CONTEXT-AWARE COMMUNICATION

BILL N. SCHILIT, INTEL CORPORATION
DAVID M. HILBERT AND JONATHAN TREVOR, FUJI–XEROX PALO ALTO LABORATORY

ORL/STL active badge p

on	Prob.	Name
Accs	100%	J Martin
DVI Rm.	80%	O Mason
R302	Tue.	D Milway
R321	10:30	B Minors
R435	Away	P Mital
R310	100%	J Porter
c.	Away	B Roberts
F3	Mon.	C Turner
AH	100%	R Want
AJ	90%	M Wilkes
Coffee	100%	I Wilson
Meet. Rm.	11:20	S Wray
R311	100%	K Zielinski

12.00 1st January 19

An emerging theme in pervasive computing is the use of context to facilitate and mediate communication among people. Along with the advantages of ubiquitous communication have come new problems with "staying in touch."

ABSTRACT

This article describes how the changing information about an individual's location, environment, and social situation can be used to initiate and facilitate people's interactions with one another, individually and in groups. Context-aware communication is contrasted with other forms of context-aware computing, and we characterize applications in terms of design decisions along two dimensions: the extent of autonomy in context sensing and the extent of autonomy in communication action. A number of context-aware communication applications from the research literature are presented in five application categories. Finally, a number of issues related to the design of context-aware communication applications are presented.

INTRODUCTION

An emerging theme in pervasive computing is the use of context to facilitate and mediate communication among people. Along with the advantages of ubiquitous communication have come new problems with "staying in touch." Fortunately, the convergence of cellular telephony, palm-sized computers, location information, and other sensor data may well provide a basis for context-aware solutions to some of consumer's pervasive communication problems. This article presents a cross section of research that has applied context-aware concepts to reduce communication barriers. Our objective is not to provide an exhaustive survey, but rather to give a historical perspective, as well as describe some recent advances.

It is probably no coincidence that PARC's Etherphone project in the late 1980s and Olivetti's Active Badge location system in the early 1990s both pursued call routing to a mobile user as a key application [1, 2]. At the time, before cell phones were widespread, the notion of phone calls that could follow people as they moved was compelling. Even though mobile phones have lessened the need for call routing, many of us still look forward to integrating, coordinating, taming, and, in general, making our communication technologies even smarter. The approach begun at Xerox PARC and Olivetti Research was to add context (i.e., location) into that process, and continuing this agenda with more types of context will likely be important for future communication systems.

Along with the early work described above, context-aware communication has roots, in part, in two other fields of computer science research, computer supported cooperattive work (CSCW) and human-computer interaction (HCI), and in particular media space research and awareness systems. As Jonathan Grudin points out [3], early media space researchers recognized the importance of shared context in group communication systems. Indeed, the foundational abstraction "What You See Is What I See" (WYSIWIS) aims to support the peripheral context that makes face-to-face interaction work so well [4]. In recent years research on contextualizing collaborative systems has generated an interest in awareness as an independent research focus. For example, the recent work of Hudson [5], Pedersen [6], and others apply abstract visual, or auditory mappings of people's activities to provide situational and social awareness for others, in part to help them construct communication channels. The influence of CSCW and HCI can be seen in the systems described in this article.

In the next section we present a definition for context-aware communication and contrast it with other forms of context-aware computing. This gives a simple set of dimensions by which we discuss a number of context-aware communication systems. We conclude with some challenges and open issues for further research.

DIMENSIONS OF CONTEXT-AWARE COMMUNICATION

Context-aware computing applications examine and react to a user's changing context in order to help promote and mediate people's interactions with each other and their environment. An early overview paper on context-aware applications from Xerox PARC's Ubiquitous Computing Initiative laid out the dimensions shown in Table 1 [7]. These dimensions encompass many types of context-aware applications, including context-aware software to initiate and facilitate communication. Indeed, one of the applications from that paper, a contextual multi-user white board, is presented later as an example of contextual group communication. Over the last decade, it has become clear that there is a continuum from manual to automatic, instead of discrete categories.

In this article we focus on context-aware com-

 1070-9916/02/$17.00 © 2002 IEEE

munication, which is a subset of context-aware computing as it has been described in the literature. However, much that we associate with context-aware computing does not involve communication. For example, researchers have been exploring how context can be used to manage devices in our environment and to deliver and filter all types of information from restaurant guides to operating instructions for a nearby copier. Neither of these topics is associated with communication in the sense we are considering in this article. Nevertheless, the line between information and communication is not always clear. For example, is the Lovegety toy that facilitates conversation by chirping when a "compatible" person is nearby an information or communication device? This article takes a broad definition of communication and includes these and other awareness systems that aim to facilitate, in addition to mediate, human-human communication.

We define *context-aware communication* as the class of applications that apply knowledge of people's context to reduce communication barriers. We suggest a two-dimensional space for such applications based on a simple distinction between "context acquisition" and "communication actions." Along the "acquisition" dimension, an application might ask people to manually enter their context, such as whether they are in a meeting or at lunch, or it may sense and infer a person's context with varying levels of autonomy and sophistication. Along the "action" dimension communication might be manually controlled. For example, an answering machine that says "Lee has been motionless in a dim place with [low] ambient sound for the last 45 minutes. Continue with call or leave a message [8]" relies on the caller to take manual action. In contrast, applications might act more autonomously, such as automatically routing a voice call to a nearby phone. As discussed later, it is not obvious that application designers should simultaneously try to maximize autonomy in both dimensions since this removes human common sense, a quality that Tom Erickson describes as "(at best) awfully hard to implement."

