CONTEXT-AWARE TELECOMMUNICATION SERVICES

Bill N. Schilit, David M. Hilbert, and **Jonathan Trevor**
Fuji-Xerox Palo Alto Laboratory, CA, USA

Keywords: Telecommunication Services, Context, Context Awareness, Middleware, Routing, Addressing, Messaging, Screening.

Contents

1. Introduction
2. Dimensions of Context-aware Communication
3. Context-aware Communication
3.1. Routing
3.2. Addressing
3.3. Messaging
3.4. Providing Awareness
3.5. Screening
4. Context-aware Communication Products
5. Conclusions
Bibliography
Biographical Sketches

Summary

This chapter 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.

1. 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 consumers a basis for context-aware solutions to some of their pervasive communication problems. This chapter presents a cross section of research that has applied context-aware concepts to reducing 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 1980's and

Olivetti's Active Badge location system in the early 1990's both pursued call routing to a mobile user as a key application. 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, CSCW and HCI, and in particular media space research and awareness systems. As Jonathan Grudin points out, 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. 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, Pedersen, 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 chapter.

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.

2. 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 Figure 2. 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 chapter we focus on context-aware communication which is a subset of context-aware computing as it has been described in the literature. However, there is much that we associate with context-aware computing that does not involve communication. For example, researchers have been exploring how context can be used to manage devices in our environment and how context can be used 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 chapter. 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 chapter takes a broad definition of "communication" and includes these and other awareness systems that aim to facilitate, in addition to mediating, 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" 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."

	MANUAL	AUTOMATIC
INFORMATION	Seeing a selection list of nearby devices & information regarding nearby places	Collaboration channels (e.g., chat) established based on location & pop-up messages triggered by context
COMMAND	"Print" routes to the nearest printer	Mobile computers cache files onto nearest server

Table 1: Context-Aware Software Dimensions

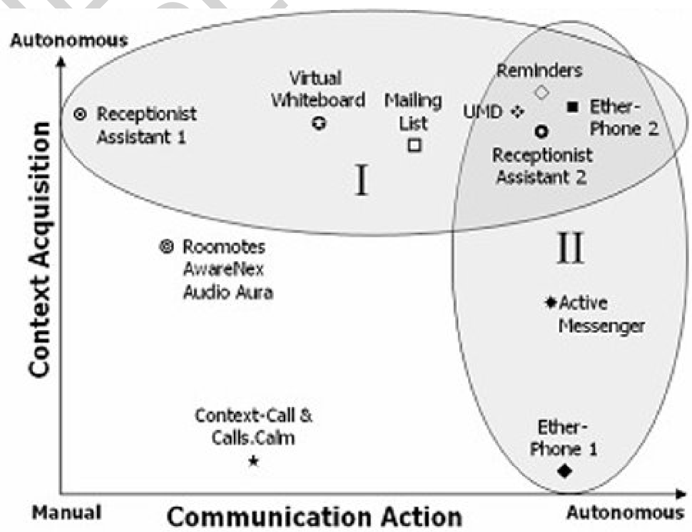


Figure 1: Context-Aware Communication dimensions. Context (e.g. location) can be entered manually or sensed automatically and the communication act (e.g. call routing)

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.

The two dimensions in Figure 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 suggest that stages of communication (initiating, mediating, and terminating) can categorize context-aware communication, which is a different yet useful point of view.

3. 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.

3.1. 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 Figure 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, yet 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 provided lessons on the difficulty in adding autonomy. We end this section on routing by describing Ubiquitous Message Delivery, another fully autonomous approach employing intelligent software agents.

Following Callers on PARC’s Etherphone System

In the 1980’s 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. 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 visiting 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 Figure 1, the early Etherphone system ("Etherphone 1") provided autonomous phone routing (action) but tended towards manual location sensing, as visitors had to manually enter themselves into the system.

