A Survey of Context-Aware Mobile Computing Research

Guanling Chen and David Kotz
Department of Computer Science
Dartmouth College

Dartmouth Computer Science Technical Report TR2000-381

Abstract

Context-aware computing is a mobile computing paradigm in which applications can discover and take advantage of contextual information (such as user location, time of day, nearby people and devices, and user activity). Since it was proposed about a decade ago, many researchers have studied this topic and built several context-aware applications to demonstrate the usefulness of this new technology. Context-aware applications (or the system infrastructure to support them), however, have never been widely available to everyday users. In this survey of research on context-aware systems and applications, we looked in depth at the types of context used and models of context information, at systems that support collecting and disseminating context, and at applications that adapt to the changing context. Through this survey, it is clear that context-aware research is an old but rich area for research. The difficulties and possible solutions we outline serve as guidance for researchers hoping to make context-aware computing a reality.

1. Introduction

Two technologies allow users to move about with computing power and network resources at hand: portable computers and wireless communications. Computers are shrinking, allowing many to be held by hand despite impressive computing capabilities, while the bandwidth of wireless links keep increasing. These changes have increasingly enabled people to access their personal information, corporate data, and public resources "anytime, anywhere".

There are already many wireless handheld computers available, running different operating systems such as Palm OS, Microsoft Pocket PC (Windows CE), and Symbian EPOC. There are typically several ways that these devices can be connected without wires:

Wireless cellular networks. For example, Palm VII automatically connects to the portal www.palm.net whenever the embedded antenna is flipped on. There are also several vendors making cellular modems that snap on to a device's serial port or into an expansion port. Omnisky provides the cellular modem that outfits the Palm V. Qualcomm integrates the Palm device and cellular phone as a new product pdQ™ SmartPhone, while Handspring has a cellular module to make their Palm OS Visor a mobile phone too.

Wireless LAN networks. For example, the Symbol PPT 2700 has an embedded Spectrum24® antenna that supports the IEEE 802.11 airwave standard for wireless communications and the ITU H.323 standard for multimedia communications. There are also various wireless LAN expansion modules supporting IEEE 802.11 and OpenAir™ standards in the form of CompactFlash, PCMCIA and Springboard cards available on market.

Wireless PAN (Personal Area Network) or BAN (Body Area Network). Wireless PAN or BAN allow communication among devices within a personal operating space, with typical characteristics such as short range, low power, low cost, and small networks with 8 to 16 nodes. Bluetooth is one of the promising RF-based standards intended to replace the cable between devices. There are already Bluetooth-enabled devices available, such as the Ericsson R520 mobile phone. The IEEE 802.15 working group is also developing Personal Area Network consensus standards for short distance wireless network (WPAN). Of course, the venerable IrDA standard allows line-of-sight short-range IR communications.

As computers become more portable, people want to access information anytime and anywhere with the personal devices they carry all the time. Traditional

---

Authors' addresses: Dartmouth College, Hanover, NH 03755. Email: First.Last@Dartmouth.EDU. This research has been supported by DARPA contract F30602-98-2-0107, by DoD MURI contract F49620-97-1-03821, by Microsoft Research, and by Cisco Systems University Research Program.

distributed systems that assume a stationary execution environment are no longer suitable for such extremely mobile scenarios. In light of this, many mobile computing researchers have tried to shield the mobility and make frequent disconnection transparent to end-users (e.g., the Coda file system [SKM+93] and Mobile IP [PJ96] work).

We believe that, instead of adapting systems and applications so that mobility is hidden, we should explore and provide infrastructure to support new *mobile-aware applications*. Such mobile-aware applications will be more effective and adaptive to users' information needs without consuming too much of a user's attention, if they can take advantage of the dynamic environmental characteristics, such as the user's location, people nearby, time of the day, and even light and noise levels.

Researchers at Olivetti Research Ltd. (ORL)[1] and Xerox PARC Laboratory pioneered the context-aware computing area [WHF+92, STW93] under the vision of *ubiquitous computing* [Wei91, Wei93], also called *pervasive computing* (the goal is to enhance computer use by making many computers available throughout the physical environment, but making them effectively invisible to the user). Since then, many other researchers have studied this topic and contributed to this area.

The goal of this paper is to survey the most relevant literature in this area. In Section 2 we discuss various definitions of "context". Section 3 summarizes ways in which context is used. We list in Section 4 most context-aware applications that we have seen. Section 5 and 6 are about how to sense and model the contextual information. Section 7 discusses the infrastructure to support context-aware applications. Section 8 concerns security and privacy. Section 9 gives a final summary and our conclusion of this survey.

2. Definition of "Context"

The word "context" is defined as "the interrelated conditions in which something exists or occurs" in Merriam-Webster's Collegiate Dictionary.[2] While this is a general definition, it does not help much for understanding the concept in a computing environment. The use of the word "context" tends to be vague because everything in the world happens in a certain context. The term has been used in many ways in different areas of computer science, such as "context-sensitive help", "contextual search", "multitasking context switch", "psychological contextual perception", and so on. We only focus on the context used by applications in mobile computing.

Not satisfied by a general definition, many researchers have attempted to define context by enumerating examples of contexts. Schilit divides context into three categories [SAW94]:

- *Computing context*, such as network connectivity, communication costs, and communication bandwidth, and nearby resources such as printers, displays, and workstations.
- *User context*, such as the user's profile, location, people nearby, even the current social situation.
- *Physical context*, such as lighting, noise levels, traffic conditions, and temperature.

