site-specific or shared. For example, the same grocery list could be used whenever the user shops at any one of three different stores. The items on these lists can be text or recorded audio. The user could be walking down the street, record a thought, and have the system deliver a reminder next time the user is at work. Reminders can also be received from other people via e-mail.

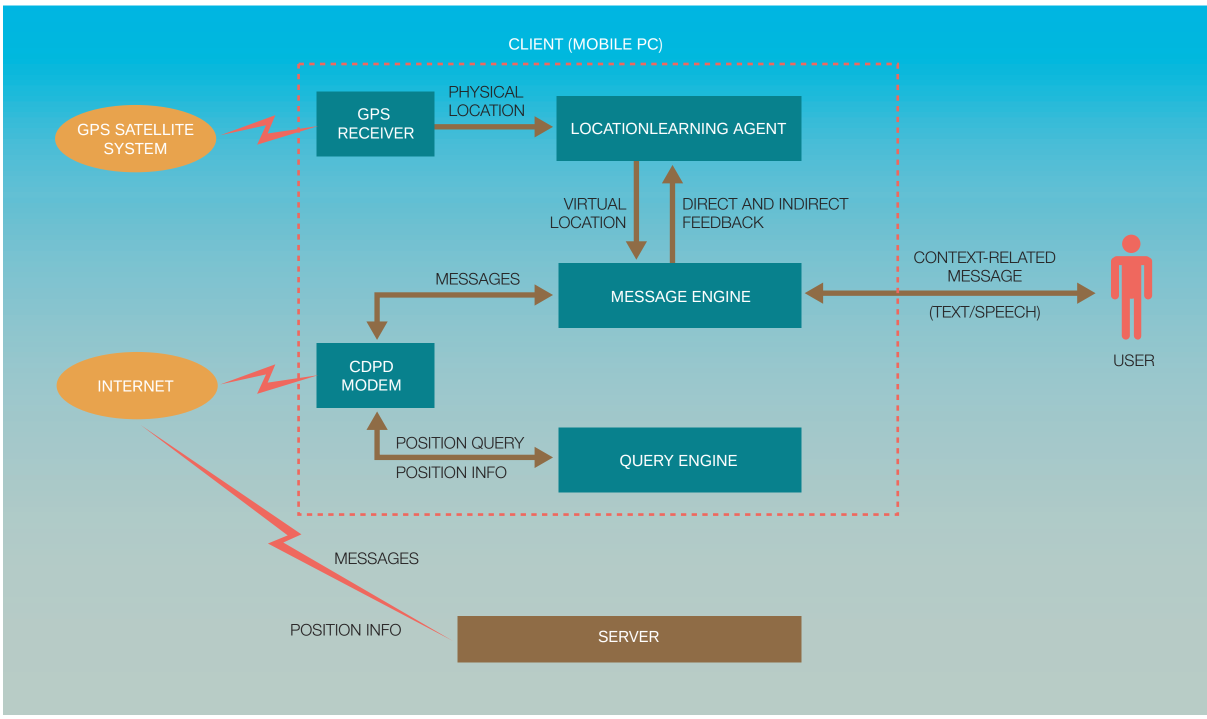
Web content information can be similarly delivered and typically includes weather reports, headline news, and theatre listings. The user can, for example, request to have the headline news delivered at the time he or she leaves home on weekdays. The request need only be specified once. The user simply specifies the desired service, whether he or she wants it when arriving or leaving the specific location, and whether the request applies to weekdays and/or weekends, or to a specific day. Calendar entries with important scheduled events for that day are delivered when the user arrives at work.

Adaptation to user by location learning. Latitude and longitude coordinates are obtained via the Global Positioning System (GPS) receiver connected to the client device (Figure 7). The stream of data is continuously analyzed in order to identify locations routinely visited by the user. Once a user has been "seen" three times (configurable) at a new location, the user is prompted for a location name, which can be either provided immediately or tagged at a later time while viewing the location on a map. By naming a location, the user indicates to the system that the place is in fact important. At this point the geographic location is converted to a virtual one (such as "work" or "home") and a to-do list is immediately associated with it. If the newly identified location is of no interest, for example a bus station, the user can indicate that it should be ignored. No initial configuration by the user is necessary since the system incrementally learns the user's frequently visited places. And as the user's routine changes, the system adapts and incorporates new locations. This location learning module was tested for two users over several months of daily use.