The two dimensions in Fig. 1 provide one way of categorizing various aspects of context-aware communication. The table is populated with examples from the following section. It should be noted that this categorization is only one of many possible ways to discuss context-aware communication. For example, Nagel *et al.* at Georgia Institute of Technology [9] suggest that stages of communication (initiating, mediating, and terminating) can categorize context-aware communication, which is a different yet useful point of view.

	Manual	Automatic
Information	Seeing a selection list of nearby devices and information regarding nearby places	Collaboration channels (e.g., chat) established based on location and popup messages triggered by context
Command	"Print" routes to the nearest printer	Mobile computers cache files onto nearest server

Table 1. *Context-aware software dimensions (adapted from [7]).*

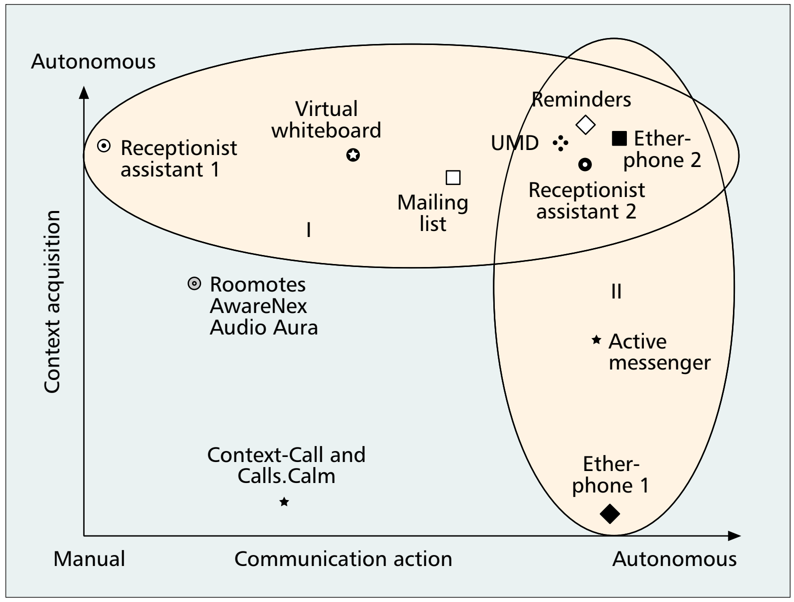


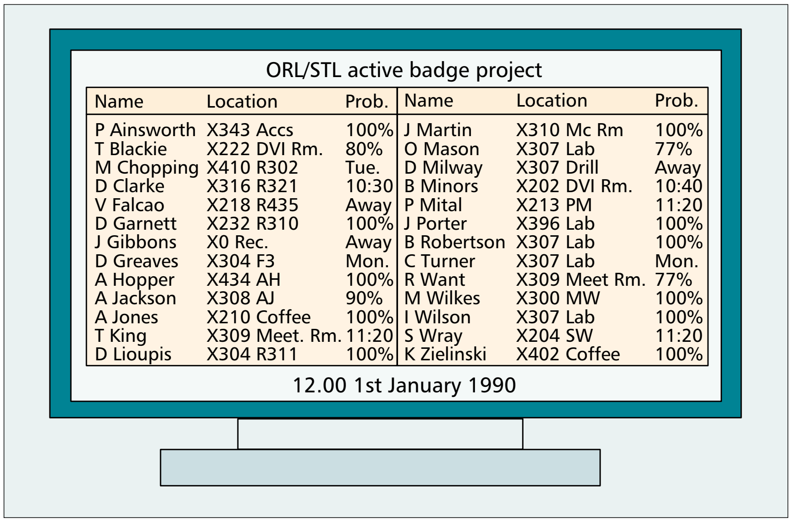
Figure 1. *Context-aware communication dimensions. Context (e.g., location) can be entered manually or sensed automatically, and the communication act (e.g., call routing) can be achieved manually, with user assistance, or autonomously. For example, receptionist assistant 1 automatically detects and displays user location, but requires a person to forward telephone calls. Region I systems tend to automate sensing, and region II systems tend to automate communication acts.*

Context-Aware Communication

In the following section we present a range of context-aware communication systems organized functionally. We include five application types that have been explored by the research community: routing, addressing, messaging, providing caller awareness, and screening. We start with the function of routing a message or call to an appropriate nearby communication device, such as an office telephone. As we describe these applications and systems we explain their position on the scale from manual to autonomous for context acquisition and communication action.

Routing

Location information has long been used as a way to route voice calls. Perhaps the first context-aware communications applications were developed at nearly the same time at Xerox PARC and Olivetti Research Labs (ORL) for routing telephone calls. As shown in Fig. 1 under labels "Etherphone 1" and "Receptionist assistant 1," these systems began at different design points. PARC's Etherphone had the initial strength of autonomous action, being able to automatically route calls, but required manual entry of a person's location. The Olivetti system had the initial strength of automatic person location, but required manual phone routing. Both systems converged on a fully automated approach in their second generation, and provide lessons on the difficulty in adding autonomy. We end this section on routing by describing ubiquitous message delivery, another fully autonomous approach employing intelligent soft-