Time is also an important and natural context for many applications. Since it is hard to fit into any of the above three kinds of context, we propose to add a fourth context category as:

- *Time context*, such as time of a day, week, month, and season of the year.

More importantly, when the computing, user and physical contexts are recorded across a time span, we obtain a *context history*, which could also be useful for certain applications.

Some other researchers try to formally define context. Schmidt et al. define context as "knowledge about the user's and IT device's state, including surroundings, situation, and to a less extent, location" [SAT+99]. Dey defines context as "any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves" [DA99].

Combining several context values may generate a more powerful understanding of the current situation. "Primary" contexts, including location, entity, activity and time, act as indices into other sources of contextual information [DA99]. For example, knowing the current location and current time, together with the user's

[1] Now part of AT&T Laboratories Cambridge.
[2] http://www.m-w.com/

calendar, the application will have a pretty good idea of the user's current social situation, such as having a meeting, sitting in the class, waiting in the airport, and so on.

From the above discussion, we understand that context is a general word and has a loose definition. We believe, however, that context in mobile computing really has two different aspects. One aspect of context includes the characteristics of the surrounding environment that *determine* the behavior of the mobile applications. The other aspect of context is *relevant* to the application, but not critical. It is not necessary for applications to adapt to the second kind of context except to display them to interested users. None of the formal definitions we have read delivers this difference, so we give our definition here:

> *Context is the set of environmental states and settings that either determines an application's behavior or in which an application event occurs and is interesting to the user.*

Further, we define the first kind of context as *active* context that influences the behaviors of an application, and the second kind of context as *passive* context that is relevant but not critical to an application. Given one particular context class, whether it is active or passive depends on how it is used in applications. For example, a user's location information is used actively in the automatic Call Forwarding system [WHF+92] and used passively in the Active Map application [Wei93]. We found that this classification of two types of context helps to understand the use of context in mobile applications as we discuss in the following section.

3. Context-aware Computing

Although information about the current context may be available to mobile applications, how to effectively use that information is still a challenging problem for application programmers. Schilit defines *context-aware computing* by categorizing context-aware applications as follows [SAW94]:

1. *Proximate selection*, a user-interface technique where the objects located nearby are emphasized or otherwise made easier to choose.
2. *Automatic contextual reconfiguration*, a process of adding new components, removing existing components, or altering the connections between components due to context changes.
3. *Contextual information and commands*, which can produce different results according to the context in which they are issued.
4. *Context-triggered actions*, simple IF-THEN rules used to specify how context-aware systems should adapt.

While the above categorization identifies classes of context-aware applications, Pascoe [Pas98] proposes a taxonomy of context-aware features, including contextual sensing, contextual adaptation, contextual resource discovery and contextual augmentation (the ability to associate digital data with a user's context). Dey combines these ideas and maps them to three general categories of context-aware features that context-aware applications may support: *presentation* of information and services to a user, automatic *execution* of a service, and *tagging* of context to information for later retrieval [DA99].

From our definition of context in the previous section, however, we have a different perspective on how a mobile application can take advantage of context. There are essentially two ways to use context: automatically adapt the behaviors according to discovered context (using active context), or present the context to the user on the fly and/or store the context for the user to retrieve later (using passive context). Thus we give two definitions of context-aware computing:

> *Active context awareness: an application automatically adapts to discovered context, by changing the application's behavior.*
>
> *Passive context awareness: an application presents the new or updated context to an interested user or makes the context persistent for the user to retrieve later.*

Active context-aware computing is more interesting because it leads to new applications on mobile devices and it requires more infrastructure support. We believe it may help to eliminate unnecessary user cooperation and make technology as "calm" as possible.

4. Context-Aware Applications

In this section, we survey previous research work about context awareness, focusing on applications, what context they use, and how contextual information is leveraged. Dey makes a similar effort and gives a summary of such applications [DA99]. We want to expand his work and give more details in this survey paper.

Call Forwarding [WHF+92]
Group: Olivetti Research Ltd. (ORL).
Passive context: None.
Active context: User's location.
Description: Based on the Active Badge system[1], the location context is presented to the receptionist, who routinely forwards the telephone calls to the destination user's nearest phone (location context is used passively to display to receptionist). Recently, with the help of a PBX that has the digital interface designed for computer-integrated telephony, the location context becomes active to help automatically forward the phone calls[2]. This application proves to be useful to both the staff and telephone receptionist. It was observed, however, that people want to express more control over when calls are forwarded to them, depending on their current context; for example, they prefer not to take unexpected calls when having a meeting with their boss.

Teleporting [BRH94]
Group: Olivetti Research Ltd. (ORL).
Passive context: None.
Active context: User's location, and location of workstations.
Description: Teleporting (sometimes called "follow-me" computing) is a tool to dynamically map the user interface onto the resources of the surrounding computer and communication facilities. Also based on the Active Badge system, this tool can track the user location so that the application follows the user while they move around. A new version of the Teleporting system developed at AT&T Laboratories uses a new location tracking system called the Bat[3], which is based on both ultrasonic and radio signals [HHS+99]. An ongoing work Composite Device Computing Environment (CDCE) [PSG00] is also aimed to augment resource-poor PDA with surrounding computing resources such as PCs, workstations, TVs, and telephones.

Active Map [Wei93, WSA+95, WSA+96]
Group: Xerox PARC.
Passive context: User's location.
Active context: None.
Description: The location information (room number) is collected directly from the PARC Tab system, because the PARC Tab network uses a separate wireless base station for each room. The individual faces are shown at the locations of people on the map, which is updated every few seconds, allowing people to be located quickly, or to notice a meeting one might want to attend.