Until very recently, the generated GPS data was, by design, limited in its accuracy, having errors of 100 meters on the average (this restriction known as Selective Availability was turned off by the U.S. Department of Defense on May 2, 2000). Even data from a static receiver left for several days would fluctuate and not converge on definite latitude/longitude coordinates. This noisy data made location learning problematic, so comMotion also used the fact that GPS signal is lost in most buildings. The current prototype identifies only buildings as routinely visited locations, and thus places such as the user's home,

Figure 7 Architecture of the comMotion system

— For position only —



office, bank, bookstore, etc., can be learned. However, if the user jogs every morning in the local park, the system will not identify the park as a frequented location. Now, that GPS accuracy is no longer intentionally degraded, future versions will be able to learn outdoor locations and routes.

Location awareness. Location awareness is the essence of comMotion. The system uses location in three different ways.

The main feature of the system is knowing where the user is located, keeping track of what the user has to do, and triggering reminders at the appropriate location. Once the user's routinely visited locations have been learned (as described previously), information relevant to the location can be delivered. This could be a reminder to shop for groceries when in the proximity of the grocery store, an e-mail with a book recommendation from a friend when near the library, or a weather report when leaving home.

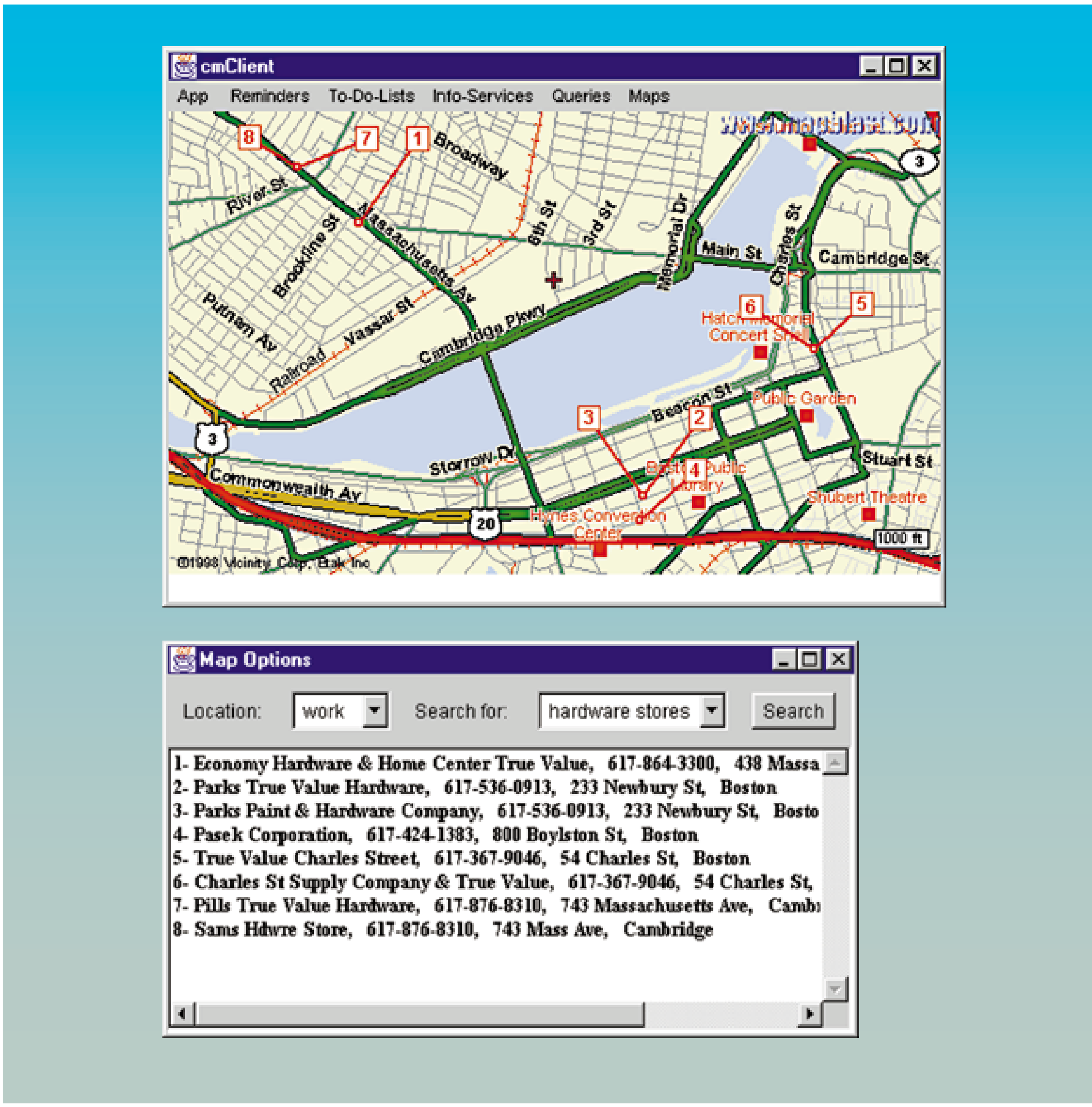
comMotion also provides awareness of locales or landmarks in the user's whereabouts. The map and locale data are downloaded from the Web upon request (Figure 8). This information is most useful when the user is not in a familiar neighbourhood. For example, one might look for a bank while driving to the airport in an unfamiliar part of town.