ORL/STL active badge project

Name	Location	Prob.	Name	Location	Prob.
P Ainsworth	X343 Accs	100%	J Martin	X310 Mc Rm	100%
T Blackie	X222 DVI Rm.	80%	O Mason	X307 Lab	77%
M Chopping	X410 R302	Tue.	D Milway	X307 Drill	Away
D Clarke	X316 R321	10:30	B Minors	X202 DVI Rm.	10:40
V Falcao	X218 R435	Away	P Mital	X213 PM	11:20
D Garnett	X232 R310	100%	J Porter	X396 Lab	100%
J Gibbons	X0 Rec.	Away	B Robertson	X307 Lab	100%
D Greaves	X304 F3	Mon.	C Turner	X307 Lab	Mon.
A Hopper	X434 AH	100%	R Want	X309 Meet Rm.	77%
A Jackson	X308 AJ	90%	M Wilkes	X300 MW	100%
A Jones	X210 Coffee	100%	I Wilson	X307 Lab	100%
T King	X309 Meet. Rm.	11:20	S Wray	X204 SW	11:20
D Lioupis	X304 R311	100%	K Zielinski	X402 Coffee	100%

12.00 1st January 1990

■ Figure 2. *The Olivetti Active Badge displayed the locations of people in the laboratory and was used as an aid for a telephone receptionist to forward calls from the main switchboard [2].*

ware agents.

Following Callers on PARC's Etherphone System

In the 1980s researchers at Xerox PARC developed the Etherphone system that used an Ethernet office network, desktop computers, and office phones to provide enhanced functions for transmitting, storing, and manipulating digital voice [1]. Around 50 Etherphones were deployed in offices at PARC, an environment where people tended to move from office to office for impromptu meetings. When a person visited a colleague, they could register as a "visitor" using the desktop computer interface, and phone calls for them would ring at their own office as well as the visitied location. Alternatively, if an Etherphone user logged into any Etherphone equipped desktop computer, the system would automatically register their new visitor location. An unusual aspect of the Etherphone system was that each user was assigned a distinctive ring tone, or motif, such as the first few bars of "Mary Had a Little Lamb," so no matter where a call occurred, people were able to recognize their calls before answering. This was particularly useful for the call routing function because it meant that visitors could answer their own calls and avoid any confusion to the calling party. In terms of our dimensions in Fig. 1, the early Etherphone system (Etherphone 1) provided autonomous phone routing (action) but tended toward manual location sensing, since visitors had to manually enter themselves into the system.

Toward the latter part of the project, an Olivetti Active Badge system (see below) was installed at PARC and provided automatic location information for the Etherphones. This later system (Etherphone 2 in Fig. 1) combined autonomy in both sensing and action dimensions, reducing work for users, but also making the system more brittle when location sensing didn't work quite right. Swinehart tells a story of walking down an active badge enhanced hallway and hearing his ring motif follow him in the offices along the corridor [10]. Automating Etherphone location sensing had another consequence: the case when call routing was not desired became exceptions, requiring user action, rather than the default, requiring none.

Olivetti's Active Badge Aiding a Telephone Receptionist

Olivetti Research Lab [2] designed and built a novel system for locating people within an office utilizing infrared emitting "active badges" and a network of infrared receivers installed in offices, common areas, and major corridors. The original software application, an "aid for a telephone receptionist," produced a table of names alongside a constantly updating display of locations and telephone extensions. The display is shown in Fig. 2. The column marked Prob. indicated a probability that the badge wearer was still at the sighted location based on the number of sightings and the time of the last sighting.

In contrast to PARC's use of Active Badges for automated phone routing, the purpose of this application was to provide a human receptionist with information useful for tracking down and manually routing incoming telephone calls. Even if people were not recently sighted by the system, the receptionist could call their last sighted location to talk with colleagues in the area and find out if they knew their whereabouts. Whereas PARC started with automation in telecommunications routing, Olivetti began with automation in location sensing (see "Receptionist assistant 1" in Fig. 1). Manual phone routing had the advantage of human intelligence to track down people missing from the badge network, something that would be difficult to build into software systems.

Later on, a proof of concept interface was built to allow certain types of office phone systems to automatically route calls. Olivetti's second system ("Receptionist assistant 2") is similar, in our dimensions, to the second Etherphone system. Although the original ORL system did not take context other than location into account, badge wearers expressed a desire for finer control. For example, people wanted to control call forwarding based on who they are with, where they are, and what time of day it is. Personal control scripts were introduced to address this need.

It is interesting to note that as automation increased, the "intelligence" of this system decreased, since there was no longer an operator using human judgment to track down people. Also, as the system became more autonomous, users wanted more control, but this came at the expense of more work for users up front in specifying rules and exceptions for call routing.

Ubiquitous Message Delivery

Another early example of message routing is the Ubiquitous Message Delivery (UMD) application prototyped at Xerox PARC [11]. A main contribution of this work was a system architecture that provided a level of security in the face of compromised servers. Text messages sent to a user through UMD were delivered "at the soonest *acceptable* time via the most *appropriate* terminal near the recipient." The system employed active badges, keyboard input activity, and explicit commands as a means of detecting user location. The architecture consisted of user and terminal agents. User location and the user's policy regarding message delivery were maintained by a user agent process. Similarly, since desktop terminals have owners, there are also terminal agents to manage the policy of outputting messages on terminals.

Anyone wishing to send a message invoked SendMsg to submit the message to the person's user agent. The user agent maintains information about which public and private terminals the user is currently accessing, as well as what people are near the user's location. The user agent can then check if the user's current situation allows delivery of the message. So, for example, the user may specify that no low priority messages should be delivered to public terminals, or when the user is in the presence of other people. Terminal properties are exported by terminal agents so that user agents can make reasonable choices, such as delivering a message to a handheld device rather than a desktop display when other people are present. When the user's context is suitable for delivery, and a suitable terminal agent exists, the user agent will send the message to the terminal agent for display; otherwise, it will wait until a suitable context or terminal agent exists.