Mobisaic Web Browser [VB94]
Group: Voelker and Bershad at University of Washington.
Passive context: None.
Active context: Location and time.
Description: Mobisaic extends standard client browsers to allow authors to reference dynamic contextual information in hypertext links called dynamic URLs, which may contain environment variables. The dynamic URL is interpreted using current values of the environment variables and the appropriate page is returned. Mobisaic also supports active documents, which are web pages embedded with environment variables. Whenever the values of the environment variables change, the Web page gets automatically updated.

Shopping Assistant [AGK94]
Group: AT&T Bell Laboratories.
Passive context: None.
Active context: Customer's location within the store.
Description: The device can guide the shoppers through the store, provide details of items, help locate items, point out items on sale, do a comparative price analysis, and so forth. There is a privacy concern since the store maintains the customer profiles. As a consequence, customers are divided into two classes: *regular customers* who shop anonymously without profiles in store, and *store customers* who signed up with a store and will get additional discounts in exchange for sacrificing their privacy.

Cyberguide [LKA+96, AAH+97]
Group: Future Computing Environments (FCE) at the Georgia Institute of Technology.
Passive context: Tourist's location.
Active context: Tourist's location and time.
Description: The variations of the system, both indoors and outdoors, provide information services to a tourist about her current location; for example, she can find directions, retrieve background information, and leave comments on the interactive map. The travel diary is automatically compiled using the history of where a tourist has traveled over time, and it is used by the

[1] http://www.uk.research.att.com/ab.html
[2] http://www.uk.research.att.com/spirit/
[3] http://www.uk.research.att.com/bat/

system to make suggestions on places of interest to visit. The location information is collected by GPS for the outdoor version, and by an infrared (IR) positioning system, which they developed from TV remote control units, for the indoor version.

Other Guide Systems and Augmented Reality

With similar motivations of Cyberguide project, several other guide systems have been developed to help visitor tour through both physical and information space. Information is disseminated to the handheld devices according to the visitor's current physical location. The GUIDE system [DCM+99] developed at University of Lancaster is a context-sensitive tourist guide for visitors to the city of Lancaster, England. Smaller-scale versions of such guide systems have also been developed for museum visitors [Bed95, OS98] and exhibition tourists [OS00]. The location and orientation contexts are vital and they are mainly used for active context awareness (different content is delivered when the user moves to a new place). A similar category of applications is augmented reality [Fit93], in which the user's view of the real world is augmented with additional virtual information.

Conference Assistant [DFS+99]

Group: Future Computing Environments (FCE) at the Georgia Institute of Technology.
Passive context: current activity (presentation).
Active context: Attendee's location, current time, schedule of presentations.
Description: The assistant uses a variety of context information to help conference attendees [DFS+99]. The assistant examines the conference schedule, topics of presentations, user's location, and user's research interests to suggest the presentations to attend. Whenever the user enters a presentation room, the Conference Assistant automatically displays the name of the presenter, the title of the presentation, and other related information. Available audio and video equipment automatically record the slides of current presentation, comments, and questions for later retrieval.

People and Object Pager [BBC97, Bro98]

Group: University of Kent at Canterbury.
Passive context: None.
Active context: User's location, nearby people and objects.
Description: The pager can send message to a visitor, who wears an active badge but no paging device, by routing the message to the closest person with the visitor. Similarly, when a person needs to locate a book, she can just broadcast this request and whoever encounters this book will be notified to pick it up for the requester.

Fieldwork [Pas98, PMR98]

Group: University of Kent at Canterbury.
Passive context: User's location and current time.
Active context: None.
Description: The system concentrates on providing a set of tools to assist in the fieldworker's observation and data-collection activities, i.e., helping the user record information about their environment. In addition to easing the conventional data collection process by automatically collecting some contextual information, such as location and time, such information is also tagged with the locations on the map for later analysis.

Adaptive GSM phone and PDA [SAT+99]

Group: TEA[1] (Technology for Enabling Awareness) at Starlab.
Passive context: None.
Active context: User's activity, light level, pressure, and proximity of other people.
Description: In the PDA scenario a notepad application is changed to adapt to the font size to the users activity (a large font when the user is walking, small font when stationary) as well as to environmental conditions (e.g., light level). In the phone scenario, the profiles of the mobile phone are selected automatically based on the recognized context. The phone chooses to ring, vibrate, adjust the ring volume, or keep silent, depending on whether the phone is in hand, on a table, in a suitcase, or outside.

Office Assistant [YS00]

Group: MIT Media Laboratory.
Passive context: None.
Active context: Office owner's current activity and schedule.
Description: The assistant is an agent that interacts with visitors at the office door and manages the office owner's schedule. The assistant is activated when a visitor approaches, which is detected by two pressure-sensitive mats placed on both side of the office door, and it will adapt its behavior to such contextual information as the identity of the visitor, the office owner's schedule status and busy status, and the owner's willingness to see the current visitor. It is intrusive, however, to recognize visitor's id by a name-asking process. An alternative approach could be the Active Floor [AJL+97] or Smart Floor [OA00]. Also, if

[1] http://tea.starlab.net/

the visitor's schedule information could be collected, it would be helpful for automatic appointment setting.