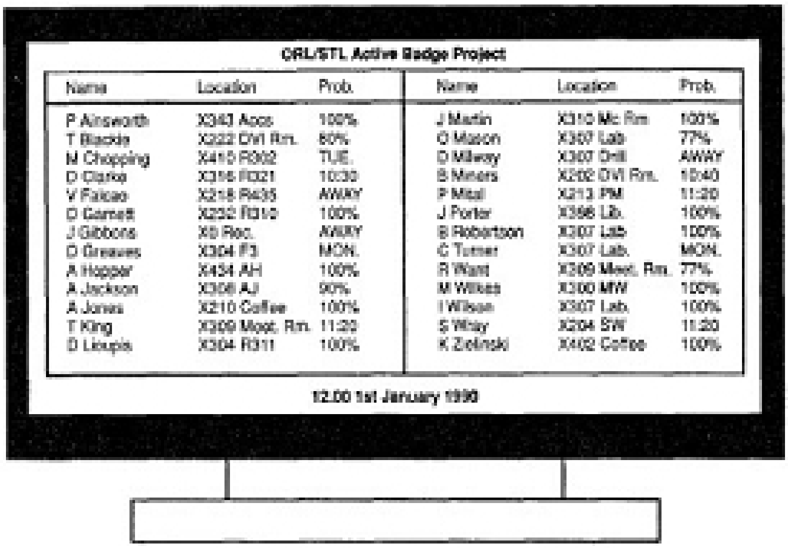
Towards the later part of the project, an Olivetti Active Badge system (see below) was installed at PARC and provided automatic location information for the Etherphones. Swinehart tells a story of walking down an active badge enhanced hallway and hearing his ring motif follow him in the offices along the corridor. This later system ("Etherphone 2") 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. 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 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 Figure 1. The column marked "Prob." indicated a probably 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 Figure 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 the time of day. Personal control scripts were introduced to address this need.



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 Miners	X202 DVI Rm.	10:40
V Falcao	X218 R435	AWAY	P Mital	X213 PM	11:20
D Garnett	X232 R310	100%	J Porter	X398 Lib.	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, (Adapted from [21]).

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. 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 user'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, for example, delivering a message to a hand-held 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.

-
-
-

Bibliography

Dey, A.K. and Abowd, G.D. (2000). CybreMinder: A Context-aware System for Supporting Reminders. *2nd International Symposium on Handheld and Ubiquitous Computing (HUC'00)*, Springer Verlag, pp.172-186, Bristol, UK. [This paper firstly proposes a distributed reminder delivery system based on dynamically applicable user contexts]

Dey, A.K., Abowd, G.D. and Salber, D. (2001). A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *Human-Computer Interaction (HCI) Journal*, Special issue on Context-Aware Computing, Vol. 16, pp. 97-166. [Presentation of a framework to help developers in designing and implementing context-aware applications with a rapid prototype approach]

Erickson, T. (2002). Some Problems with the Notion of Context-Aware Computing. *Communications of the ACM* 45(2), pp. 102-104. [Opinion paper about the concept and different definitions emerging about context awareness and context-dependent services]

Grudin, J. (2001). Desituating Action: Digital Representation of Context. *Human-Computer Interaction (HCI) Journal*, Special issue on Context-Aware Computing, Vol. 16, pp. 97-166. [Proposal of how to dynamically represent applicable contexts, with discussions about interoperability and expressive power issues]

Hudson, S.E. and Smith, I. (1996). Techniques for Addressing Fundamental Privacy and Disruption Tradeoffs in Awareness Support Systems. *Computer Supported Cooperative Work Conference (CSCW)*, ACM Press, pp. 248-257. [Seminal work about privacy issues in middleware supports with context awareness]

Marmasse, N. and Schmandt, C. (2000). Location-aware Information Delivery with ComMotion. *2nd International Symposium on Handheld and Ubiquitous Computing (HUC'00)*, Springer Verlag, pp.157-

171, Bristol, UK. [This paper presents one of the first solutions to support services whose contents depend on the current location of served clients]

Mynatt, E.D., Back, M., Want, R., Baer, M. and Ellis, J.B. (1998). Designing Audio Aura. *SIGCHI conference on Human Factors in Computing Systems*, pp. 566-573. [One of the first prototypes providing context-aware audio services]

Nagel, K., Kidd, C., O'Connell, T., Dey, A. and Abowd, G. (2001). The Family Intercom: Developing a Context-Aware Audio Communication System. *International Conference on Ubiquitous Computing (Ubicomp'01)*, Atlanta, GA. [Solution for home-based audio services with full awareness of context information about family members]

Pedersen, E.R. (2001). Calls.calm: Enabling Caller and Callee to Collaborate. *SIGCHI Conference on Human Factors in Computing Systems*, pp. 235-236, Seattle, USA. [This paper presents a prototype to facilitate interworking and collaboration between end-points of a telephone communication]

Pedersen, E.R. and Sokoler, T. (1997). AROMA: Abstract Representation of Presence Supporting Mutual Awareness. *SIGCHI Conference on Human Factors in Computing Systems*, pp. 51-58, Atlanta, USA. [Seminal work about models and formats for representation of context to enable mutual presence awareness]

Schilit, B.N. (1995). A Context-Aware System Architecture for Mobile Distributed Computing. *PhD Thesis, Columbia University*. [Foundational work, definitely one of the first articulated contribution, about the realization of supports for context awareness in distributed and open environments]