The UMD system is primarily an autonomous system for both sensing and communication acts. The architecture describes a user agent that can encapsulate arbitrarily sophisticated computations for deciding "acceptable" times and "appropriate" terminals. It is likely that for real-world use this system would require a set of very intelligent heuristics. We note, however, that the UMD design did not address the issue of a user-oriented mechanism for specifying these heuristics. Early context-aware systems designers did not tend to focus on the difficult problem of how autonomous behavior might be achieved in ways both reliable and comprehensible to users.

ADDRESSING

A second function of context-aware communication applications is addressing or determining which people should be included in a communication based on context. Whereas the previous section on routing concerned contextually appropriate physical endpoints, addressing is about contextually appropriate people. Below we describe two types of addressing: mailing lists and shared spaces. These applications are situated somewhere between manual and autonomous communication action since people are involved in initiating the communication. It is worth pointing out that communication addressing applications appear "intelligent" but don't seem to need as much work by users in tuning heuristics as does communication routing.

CONTEXT-AWARE MAILING LIST

In their 1993 paper on uses of location in ubiquitous computing [11], Spreitzer and Theimer propose the "note distribution" application to "send a message to all people at a given location." As far as we know, the context-addressed message was designed but never implemented by the authors. However, more recently Dey and others at the Georgia Institute of Technology developed a more practical version called "Context-Aware Mailing List [12]." The Context-Aware Mailing List is a dynamic distribution list that can be used to deliver email messages to members of a research group who are currently in the building. The mailing list might be used, for example, to send a "let's get lunch" message without annoying colleagues not in the building. Dey's dynamically changing mailing list is a new way of addressing email-style communication to a group of people who are selected by their location.

PARCTAB VIRTUAL WHITEBOARD

Synchronous group communication involves a shared channel or "space," such as a chat room, where communication takes place. When membership in the space is determined by the context of the participants, a type of *context-aware chat room* is created. The PARCTAB multi-user Whiteboard[1] supported real-time communication among a group of people selected by their common location and project membership [7, 13]. The motivation for this application comes from the way people use real-world objects: when a group of people are together in a room, they can easily communicate over the physical objects in that place (e.g., a table with scattered papers or a whiteboard with diagrams). To promote similar communication in the virtual world, this mobile palmtop computer application provides a freeform ink workspace for each room. All users in the same room see an icon that selects the room's whiteboard. In addition, if multiple people from a project are collocated, a project-oriented whiteboard can also be selected. Moving to a different room or into a group of different people brings up different drawing surfaces. The contextual addressing of participants makes the PARCTAB Whiteboard more powerful than the physical analogs since the virtual whiteboard can persist from meeting to meeting, and follow participants from room to room.

MESSAGING

Providing the right information at the right time is an often-stated goal of context-aware computing applications. Researchers have looked at extending this notion to getting the right message from another person at the right time. Two types of these systems are described below.

CONTEXTUAL REMINDER MESSAGES IN COMMOTION AND CYBREMINDER

Contextual reminder applications that pop up a message based on the receiver's situation have been explored in a number of research projects. The comMotion [14] and CybreMinder [15] projects recognized that explicitly supporting *another* user to set reminders creates an unusual type of communication. Both systems allow users to associate to-do items with locations in the real world. When the user is in the specified location,

Synchronous group communication involves a shared channel or "space," such as a chat room, where communication takes place. When membership in the space is determined by the context of the participants, a type of context-aware chat room is created.

[1] *Also called "virtual paper" and "tab draw" in other papers.*

Sharing an awareness of one's environment and the context of friends, family, and colleagues can help determine if another person is available for a communication, and can also create serendipitous opportunities for communication.

an audible cue is played and the user can inspect the relevant text or audio item. The CybreMinder system goes beyond comMotion by allowing more elaborate specifications of context including: "forecast is for rain and Bob is leaving home"; "Bob and Sally are together"; and "Bob is alone and has free time and the stock price of IBM is over $100."

In our categorization, the two contextual reminder/messaging applications described above tend to automate both context acquisition and communication action. CybreMinder allows describing many sophisticated situations, and although there is significant overhead for specifying situations, the application is unusual in that the caller rather than the callee does the work.

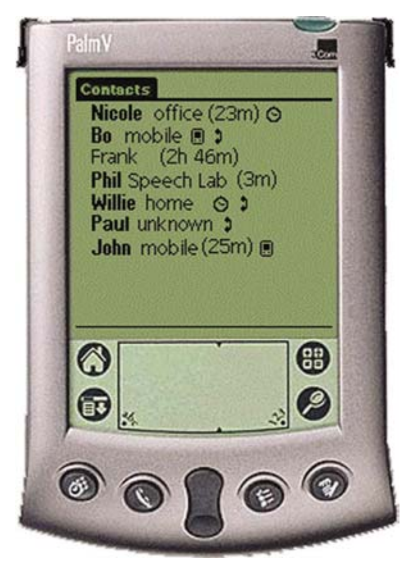
MIT's Active Messenger

The MIT Active Messenger (AM) monitors a user's incoming email messages, prioritizes them, and forwards them to phones, pagers, fax machines, and other communication channels near users [16]. If a message is not received over one channel (e.g., a pager is out of range), it is resent, after a suitable delay, on another channel. The AM determines which channels should be used based on a user preference file along with the user's current location. Although no devices directly report location, the location information is sensed indirectly through caller ID when people phone in to retrieve voice mail or are logged into a workstation. One unusual aspect of the system, and the reason it is included in this section, is that AM doesn't just route messages but also prioritizes the mail messages based on calendar, contact, and location information, and decides how to give you a relevant message at the right time. For example, an email message sent from a person who works in the San Francisco Bay Area may be deemed "timely" and forwarded to a user's pager because a calendar entry includes a phone number within the "408" area code or the user has reviewed voice mail from a phone in the Bay Area.