Location-aware Information Delivery [MS00]
Group: MIT Media Laboratory.
Passive context: None.
Active context: User's location and current time.
Description: The ComMotion project at MIT takes advantage of both location and time context. Each reminder message is created with a location, and when the intended recipient arrives at that location, the message is delivered via voice synthesis without requiring user to hold the device and read the message on screen. The Rome project at Stanford also demonstrates a similar reminder application [HLP+00]. CybreMinder [DA00] at Georgia Tech augments the reminder tool with more complex context, such as nearby people, and current weather conditions. Researchers at TecO went further, and built a device called MemoClip [Bei00] to support this kind of active reminder application.

Comments
The above list of context-aware applications is not intended to be complete. We focused on the applications that were actually implemented and we want to gather a general sense of what contexts people care about and how they are actually used.

Interestingly, from the above list, we observed that few contexts other than location have been used in actual applications. It is not clear whether other contexts are difficult to sense or whether they are not as useful as we have thought. Also, context history is generally believed to be useful, but it is rarely used. Any reliance on the user to explicitly provide contextual information, such as in Office Assistant projects, proves to be obtrusive and inconvenient for the user. We did not find a "killer app" for context-aware computing; most of the applications use small pieces of contextual information and none of them are especially compelling.

5. Sensing the Context

To be able to use context in applications, of course, there must be a mechanism to sense the current context and deliver it to the application. Here we describe some of the mechanisms that have been used.

5.1 Sensing the Location

Since the location is an important context that changes whenever the user moves, a reliable location-tracking system is critical to many context-aware applications. It is easy to gather such location information if the user is willing to (and always remembers to) supply her location context to the system. Typical techniques include user sliding her badge or pressing a fingerprint reader before entering and leaving (ideally) a room. The system can also watch which workstation the user logged in. These methods, however, need user's explicit cooperation and only provide coarse granularity and low accuracy (if the user forgets to let system know when she leaves the room). In this section we will discuss only automatic location sensing techniques with various granularity.

5.1.1 Outdoors
The obvious choice for an outdoor positioning system is the Global Positioning System (GPS). Recently, the US Government turned off degradation of the civil GPS signal to allow an accuracy of 10 to 20 meters, which is 10 times more accurate than before. Automobile navigation systems instantly benefit from this new policy, and we can certainly imagine many other applications will become possible.

After observing that many small and low-cost devices do not necessary have GPS capability, Bulusu and others propose a connectivity-based localization technique [BHE00]. The localization algorithm produces accuracy of 3.0 meters to the known reference points with an idealized radio model.

5.1.2 Indoors
On the other hand, the GPS signal does not work indoors because the signal strength is too low to penetrate most buildings. Although reflections can sometimes allow a reading inside a building, multipath reflection can make that reading unreliable or cause it to fluctuate. It is a challenging problem to build an ideal indoor location sensor that provides fine-grain spatial information at a high update rate, which is also unobtrusive, cheap, scalable, and robust [HHS+99]. Most related research projects make their own location tracking system. The Olivetti Active Badge system [WHF+92], Xerox ParcTab [WSA+96], and the Cyberguide project [AAH+97] built tracking systems based on infrared (IR). The Personal Shopping Assistant [AGK94] proposed by AT&T and the Pinger near-field tagging system [HNR97] developed at the Hewlett-Packard Laboratories use a radio frequency (RF) transmitter and receiver to track either handheld devices or people. The commercially available 3D-iD system from Pinpoint [WL98] is also based on pure RF with granularity about 10 meters.

Researchers from Olivetti and Oracle Research Laboratory have developed a new tracking system based on both ultrasonic and radio signals, aiming at a location granularity of 15cm [WJH97]. Every 200ms, a controller sends out a radio message to all transceivers on mobile devices, indicating which transceiver should ping in the next time slot. Meanwhile, the control also sends a reset message to the receivers' network, requesting them to be ready to pick up the incoming ultrasonic signal. Both location and orientation can be calculated. The prototype later became the Bat system at AT&T Laboratories Cambridge [HHS+99].

The Cricket location-support system [PCB00] from MIT Laboratory also takes advantage of both ultrasonic and radio signals. Rather than the system tracking the user's location, however, each portable device determines its own location. The mobile device listens to two signals (one RF and one ultrasonic) that simultaneously originate from the base station by measuring the distance to that base station calculated by the time interval between the arrivals of two signals. The resulting location accuracy is a few feet.

The indoor tracking schemes mentioned above are all based on a cellular approach, in which either the mobile device detects its cell or the system determines which mobile devices are in each cell. There are other approaches, however. The RADAR system from Microsoft Research [BP00] uses the RF signal strength in the communications network as an indicator of the distance between a transmitter and a receiver, without setting up an additional location tracking system. The location is determined by querying a central database of RF signal strength at a set of fixed receivers, for known transmitter positions. Similarly, Castro and Muntz measures the signal-to-noise ratio (SNR) between the transmitter and different base stations, while a pre-encoded Bayesian network model describes the joint probability distributed for location given base-station SNRs [CM00].

Other non-cellular approaches include the Active Floor [AJL+97] and Smart Floor [OA00] projects that try to identify persons by their footstep force profiles. Though the accuracy of identifying a moving user is around 90%, it is the most unobtrusive way for users to provide their location information to the system. It works, however, only for people, not for other objects such as mobile devices. Cameras can also be used to track user location [STE99, BMK+00]. Such systems, however, have line of sight problem as IR and work well with only a small number of persons in a room with non-frequent occlusions.