Awareness of the location of others is relevant for a community, such as members of a family, or a group of colleagues. The whereabouts of a comMotion user can be queried by another comMotion client or via the regular e-mail system. Authorization to access location information can be set per location and per user, giving full control and privacy. The queried user client can log who requested location information and the time of the request.

Auditory user interface. When the user nears a location that triggers information, the user must be notified. When mobile, the user's hands and eyes are dedicated to a navigating task, such as driving. Therefore, notification is provided by auditory cues, delivered in a minimally distracting manner. Different cues can be selected and associated to the different

Figure 8 Map indicating current location and locale information (the same locale can appear under different listings in the database)

— For position only —



types of data: to-do lists, reminders and subscribed information.

comMotion has a graphical user interface supporting all system functions, as well as a speech interface to a subset of functions. Speech recognition is error-prone, especially in noisy environments. Reading a message is faster than playing its audio recording, and it is difficult to take an auditory "glance" at a to-do list. Moreover, certain information is more suitable for graphical display, for example maps. However, it is essential that the on-the-go user have access to the core functions. The speech interface enables basic list management as well as message creation and retrieval, and navigation. The to-do lists and reminders are browsed using speech commands. Text items and reminders are synthesized to speech, and recorded items are played back.

Usage and summary. comMotion minimizes interruption by delivering messages at the right place, at the right time. It adapts to an individual by learning about routinely frequented locations. It is location-aware: it delivers information relevant to the user's current location, it extracts information about other locations in the area, and it can supply the user's current position. comMotion needs an auditory user interface since it is meant to be used by a mobile user. The comMotion system is a prototype whose location learning module was tested on data collected by two users over several months. A small field study, done by users on foot, indicated that there is reticence associated with speaking in public places, but this may not occur when traveling in an automobile.

Conclusions

Successful deployment of everywhere messaging requires integration of wireless devices and services in our already complex lives. In order to justify the added burden, everywhere messaging must solve more problems than it creates. We have considered four research projects and described the various ways in which they address the four challenges of everywhere messaging. These findings, shown in tabular form in Figure 1, are summarized below.

Subscribers, as well as their friends and colleagues, do not wish their wireless devices to interrupt too often. Clues minimizes interruptions by identifying those messages that can be safely ignored, or have low priority. Active Messenger only sends messages to the appropriate devices, and reduces the number of messages sent when its subscriber is less accessible. Nomadic Radio listens in and determines the appropriate time to deliver a message. comMotion waits until the proper place to alert its user.

Devices used throughout the day can become more personal and valuable by adapting to their user. Clues adapts to the user's schedule and communication history through the rules it generates. Active Messenger adaptation takes place at the device level. Nomadic Radio adapts to its user's responses to message alerts; busy users will not respond to less urgent messages. comMotion adapts by learning its user's frequently visited locations and travel routes.

The location we visit often provides a strong hint as to our activity and availability. Clues derives a coarse sense of location from data found in a subscriber's calendar. Active Messenger infers a virtual location, based on device characteristics such as caller-ID and device of origin for information requests. Nomadic Radio has no implicit sense of location but uses audio analysis to characterize the type of location. comMotion has the most refined sense of location, as the subscriber's actual physical location.

Clues provides no explicit user interface but does provide explanations for rule generation and thus builds user confidence. While Active Messenger relies on the user interfaces of its constellation of devices, Nomadic Radio provides a gracefully scaled level of auditory alerting tuned to the message and the state of the user. Both Nomadic Radio and comMotion are meant to be used in speech-only modes, which allow operation in "hands and eyes busy" situations.