The AM system provides contextual messages in a compelling way because it takes advantage of the relationships of personal information stored in calendars, contact lists, and mail, and links that to a simple technique for learning a user's location (caller ID). In our categorization active messaging is mostly autonomous in its action, and somewhere between manual and autonomous in context acquisition.

Providing Awareness

Sharing an awareness of one's environment and the context of friends, family, and colleagues can help determine if another person is available for a communication, and can also create serendipitous opportunities for communication. For example, Instant Messenger uses a status (Online, Away, On the Phone, etc.) that gives an indication of a person's availability and willingness to chat. An example of serendipitous interactions occurred in the use of Watchdog [7], an application that allowed people to play an audio clip (e.g., a rooster crowing) when a sensor at the office coffee pot signaled fresh coffee. This contextual reminder had the side-effect of synchronizing people's visits to the kitchen and fostering unplanned discussions and meetings. Users of Active Badge systems are also familiar with the practice of joining a group (meeting) of colleagues they saw reported in the same room. In the next section we describe three systems designed with awareness in mind.



■ Figure 3. *AwareNex shows colleagues and their activities. For example, Nicole is in her office, idle for the last 23 minutes on her workstation, and in the middle of a meeting scheduled in her calendar [17].*

Awareness with AwareNex

The AwareNex system from Sun Laboratories [17] is a portable awareness and communication tool for wireless PDAs, RIM pagers, and cell phones. As shown in Fig. 3, a contact list in AwareNex displays a list of colleagues and information about their locale and activities. This list might show that Cathy was recently in her office, has been idle from her workstation for 23 minutes, has a scheduled appointment in her calendar, and is on her phone. AwareNex goes beyond mobile instant messaging applications since it connects into calendar information, desktop activity, and a telephone switch that allows it to place calls as well as report if a user is on their phone. In terms of the dimensions, AwareNex does a good job of autonomously acquiring context, in part because AwareNex mediates telephone calls so that it can keep track of when users are on their cell or office phones. Similar to other awareness tools, the system expects users to manually make contact (the communication action), although this is facilitated by the contact list that not only indicates people's current activities, but also how to reach them.

Audio Aura

Audio Aura [18] is less about establishing communication and more about augmenting traditional communication mechanisms. The system uses active badges, wireless headphones, and various data sources (e.g., online calendars and email) to create auditory cues as people walk around a workplace. For instance, a person stopping by a colleague's office and finding it empty will hear an auditory cue indicating whether the colleague has been in today and how long they have been away. The system can also produce a "group pulse," indicating when a group of coworkers is currently editing shared group resources, or whether some coworkers are together for a meeting. In terms of dimensions, Audio Aura provides little in the way of automated communication action, but does use automated sensing and casually presented information so people can track down (or just be aware of) their colleague's activities.

Triggering Real-World Meetings with Roomotes

Roomotes gives people remote control of their physical surroundings through Web phones [19] (Fig. 4). The system manages virtual rooms that mirror physical rooms, and presents not only the devices in a room, but also the people. Roomotes allows users to control the lighting and audio-video equipment in a conference room from any Web phone. An unusual aspect of Roomotes is notification: users can request that text messages be sent to their phone whenever the people or status of the devices in a room changes. These text messages sent by the system produce an awareness that can bring people together. For example, Bill and David are in the conference room where they are using their Web phone to control the projector and screen. This has the effect of marking their presence in the conference room. While Bill and David continue to set up for their meeting, Jonathan gets an alert on his Web phone because he asked Roomotes to notify him whenever presentation equipment is being used in the conference room. By accepting the message, Jonathan's phone jumps to the room's page where he sees who is present in the room, and can select a person and then dial through to their cell phone to let them know he is on his way. In terms of our dimensions this system provides context acquisition as a side-effect of remote controlling devices in an environment. The communication is facilitated by letting people connect to others with a single click on their Web phone.

Figure 4. *Roomotes is a Web-phone-based universal remote control that notifies friends and colleagues of interactions with the physical world. When a person enters a virtual room or a physical device in the room changes state, the system sends alerts to other people's cell phones. This creates an awareness that brings people together in physical spaces [19].*

Screening

Call screening concerns establishing communication under appropriate conditions. This class of application uses context to better inform both callers and callees. When initiating conversations in person we usually pay attention to people's situation: who they are with, what they are doing, and where they are. Such context helps to be polite, but also helps to have a productive conversation. For example, it is unlikely you will get an answer to a personal question during a business meeting, or an answer to a business question during a family dinner. The Context-Call and Calls.Calm projects described below share the similar aim of providing callers with context information so that they can make reasonable decisions about initiating conversations.