5.1.3 Hybrid

There are some location-tracking systems that are based on network domains, and they are not specifically targeted for indoor or outdoor use. The GUIDE project [DCM+99] takes the approach by setting up a cellular system based on the IEEE 802.11 "wireless Ethernet" standard. The cells are defined by the range of their WaveLAN 802.11 base stations. This unusual approach, using a wireless LAN technology deployed on a metropolitan area, leads to medium-granularity location information and non-overlapping cells, which causes blank spots where visitors will lose track of their location. Their Position Sensor component on a mobile device is able to determine current location via listening for beacons from cell servers since the cells are not overlapped. Such a connectivity-based approach to track mobile users can also be realized using Bluetooth[1].

Another interesting approach to detect location change is to take advantage of the Mobile-IP protocol [CK99]. When the mobile host enters a new zone, it must discover the Foreign Agent (FA) to be assigned a temporary IP address. By installing a context manager service on the same host of FA, the mobile host imports the context of the current zone from the context server just discovered during registration with FA.

5.1.4 Issues

There has been no uniform way to track locations with fine granularity that works both indoors and outdoors. In practice, a system may have to consult different locators to locate different types of objects, while the system can also find location of a particular object via several different types of locators. On the other hand, each sensed data has an associated uncertainty due to environmental noise or sensor errors. Even worse, context sensed from different sensors may conflict to each other.

It is possible, however, to use sensor-fusion techniques to solve such conflicts and improve context accuracy and completeness. For example, a pre-trained Bayesian network is able to represent the joint probability distribution for location, given RF signal-to-noise measurements [CM00]. The location with highest probability gives a certain confidence to the

[1] http://www.lesswire.com/

application about current location even though the underlying measured SNRs are noisy. Rizzo and others also propose a Master Location System (MLS) as a framework to allow multiple location mechanisms to co-exist and co-operate, and capability-based access control and several corroboration functions are discussed [RLU94].

The communication and positioning systems may be combined or independent. The Active Badge and ParcTab systems use the same wireless infrared link for both tracking and data transfer, and the RF link of the GUIDE and RADAR systems has the same role, while most of other systems use separate channels. We believe that it is necessary to decouple the positioning and communication channels. By separation we can utilize the best solution for each problem. The uniform wireless LAN technology, for example IEEE 802.11, has cells with a range of about 100 feet indoors, and this conflicts with a goal of room-level (or better) granularity. Current outdoor wireless MAN/WAN technologies (such as cellular systems) are too coarse for good location sensing, while GPS is clearly unavailable for data communication.

There are two general approaches to make the mobile device aware of its current location. Either the system tracks the location by monitoring beacons from mobile devices and the mobile device queries a central database to get the current location, or the mobile device passively listens to beacons from the cell base-station and queries a local database for its current location. In the latter case, if the mobile device only queries a local database for location, it has complete privacy and it can choose to advertise its current location to the world or only to selected third parties.

5.2 Sensing Low-level Contexts Beyond Location

There are many types of context other than location. We observed in the last section, however, that few contexts other than location have been widely studied. Here is a list of other contexts that have been addressed and possible approaches to sense them:

- **Time**: This contextual information is not difficult to obtain, of course, from the built-in clock of the computer. Many applications correlate the location information with timestamp, such as Active Badge [WHF+92], ParcTab [WHF+92, Wei93], and Cyberguide [LKA+96, AAH+97]. Though there are other forms of time context, such as day of the week, day of the month, month of the year, season of the year, time zone, and so forth, only time-of-day information has been used as far as we know. Also, time context tends to be used together with schedule information, such as in the Office Assistant systems [YS00].

- **Nearby objects**: If the system records the locations of people and other objects, it is easy to figure out who and what are near us by just querying the location database. The Teleporting system [BRH94, HHS+99] and the context-aware Pager [BBC97, Bro98] are good demonstrations of how this context can be used.

- **Network bandwidth**: As stated before, network bandwidth is also an important computational context. There is no easy way, however, for applications to adapt to the bandwidth changes without underlying system support. Implemented as a user-level module, the Odyssey system [NSN+97] provides API calls by which applications can be notified when the network bandwidth changes. More recent work includes the Congestion Manager [ABC+00] as an in-kernel module that measures the bandwidth and notifies applications via upcalls.

- **Orientation**: Orientation is also possible to measure. A simple orientation sensor, based on two mercury switches, is attached to the Newton MessagePad [SBG98] to allow applications to adjust the display when the device orientation changes. For indoor cellular-based location-tracking systems, orientation can also be calculated, either by placing three transmitters on a rigid body at known non-collinear points of the object or by using the knowledge of the set of receivers that detected ultrasonic signals because of the hemispherical directional transmission pattern of an ultrasonic pulse [WJH97, HHS+99].

- **Other low-level contexts**: Specially designed sensors can sense some low-level types of physical context. For example, Researchers of the TEA project built a multi-sensor prototype to sense more contexts [SBG98], including a photodiode to detect *light* level, two accelerometers to provide *tilt* and *vibration* measurements, a passive IR sensor to detect the *proximity of humans*, a omni-directional microphone to detect *sound*, and other built-in sensors for *temperature*, *pressure,* and *CO gas*.

One concern is that adding these sensors to the mobile device will in turn reduce the user's mobility and may require the user's cooperation, because the additional size and weight added to the mobile device. The sensors need to be made small and unobtrusive, and the user should have the control to leave behind sensors unnecessary for her specific applications. We can, however, also deploy the sensors in the environment as public infrastructure, and provide contextual information (such as temperature) through an online service to the mobile device based on its current location.