Although their contributions are different, these systems are clearly complementary, and each could incorporate aspects of the others. In fact Clues provides infrastructure for both Active Messenger and Nomadic Radio. Active Messenger's sense of virtual location would be enhanced by knowledge of physical location. Hopefully these, and other connections, will be enhanced in further research. Active Messenger relies on commercially available wireless devices, but a more sophisticated receiver could incorporate Nomadic Radio's scaled auditory alerting.

Do these systems work? Some of them, Clues and Active Messenger, have been in continuous use by a very small community of users. This should not imply suitability for the general population yet, but does suggest that the underlying technology is feasible. Nomadic Radio and, so far, comMotion are best classified as working prototypes that let us perform a first pass within an iterative design of the user interface and the service specifications.

We do not expect to see devices, or services, closely matching these projects to appear as commercial products in the immediate future. But we are convinced that the challenges we outlined must be addressed by such future devices for everywhere messaging to gain widespread acceptance.

Acknowledgments

Matt Marx originally developed the Clues filtering scripts. The authors would like to thank the anonymous reviewers and the editorial staff of the *IBM Systems Journal* for significant suggestions as to the structure and details of this paper; they deserve much credit for its readability.

Cited references and note

1. B. O'Conaill, and D. Frohlich, "Timespace in the Workplace: Dealing with Interruptions," *Proceedings of CHI '95*, ACM, New York (1995).
2. M. Marx, "CLUES: Dynamic Personalized Message Filtering," *Proceedings of ACM Computer Supported Cooperative Work*, ACM, New York (1996), pp. 113–121.
3. C. Schmandt, "Phoneshell: The Telephone as Computer Terminal," *Proceedings of the ACM Multimedia Conference*, Anaheim, CA, ACM, New York (August 1993).
4. S. Dorner, Eudora: Bringing the P.O. Where You Live, Qualcomm, Inc. (1992).
5. E. Ly, *Chatter: A Conversational Telephone Agent*, master's thesis, MIT, Cambridge, MA (1993).
6. J. Rennie, *IFile: An Application of Machine Learning to Mail Filtering*, http://www.ai.mit.edu/~jrennie/papers/ifile98.ps.gz (1998).
7. M. Metral, *A Generic Learning Interface Agent*, S.B. thesis, MIT, Electrical Engineering and Computer Science, Cambridge, MA (1992).
8. The UNIX *finger* command displays information about users on a given host. The host can be either local or remote. For example, *finger stefanm@media.mit.edu* will display information about the author for his computer account at the Media Lab.
9. P. R. Chesnais, "Canard: A Framework for Community Messaging," *Proceedings of the First International Symposium on Wearable Computers*, Cambridge, MA, IEEE, New York (1997), pp. 108–115.
10. G. Appenzeller, K. Lai, P. Maniatis, M. Roussopoulos, E. Swierk, X. Zhao, and M. Baker, *The Mobile People Architecture*, Technical Report CSL-TR-99-777, Computer Systems Laboratory, Stanford University (January 1999). May also be found on line at http://gunpowder.stanford.edu/~laik/projects/mpa/publications/TechReport/html/TechReport.html.
11. See the iPulse homepage at http://www.oz.com/ipulse/ index.asp.
12. S. Manandhar, *Activity Server: A Model for Everyday Office Activities*, master's thesis, MIT, Cambridge, MA (1991).
13. See the Knothole web site at *http://www.media.mit.edu/~stefanm/pager/*.
14. N. Sawhney, and C. Schmandt, "Nomadic Radio: Speech & Audio Interaction for Contextual Messaging in Nomadic Environments," *ACM Transactions on Computer Human Interaction (ToCHI)*, (to appear).
15. B. Clarkson, N. Sawhney, and A. Pentland, "Auditory Context Awareness via Wearable Computing," *Workshop on Perceptual User Interfaces*, 37–42 (1998).
16. L. P. Kaelbling and M. L. Littman, "Reinforcement Learning: A Survey," *Journal of Artificial Intelligence Research* **4**, 237–285 (1996).
17. N. Sawhney, *Situational Awareness from Environmental Sounds*, Media Lab Technical Report, MIT (June 1997). May also be found at http://www.media.mit.edu/~nitin/papers/Env_Snds/EnvSnds.html.
18. S. E. Hudson and I. Smith, "Electronic Mail Previews Using Non-Speech Audio," *Proceedings of CHI '96*, ACM, New York (1996), pp. 237–238.
19. S. Whittaker, J. Hirschberg, and C. H. Nakatani, "All Talk and All Action: Strategies for Managing Voicemail Messages," *Proceedings of CHI '98*, ACM, New York (1998), pp. 249–250.
20. Y. Sumi, T. Etani, S. Fels, N. Simone, K. Kobayashi, and K. Mase, "C-MAP: Building a Context-Aware Mobile Assistant for Exhibition Tours," *Social Interaction and Communityware*, Kyoto, Japan (June 1998).
21. S. Long, D. Aust, G. Abowd, and C. Atkeson, "Cyberguide: Prototyping Context-Aware Mobile Applications," *Proceedings of the Conference on Human Factors in Computing Systems CHI '96*, ACM, New York (1996).
22. B. Kreller, D. Carrega, J. Shankar, P. Salmon, S. Bottger, and T. Kassing, "A Mobile-Aware City Guide Application," *ACTS Mobile Communication Summit*, Rhodos, Greece (1998).
23. J. Rekimoto, Y. Ayatsuka, and K. Hayashi, "Augment-able Reality: Situated Communication Through Physical and Digitial Spaces," *Proceedings of the International Symposium on Wearable Computing*, IEEE, New York (1998).