Context-Call and Calls.Calm Inform Caller of Context

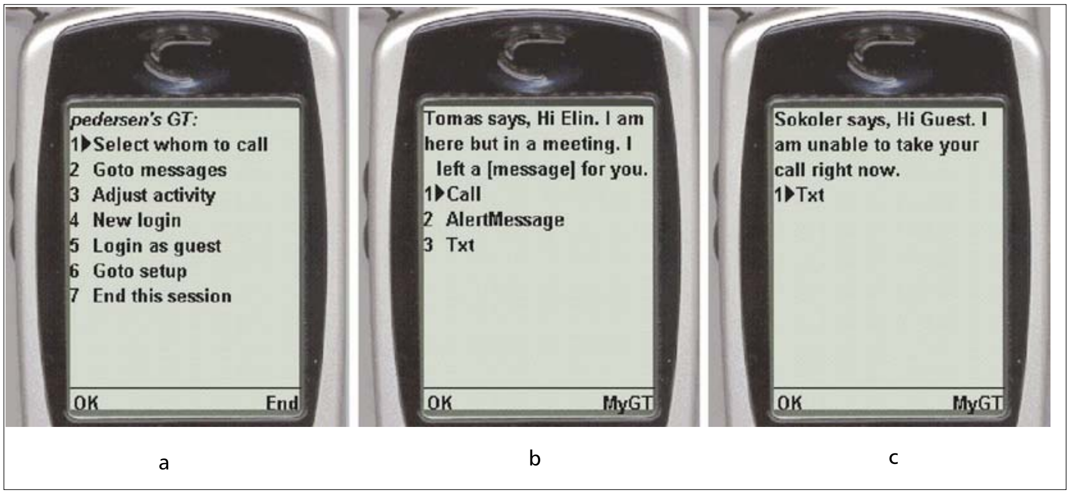
People use many communication devices to facilitate communication, but this can often lead to the phenomenon known as "phone tag" where callers go back and forth communicating with each other's devices but not actually with each other. Both Context-Call [20] and Calls.calm [21] provide callers information about the callee's situation and then rely on callers to make reasonable choices regarding the time and mechanisms for communication. In Calls.calm, the callee specifies the extent of a caller's access to situation information and communication channels in a database of relationships. The specifications from the relationship database together with current data about the callee's situation are combined to create a custom interaction page that presents context information, communication options, and also short messages. Figure 5 shows an example interaction page. One advantage of Calls.calm is that it allows smoother transition between synchronous and asynchronous communication, and allows people to coordinate on a suitable time to communicate synchronously. In both these systems, the context is entered manually by callees, and the communication action is initiated by callers after receiving contextual feedback.

Conclusions

Building communication applications that have an awareness of people's context can help reduce barriers that routinely complicate communication between individuals and groups. This article provided a sample of research that applies the notion of context to route, address, message, provide awareness, and screen our communications. We have presented a categorization of these systems and characterized them in terms of their level of autonomy with respect to context acquisition and communication action.

In viewing the systems from the manual-autonomous perspective, we noticed some tensions between the goal of autonomy and other system goals. On one hand, increasing autonomy of context acquisition is desirable since it reduces the need for communication *recipients* to specify their context. Increasing autonomy of communication actions is also desirable since it reduces

Automating context acquisition remains a difficult problem because there is a considerable gap between what can be sensed and what is "actually going on" in social interactions and people's minds.



■ **Figure 5.** *Calls.Calm uses Web phones to mediate communication with subscribers. A person (a) selects who to call and (b) is greeted by the callees contact page* contextualized *and* customized for the caller*; or, if the caller is unknown, (c) a generic page. For trusted callers, Calls.Calm reveals status, messages, and a list of preferred communication channels. The system supports negotiating a time to make a voice call by an exchange of short text messages ([21]).*

the need for third-party *operators* or communication *initiators* to perform actions. On the other hand, these benefits do not come without a cost. As context acquisition becomes more autonomous, recipients may feel their privacy is eroding because systems will become more and more aware of their day-to-day activities. Thus, reducing the work required from recipients may also reduce their sense of privacy. In addition, as communication actions become more autonomous, recipients and initiators alike will notice a reduction in "common sense" as humans are removed from the loop. Historically, reducing work for third-party operators and initiators has shifted the burden back to recipients who must manage complicated rules and exceptions to model and deal with real-world complexity.

The systems in this area, while focusing on a wide range of problems, suggest a general set of design objectives that designers of context-aware communications system should consider:

Improving relevance. Deciding when a communication is relevant to the person's current (or near future) situation; for example, getting notification about an email from your travel agent regarding itinerary changes while packing to leave for the airport.

Minimizing disruption. Deciding when and how to notify people that they have a communication. For example, your phone should vibrate and not ring when you are at the symphony (unless it is truly urgent).

Improving awareness. Deciding which information and mechanisms can help people make intelligent communication decisions. For example, the caller should be told you are at the movies before the call goes through.

Reducing overload. Deciding how to reduce the number of communications that don't apply given your context; for example, filtering out emails about going to lunch when you are away from the office (or already at lunch).

Selecting channels. Deciding which communication device should be used to get in touch with somebody; for example, routing calls to your home phone instead of your cell phone when you are at home and cellular reception is poor.

There are many challenges in accomplishing these objectives in terms of both automating context acquisition and automating communication actions. The research community must continue to identify which context is useful for meeting the objectives and how sensors can reliably provide this information. However, automating context acquisition remains a difficult problem because there is a considerable gap between what can be sensed and what is "actually going on" in social interactions and people's minds. For instance, is somebody quiet because they are deep in thought, about to make a mathematical breakthrough, or are they daydreaming, waiting for a call for lunch? For automating communication actions, we need to understand how the burden of work can be placed appropriately (e.g., on the caller is some cases, on the callee in others) so that the cost doesn't exceed the benefit for all parties.