5.3 Sensing High-level Contexts

In addition to the raw contextual information such as location, noise level and temperature, we are also interested in high-level context information such as the user's "current activity". It is, however, a big challenge to sense complex social contexts. One approach is machine vision, based on camera technology and image processing. Another possible approach is to consult the user's calendar directly to find out what the user is *supposed* to do at certain time. The user, however, is not always willing or able to put her activities in the calendar and she may not always follow the calendar. A third method is to use Artificial Intelligence techniques to recognize complex context by combining several simple low-level sensors [SBG98].

The TEA project illustrates how to recognize whether the phone or PDA is in hand, on a table or in a suitcase, by using three sensors: light, and acceleration in two directions (X and Y) [SAT+99]. The idea of the learning architecture deployed in TEA is to cluster the input signals with a neural network called the 'Kohonen Self-Organizing Map' (KSOM). The outcome of this clustering is then labeled or classified, after which it is checked by a probabilistic finite state machine [Lae99].

Such rule-based systems, however, suffer from the difficulties originating from ambiguity, boundary conditions, and undefined models. For example, the TEA system will not be able to recognize that the phone is actually in a user's pocket, not in the suitcase, because this context model is not pre-defined. Further, it is extremely hard to recognize user's emotional context, because most people are unwilling to wear the necessary intrusive biosensors [SBG98].

5.4 Sensing Context Changes

Many applications may be interested in being notified about *changes* of context. Typically, the context source monitor polls the current context and sends the changes to some context service that has a publish-subscribe-notify interface. The context service is responsible to deliver the context changes to the clients who have subscribed to the related context changes.

Different context, however, has different properties. For example, the location of a moving person may change every second while the location of a printer may not change for a year. This difference leads to the necessity of a different polling rate on different context. The current approach is to preset different polling rates either by the user or by the application at the startup [SAT+99].

These polling rates, however, are chosen in an ad-hoc way and based mostly on people's experiences and assumptions. If the polling rates can be optimized, which means that the polling happens almost in sync with the changes, it will help relieve system burdens and prolong the lifetime of battery in many sensors.

If we regard the context-generating source as a periodic information publisher, the published information depreciates over time. Brewington and Cybenko develop a formal model for monitoring such an information source [BC00]. They use previous observed information lifetimes to estimate change rates, so the monitor can decide when is a good time to poll the source to get a new copy of the information. Although they test their model for Web pages, it is certainly applicable to many information-monitoring problems, including context sensing.

All context-sensing techniques mentioned above need to be robust and reliable. Sometimes, this requirement could be hard to achieve. For example, the location system tracks the active badge worn by people. It is quite possible that the badge is just placed on a table temporarily or in a desk drawer for a long time. The system may be unable to determine that the location information is false.

6. Modeling Context Information

There are many types of context information. Their different properties lead to different ways to express and model them. To our knowledge, all the current systems use their own way to model the contextual information. It is impossible to exchange context information between them, or to notify the applications on one system about context changes based on the context sensed by another system. Most research of modeling context focuses on location information only, which we discuss in Section 6.1. In Section 6.2, we discuss the data structures used to express general context information in various context-aware systems.

6.1 Location Model

A good location model is necessary to handle object mobility and facilitate location-related queries, such as "given an object return its current location", and "given a location return a list of all the objects there". Although Geographical Information Systems (GIS) store and process geometry and topology data, they are basically *static* map databases and can not handle real-time object tracking.

There are typically two location models: a symbolic model (representing location as abstract symbols) and a geometric model (representing location as coordinates). The best choice often depends on the output format of the underlying sensor system. For example, a GPS-based system usually uses the geometric model and a system based on Active Badge usually uses the symbolic model.

For scalability and abstraction, locations are typically organized hierarchically in both models. For example, Schilit uses a symbolic model with a location containment hierarchy [Sch95] in the Active Map service. In their geometric models, Nelson and Ward use an R-tree index [Nel98] and a Quad-tree index [War99] respectively to facilitate location searching and updating. The EasyLiving project at Microsoft Research [BMK+00], however, uses a geometric model with flat layout that works well in a small area. A measurement with position and orientation information defines the geometric relationship between entities, and measurements in the graph are continuously updated as the object moves.

Trying to take advantage of both symbolic and geometric models, Leonhardt proposes a combined model [Leo98], in which a location contains both a symbolic name and geometric coordinates. The symbolic name and geometric coordinates can convert to each other via pre-defined predicates. Such a combined model shields the details of underlying sensors and can support applications that need both symbolic and geometric location information.

6.2 Data Structures

Data structures are used to express and exchange context information in the system. Although most systems use ad-hoc data structures, they typically fall into several categories as follows.

Key-value pairs. Schilit models the contextual information in a key-value pair [STW93], with an environmental variable acting as the key and the value of the variable holding the actual context data. It allows pair recursion. Such a simple model is powerful enough to allow pattern-matching queries. Mobisaic [VB94] essentially uses the same approach. The Location Information Server developed at Philips Research Laboratories [Maa97] also stores location information as attribute-value pairs in an X.500-conformant directory information tree.

Tagged encoding. A "Stick-e" note is the electronic equivalent of a Post-it note, based on the Standard Generic Markup Language (SGML) [BBC97, Pas97]. The contexts are modeled as tags and corresponding fields. The stick-e fields can recursively contain other tags and corresponding fields. The note of the <body> tag will automatically be triggered when the contextual constrains in the <require> tag are met. This model evolved into ConteXtML,[1] which is a simple XML-based protocol for exchanging contextual information between a mobile client and a server.