Accepted for publication June 13, 2000.

Chris Schmandt *MIT Media Laboratory, 20 Ames Street, Cambridge, Massachusetts 02139-4307 (electronic mail: geek@media.mit.edu).* Mr. Schmandt received the B.S. and M.S. degrees from MIT, where he has been building speech systems since 1979. He is the director of the Speech Interface Group at the Media Laboratory, a position he has held since the creation of the laboratory. Before that he worked on speech applications research at the Architecture Machine Group, including the "Put That There" and "Phone Slave" projects, as well as projects in digital video typography and gestural input for stereoscopic video displays. His current research focuses on speech interfaces for highly mobile computers.

Natalia Marmasse *MIT Media Laboratory, 20 Ames Street, Cambridge, Massachusetts 02139-4307 (electronic mail: nmarmas@media.mit.edu).* Ms. Marmasse is a research assistant in the MIT Media Lab Speech Interface Group. She received a master's degree in media arts and sciences from MIT in 1999 and is currently a Ph.D. student. Her current interests, as can be seen in the comMotion system, include context-aware applications for mobile and wearable computers. More information may be found at http://www.media.mit.edu/~nmarmas/.

Stefan Marti *MIT Media Laboratory, 20 Ames Street, Cambridge, Massachusetts 02139-4307 (electronic mail: stefanm@media.mit.edu).* Mr. Marti is a doctoral student in the Speech Interface Group at the MIT Media Lab and a Motorola Fellow. In 1993 he received a master's degree in special psychology, philosophy, and computer sciences from the University of Bern, Switzerland. For his second M.S. degree in media arts and sciences, received from MIT in 1999, he developed Active Messenger and Knothole. Mr. Marti is interested in handheld communication devices, messaging, and autonomous free-flying microhelicopters. More information may be found at http://www.media.mit.edu/~stefanm/.

Nitin Sawhney *MIT Media Laboratory, 20 Ames Street, Cambridge, Massachusetts 02139-4307 (electronic mail: nitin@ media.mit.edu).* Mr. Sawhney is a doctoral student in the Speech Interface Group at the MIT Media Laboratory and a British Telecom Fellow. He received his B.E. and M.S. degrees from Georgia Institute of Technology in 1993 and 1996. He completed a second M.S. degree at the Media Lab in 1998, focusing on Nomadic Radio, a platform for wearable audio messaging. Nitin previously worked at Fuji-Xerox Research Labs in Palo-Alto and Mitsubishi Electric Research Labs in Cambridge on mobile audio interfaces and speech modification techniques. His current research interests include computer-mediated awareness and perceptual interfaces for everyday social settings.

Sean Wheeler *MIT Media Laboratory, 20 Ames Street, Cambridge, Massachusetts 02139-4307 (electronic mail: swheeler@media.mit.edu).* Mr. Wheeler is a research assistant in the MIT Media Laboratory Speech Interface Group. He received his B.A. degree in linguistics from Brandeis University in 1991. He is currently an M.S. student at the Media Lab, where his current work extends the Clues message filtering system. His areas of interest include contextual systems and human-computer interfaces for mobile devices.