As researchers we should question whether systems that push on autonomy in both acquisition and action are fundamentally brittle when faced with the real world. Is it possible to avoid this brittleness by using machine learning and statistical techniques that let people incrementally improve communication preferences? This may lessen the upfront cost for users, and also adapt to users' changing patterns of behavior. An alternative approach to avoid brittleness is to balance autonomous with manual mechanisms. The manually assisted sensing in AM (it uses Caller ID to obtain a person's location) provides such a balance because people accessing voice mail are logically, at the same time, also interested in other communication information.

Context-aware communication has made considerable progress in reducing barriers to communication, but there are still significant challenges to be overcome. In the future, the goal is a context-aware communication system

Context-Aware Communication Products

■ Figure 6. *The Nokia 6150 gives a user's profile settings (e.g., general, meeting, outdoor, car) that can be used in conjunction with caller groups to provide sophisticated notification.*

Nokia 6100 Cell Phone

The profile function on the Nokia 6100 phones lets users adjust ring tones according to situations and caller groups (Fig. 6). Up to seven profile settings are designed to suit the different roles in people's lives: General (Default), Meeting, Outdoor Pager, Silent, Car Kit, and Headset. The profile settings when combined with priority grouping give users sophisticated control of which calls they choose to receive. However, since there are no sensors, the user must manually set active profiles, although researchers have demonstrated how to connect a sensor unit to this phone [1].

Parent Pager Child Security System

The Parent Pager™ sounds an alert when children are about to wander out of earshot of their parents (Fig. 7). The adult's unit can be slipped into a pocket or clipped to a belt; the child's unit is housed in a pouch on an adjustable belt, and can be worn in back to prevent tampering. The adult's unit has a specially designed short-range setting that will signal an alert if the child wanders out of a 10–15-ft radius. The unique sound on the adult's unit will not be confused with other regular beepers and can easily be heard in noisy areas. The second longer range on the adult's unit is up to 50 ft. In addition, the Parent Pager incorporates a pool alarm. The adult's unit immediately sounds if the child's unit becomes submerged.

■ Figure 7. *The Parent Pager is a child safety product that activates an alarm when a child wanders more than 15 ft from the adult unit.*

Garmin Rino Peer-to-Peer Positioning System

The Garmin Rino (Radios Integrated with Navigation for the Outdoors) is a GPS receiver combined with an FRS (Family Radio Service) communicator (Fig. 8). The Rino has the ability to beam your location to other Rino users within a 2-mi range using the FRS spectrum. Other users can then see your location on a detailed map and navigate to your location. The device can store and display downloaded topographic, bathymetric, and street-level map information.

Lovegety

Lovegety devices help strangers meet and start conversations by alerting their owners that someone of the opposite sex, also carrying a Lovegety, is nearby (Fig. 9). When a male Lovegety and a female Lovegety device come within 4.5 m (15 ft) of each other, the device's lights flash and a beeper goes off, alerting the owners of a possible rendezvous. Owners set their mode, whether they are interested in talking, ready to sing karaoke, or ready to do anything (the Get2 mode). Upon encountering another Lovegety, the mode of the potential partner is displayed.

Friend Finder

In November 2001, the Swedish phone company Telia and SignalSoft, Inc., introduced a "friend finder" mobile phone service. FriendFinder uses automatic location via the cellular phone network to bring "youthful people with active social lives" together. People subscribing to the service set up group lists, similar to buddy lists, which includes their friends, coworkers, or other lists. Then the subscriber sends an SMS message to search for friends on their list. A return message informs the subscriber of the location of the people on the list, and then subscribers can communicate individually or collectively with their friends.

References

[1] A. Schmidt *et al.*, "Advanced Interaction in Context," *Proc. 1st Int'l. Symp. Handheld and Ubiquitous Computing (HUC'99)*, vol. 1707 of Lecture Notes in Computer Science, Springer-Verlag, pp. 89–101.

■ Figure 9. *Lovegety. A Japanese toy for meeting people, it beeps when a compatible partner is nearby.*

■ Figure 8. *Garmin's Rino GPS Radio gives users the ability to beam their location to other Rino users within a two-mile range and carry on conversations at the same time.*

In the future, the goal is a context-aware communication system that refuses to ring your phone at the opera, unless it's the babysitter calling to say your kids just set the house on fire.

that refuses to ring your phone at the opera unless it's the babysitter calling to say your kids just set the house on fire.