Object-oriented model. The GUIDE system provides an information model based on the concept of integrating an active-object model with a hypertext information model [CMD98, DCM+99]. Based on a location object and a navigation point object, this object-oriented model is specifically designed for location context. The contextual information is embedded as the states of the object, and the object provides methods to access and modify the states.

Logic-based model. A Location-oriented multimedia system [BBH97] expresses the existing contextual information in a domain-centralized database using an entity-relationship data model implemented in Prolog. Context data are expressed as facts in a rule-based system. It is possible to add new rules as well as submit queries to the database.

[1]http://www.cs.ukc.ac.uk/research/infosys/mobicomp/fnc/ConteXtML.html

Others. The TEA project [SAT+99] models the context data in a layered structure. Raw contextual data from sensors (first layer) are time-stamped. The second layer "Cue" gathers these data and outputs a symbolic or sub-symbolic value, while a set (finite or infinite) of possible values for each cue are defined. Each cue is dependent on one single sensor but using the data of one sensor, multiple cues can be calculated. The third layer "Context" is derived from available cues across all sensors. The context is described by a set of two-dimensional vectors, where each vector consists of a symbolic value describing the situations and a number indicating the probability. Finally, certain scripting primitives are provided to the highest application layer.

7. System Infrastructure

It is observed that the sheer diversity of exploitable contexts and the plethora of sensing technologies are actually working against the deployment of context-aware systems [Pas98]. It is necessary to decouple the application and the actual context sensing part, because of the large development overhead of interacting with a variety of sensors to capture the context, interpreting it to the desired format, and using it in a meaningful way. Also, the decoupling is important to generalize the system to other applications.

To separate the low-level sensor data processing from high-level applications, it is necessary to introduce a middleware layer whose functionalities are collecting raw sensor information, translating it to an application-understandable format, and disseminating it to interested applications. There are typically two approaches, centralized and distributed, with different emphasis in various systems.

7.1 Centralized Architecture

The simplest way to decouple is to use a centralized context server, which provides contextual information to the applications. Schilit's mobile application customization system [STW93] contains dynamic environment servers, which manage a set of variable names and values (an environment) and delivers updates to clients that have previously shown interest by subscribing to the server. Typically there is one environment for each user, plus environments for rooms, workgroups, and so forth. Clients use RPC to get variables from servers, and servers use RPC to provide callbacks to clients. Essentially, the Situated Computing (SitComp) Service [HNR97] and Contextual Information Service (CIS) [Pas98] take a similar approach. Both of them act as the middleware that acquires raw contextual information from sensors and provides interpreted context to applications via a standard API. They can also monitor the context changes and send events to interested applications.

The Location Information Server [Maa97] offers a set of generic location retrieval and notification services to the application, based on the X.500 directory service and the lightweight directory access protocol LDAP, since these are becoming the standard attribute-value-pair retrieval mechanisms for Internet and Intranet environments. The Directory System Agent (DSA) offers the directory service to the Directory User Agent (DUA). The content of directory is not held by a single DSA but may be distributed over a set of co-operating DSAs to enable the establishment of a worldwide global directory service. The DSAs talk to each other using the Directory System Protocol (DSP) to hide the physical data distribution from the directory user.

A "context widget" is a software component that provides applications with access to context information from their operating environment [SDA99]. Each context widget has a state that is a set of attributes and a behavior that is a set of callback functions triggered by context changes. The widget obtains raw contextual information from sensors and passes them either to interpreters or to servers for aggregation. Interpreters and servers provide simple APIs for the application to access [DSF+99]. The infrastructure uses the HTTP protocol for communication and the eXtensible Markup Language (XML) as the language model.

Bacon et al. successfully integrated an event-based distributed system and a mobile-agent system (The Tube) with location awareness [BBH97]. Their Location Service has an Event Server Module monitoring the context change, and notifies the Event Client Module sitting at the Location Client. When a user moves to a new place, her representing agent will follow her and carry related objects to a nearby workstation. The mobile agent can also bind with new local resources, and yet maintain the previous multimedia connections, such as video and audio feeds.

The centralized approach usually has scalability problem. José and Davies propose a location-domain hierarchical model [JD99]. To support notions of proximity over the same infrastructure, and mechanisms to locate services based on their physical location, the service scopes are represented in terms of physical space, instead of network domain. The Location-Based Service (LBS) servers can be federated based on the relationship of the locations they represent. This model works fine with location context because location has the property of the partially ordered "contains" relationship.

7.2 Distributed Architecture

Instead of maintaining all context information in one centralized place, a distributed architecture allows context be held at several places to avoid potential bottleneck. The Rome system developed at Stanford [HLP+99] is based on the concept of a context trigger, which consists of a condition and an action. It is unique in that Rome allows decentralized evaluation of triggers by embedding triggers in end devices. This approach, however, does not allow context sharing and requires the end device to have the capability to sense and process all of the necessary raw contextual information, which may not be efficiently achieved, especially for a complicated trigger and a simple device.

A user-centric architecture [ST93] has a personal User Agent manage and control that user's personal and location information. Although the User Agent is a well-known service that external clients can ask a variety of information about the user it represents, the user determines the control policies of her agent. Clients can also query the Location Query Service (LQS), which is organized by regions with a centralized server, called the Location Broker, running in each region. The Location Broker can notify the clients via the callback RPC handle whenever the answer to the submitted query changes.