Schilit, B.N., Adams, N. and Want R. (1994). Context Aware Computing Applications. *Workshop on Mobile Computing Systems and Applications (WMCSA)*, pp. 85-90, Santa Cruz, USA. [One of the first work in the literature identifying the issue and the opportunity of designing distributed applications depending on context]

Schmandt, C., Marmasse, N., Marti, S., Sawhney, N. and Wheeler, S. (2000). Everywhere Messaging. *IBM Systems Journal* 39(3&4), pp. 660-677. [How to enhance the provisioning of messaging services with applicable context information]

Schmidt, A., Aidoo, K.A., Takaluoma, A., Tuomela, U., van Laerhoven, K. and van de Velde, W. (1999). Advanced Interaction in Context. *1st International Symposium on Handheld and Ubiquitous Computing (HUC)*, Lecture Notes in Computer Science 1707, Springer-Verlag, pp. 89-101. [This paper focuses on the aspects related to the desired and envisioned consequences of context status in massaging interactions among currently served users]

Schmidt, A., Takaluoma, A. and Mäntyjärvi, J. (2000). Context-Aware Telephony over WAP. *Personal Technologies* 4(4), pp. 225-229. [The originality of the paper is in applying the idea of context dependence to telephony services provided on top of the WAP solution stack]

Spreitzer, M. and Theimer, M. (1996). Providing Location Information in a Ubiquitous Computing Environment. In the *"Mobile Computing"* Book (Imielinski T. and Korth, H., editors), Kluwer Academic Publishers, Dordrecht, The Netherlands, pp. 397–423. [Seminal work about the provisioning of positioning information for clients that move at runtime while accessing ubiquitous services]

Stefik, M., Bobrow, D.G., Foster, G., Lanning, S. and Tatar, D. (1987). WYSIWIS Revised: Early Experiences with Multiuser Interfaces. *ACM Transactions on Office Information Systems* 5(2), pp. 147-167. [Seminal work about the design and exploitation of multi-user and multi-modal interfaces]

Swinehart, D. (1987). Telephone Management in the Etherphone System. *IEEE/IEICE Global Telecommunications Conference*, Tokyo, Japan, pp. 1176-1180. [Seminal work about the usage of integrated management solutions for telephone services provided over distributed systems]

Tang, J., Yankelovich, N., Begole, J., van Kleek, N., Li, F. and Bhalodia, J. (2001). ConNexus to Awarenex: Extending Awareness to Mobile Users. *SIGCHI Conference on Human Factors in Computing Systems*, pp. 221-228, ACM Press, Seattle, USA. [This paper presents motivations and solutions for providing users with full visibility of context awareness in pervasive computing services]

Wakikawa, R., Trevor, J., Schilit, B.N. and Boreczky, J. (2001). Roomotes: Ubiquitous Room-based Remote Control from Cell Phones. *SIGCHI Conference on Human Factors in Computing Systems*, pp.

239-240, ACM Press, Seattle, USA. [The paper presents an architecture and a solution prototype for controlling home appliances via off-the-shelf communication devices such as cell phones]

Want, R., Hopper, A., Falcao, V. and Gibbons, J. (1992). The Active Badge Location System. *ACM Transactions on Information Systems* 10(1), pp. 91–102. [Together with PARCTAB, one of the first relevant prototypes for ubiquitous computing. That experience identified primary issues and generated several design guidelines explored in following research activities]

Want, R., Schilit, B.N., Adams, N.I., Gold, R., Petersen, K., Goldberg, D., Ellis, J.R. and Weiser, M. (1996). The PARCTAB Ubiquitous Computing Experiment. In the "*Mobile Computing*" Book (Korth, H.F. and Imielinski, T., editors), Kluwer Academic Press. [Together with Active Badge, one of the first relevant prototypes for ubiquitous computing. That experience identified primary issues and generated several design guidelines explored in following research activities]

Biographical Sketches

David Hilbert is a senior research scientist at FX Palo Alto Laboratory (FXPAL). His research interests lie in the design and evaluation of novel interactive, collaborative, and ubiquitous computing applications. He received a Ph.D. in Information and Computer Science from the University of California (UC) Irvine. He is a member of IEEE, ACM, and Phi Beta Kappa Society.

Bill Schilit is research staff at Google. His research focuses on ubiquitous and proactive computing applications. Previous employers include Intel, FX Palo Alto Laboratory (FXPAL), Xerox Palo Alto Research Center, and AT&T Bell Labs.

Jonathan Trevor is employed by Yahoo! Inc. He conducted this research while employed as a senior research scientist at FX Palo Alto Laboratory (FXPAL), where he worked in ubiquitous systems, computer-supported cooperative work, and web-based applications. He received a Ph.D. in computer science from the University of Lancaster, U.K. and is a member of IEEE and ACM.