References

[1] D. Swinehart, "Telephone Management in the Etherphone System," *Proc. IEEE/IEICE Global Telecommun. Conf.*, Tokyo, Japan, Nov. 1987, pp. 1176–80.
[2] R. Want *et al.*,"The Active Badge Location System," *ACM Trans. Info. Sys.*, vol. 10, no. 1, Jan. 1992, pp. 91–102.
[3] J. Grudin, "Desituating Action: Digital Representation of Context," Special issue on *Context-Aware Computing, Human-Comp. Interaction J.*, vol. 16, 2001.
[4] M. Stefik *et al.*, "WYSIWIS Revised: Early Experiences with Multiuser Interfaces," *ACM Trans. Office Info. Sys.*, vol. 5, no. 2, Apr. 1987, pp. 147–67.
[5] S. E. Hudson and I. Smith. "Techniques for Addressing Fundamental Privacy and Disruption Tradeoffs in Awareness Support Systems," *Proc. Comp. Supported Cooperative Work*, 1996, pp. 248–57.
[6] E. R. Pedersen and T. Sokoler. "AROMA: Abstract Representation of Presence Supporting Mutual Awareness," *Proc. SIGCHI Conf. Human Factors in Comp. Sys.* , Atlanta, GA, Mar. 22–27, 1997, pp. 51–58.
[7] B. N. Schilit, N. Adams, and R. Want. "Context Aware Computing Applications," *Proc. Wksp. Mobile Comp. Sys. and Apps.*, Santa Cruz, CA, Dec. 1994, pp. 85–90.
[8] T. Erickson, "Some Problems with the Notion of Context-Aware Computing," *Commun. ACM*, vol. 45, no. 2. Feb. 2002, pp. 102–4.
[9] K. Nagel *et al.*, "The Family Intercom: Developing a Context-Aware Audio Communication System," *Proc. Int'l Conf. Ubiq. Comp.*, Atlanta, GA, Sept. 2001.
[10] D. Swinehart, Personal communication, Feb. 28, 2002.
[11] M. Spreitzer and M. Theimer, "Providing Location Information in a Ubiquitous Computing Environment," *Proc. 14th ACM Symp. Op. Sys. Principles*, 1993, pp. 270–83; also in *Mobile Computing*, T. Imielinski and H. Korth, Eds. Dordrecht, The Netherlands: Kluwer, 1996, pp. 397–423.
[12] A. K. Dey, G. D. Abowd, and D. Salber. "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications," Special issue on Context-Aware Computing, *Human-Comp. Interaction J.*, vol. 16, 2001, pp. 97–166.
[13] B. N. Schilit. "A Context-Aware System Architecture for Mobile Distributed Computing," Ph.D. thesis, Columbia Univ., May 1995.
[14] N. Marmasse and C. Schmandt. "Location-aware Information Delivery with ComMotion," *Proc. 2nd Int'l. Symp. Handheld and Ubiq. Comp.*, Bristol, U.K., Sept. 2000. Springer Verlag, pp. 157–71.
[15] A. K. Dey and G. D. Abowd. "CybreMinder: A Context-Aware System for Supporting Reminders," *Proc. 2nd Int'l. Symp. Handheld and Ubiq. Comp.*, Bristol, U.K., Sept. 2000, pp. 172–86.
[16] C. Schmandt *et al.*, "Everywhere Messaging," *IBM Sys. J. 2000*, vol. 39, nos. 3 and 4, pp. 660–77.
[17] J. Tang, N. Yankelovich *et al.*, "ConNexus to Awarenex: Extending Awareness to Mobile Users," *Proc. SIGCHI Conf. Human Factors in Comp. Sys.*, 2001, Seattle, WA, Mar. 31–Apr. 5, 2001, pp. 221–28.
[18] E. D. Mynatt *et al.*, "Designing Audio Aura," *Proc. SIGCHI Conf. Human Factors in Comp. Sys.*, Apr. 1998, pp. 566–73.
[19] R. Wakikawa *et al.*, "Roomotes: Ubiquitous Room-based Remote Control from Cell Phones," *Extended Abstracts of the SIGCHI Conf. Human Factors in Comp. Sys.*, Seattle, WA, Mar. 31–Apr. 5, 2001, pp. 239–40.
[20] A. Schmidt, A. Takaluoma and J. Mäntyjärvi, "Context-Aware Telephony over WAP," *Personal Technologies*, vol. 4, no. 4, Sept. 2000, pp. 225–29.
[21] E. R. Pedersen, "Calls.calm: Enabling Caller and Callee to Collaborate," *Proc. SIGCHI Conf. Human Factors in Comp. Sys.*, Apr. 1–4, 2001, Seattle, WA, pp. 235–36.

Additional Reading

[1] R. Want *et al.*, "The PARCTAB Ubiquitous Computing Experiment," *Mobile Computing*, H. F. Korth and T. Imielinski, Eds., Kluwer, 1996.

Biographies

DAVID HILBERT (hilbert@fxpal.com) is a research scientist at FX Palo Alto Laboratory (FXPAL) working in the area of personal and mobile computing. His research interests lie in the overlap between software engineering and human-computer interaction. His Ph.D. work at the University of California (UC) at Irvine explored techniques for capturing application usage data on a large scale to help improve the fit between application design and use. Before joining FXPAL, he also worked as a software engineer at NASA's Jet Propulsion Laboratory and as a program manager at Microsoft. He is a member of the IEEE, ACM, and Phi Beta Kappa.

BILL SCHILIT (bill.schilit@intel.com) is co-director of Intel Research Seattle. HIs research focuses on ubiquitous and proactive computing applications. His work is positioned at the intersection of networking and human-computer interaction. Prior to joining Intel, he managed the Personal and Mobile Computing Group at FX Palo Alto Laboratory, a Fuji Xerox Company. He was also a researcher at AT&T Bell Labs and Xerox Palo Alto Research Center where he championed the notion of location-aware computing and was part of the team that built the PARCTAB, the first context-aware handheld computer. He serves on the editorial boards of *IEEE Computer* and *IEEE Wireless Communications*.

JONATHAN TREVOR (trevor@fxpal.com) is a senior research scientist at FX Palo Alto Laboratory (FXPAL) working in the areas of ubiquitous systems and computer-supported cooperative work. For his Ph.D. at the CSCW Research Center, Lancaster University, he developed an infrastructure for developing cooperative systems. While at GMD, Germany, he co-developed BSCW, a Web-based shared workspace system that gained the European Software Innovation Award in 1996. He returned to Lancaster to work on a number of Europe-wide projects, including eSCAPE, an electronic landscape providing interconnections to other virtual environments. His current research at FXPAL continues within an interdisciplinary team, and focuses on the development of readily accessible groupware and HCI applications across a wide-range of technologies and platforms.