Spreitzer and Theimer also propose an alternative architecture based on multicast [ST93]. The clients just multicast their location queries to all members of a domain's multicast group. Interested parties will anonymously listen to the domain's multicast group to hear location queries. They will answer a query if they match it and if their current privacy policy allows it. The disadvantage of this approach is increased computation and communication. The advantage is that no central server is needed, and members can still maintain privacy.

8. Security and Privacy

There are two key problems in context-aware system security [ST93]: ensuring the accuracy of location information and identities, and establishing secret communications. Authenticating the supplied location information is difficult because today's sensor systems typically only detect things such as active badges that can be removed from the mobile object they represent. In addition to protecting the content of the communication, the address of the content should also be protected to prevent leaking of location information.

Not surprisingly, most people do not like the idea of being precisely located at anytime, by anyone, especially when the location data is logged. It is important to address privacy issues in context-aware computing. Few existing context-sensitive systems, however, provide a satisfactory solution at current stage, while many other systems choose to ignore security and privacy concerns.

"Perfect" privacy guarantees are in general hard (and expensive) to provide [ST93]. User should be able to have the control over their contextual information and over who may gain access to it. The system architecture needs to provide user-controllable trade-offs between privacy guarantees and both functionality and efficiency. But it is difficult to be specific about what context information should be visible to who, and when.

Indeed, "the problem is that technology in itself is rarely inherently bad: it is just that it can be used for good or bad purposes" [WHF+92]. Many existing systems already may be abused to compromise people's privacy, due to bad management policy. When technologies are vulnerable, legislation may be necessary to ensure that the user's contextual information cannot be misused.

9. Summary

To better understand context awareness, we surveyed the literature in this area. We defined the terms context and context awareness, listed the context-aware applications that have been built, discussed current approaches to sense and model the context, and looked into supporting infrastructures and security and privacy issues. We found that the research of accurately discovering context, efficiently disseminating contextual information, and making use of the available context, are still at the early stages. We believe, however, that context awareness is a key factor for new applications in the area of ubiquitous computing.

References

[AAH+97] Gregory D. Abowd, Christopher G. Atkeson, Jason Hong, Sue Long, Rob Kooper, and Mike Pinkerton. Cyberguide: A mobile context-aware tour guide. *Wireless Networks*, 3(5):421-433, October 1997.

[ABC+00] David Andersen, Deepak Bansal, Dorothy Curtis, Srinivasan Seshan, and Hari Balakrishnan. System support for bandwidth management and content adaptation in Internet applications. In *Proceedings of 4th Symposium on Operating Systems Design and Implementation*, pages 213-226, San Diego, CA, October 2000. USENIX Association.

[AGK94] Abhaya Asthana, Mark Cravatts, and Paul Krzyzanowski. An indoor wireless system for personalized shopping assistance. In *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications*, pages 69-74, Santa Cruz, California, December 1994. IEEE Computer Society Press.

[AGS+93] Norman Adams, Rich Gold, Bill N. Schilit, Michael M. Tso, and Roy Want. An infrared network for mobile computers. In *Proceedings of USENIX Mobile & Location-Independent Computing Symposium*, pages 41-51, Cambridge, Massachusetts, August 1993. USENIX Association.

[AJL+97] Michael D. Addlesee, Alan Jones, Finnbar Livesey, and Ferdinando Samaria. The ORL Active Floor. *IEEE Personal Communications*, 4(5):35-41, October 1997.

[BBC97] Peter J. Brown, John D. Bovey, and Xian Chen. Context-aware applications: from the laboratory to the marketplace. *IEEE Personal Communications*, 4(5):58-64, October 1997.

[BBH97] Jean Bacon, John Bates, and David Halls. Location-oriented multimedia. *IEEE Personal Communications*, 4(5):48-57, October 1997.

[BC00] Brian Brewington and George Cybenko. Keeping up with the changing Web. *IEEE Computer*, 33(5):52-58, May 2000.

[Bed95] Benjamin B. Bederson. Audio augmented reality: A prototype automated tour guide. In *Proceedings of Conference on Human Factors and Computing Systems, CHI '95*, pages 210-211, Denver, CO, May 1995. ACM Press.

[Bei00] Michael Beigl. MemoClip: A location-based remembrance appliance. *Personal Technologies*, 4(4):230-233, September 2000.

[BHE00] Nirupama Bulusu, John Heidemann, and Deborah Estrin. GPS-less low-cost outdoor localization for very small devices. *IEEE Personal Communications*, 7(5):28-34, October 2000.

[BHM+97] H.W. Peter Beadle, B. Harper, G. Q. Maguire Jr., and J. Judge. Location aware mobile computing. In *Proceedings of IEEE/IEE International Conference on Telecommunications, (ICT'97)*, Melbourne, Australia, April 1997.

[BMK+00] Barry Brumitt, Brian Meyers, John Krumm, Amanda Kern, and Steven Shafer. EasyLiving: Technologies for intelligent environments. In *Proceedings of Second International Symposium on Handheld and Ubiquitous Computing, HUC 2000*, pages 12-29, Bristol, UK, September 2000. Springer Verlag.

[BP00] Paramvir Bahl and Venkata N. Padmanabhan. Radar: An in-building RF-based user location and tracking system. In *Proceedings of IEEE INFOCOM 2000*, Tel-Aviv, Israel, March 2000. IEEE Computer Society Press.

[BRH94] Frazer Bennett, Tristan Richardson, and Andy Harter. Teleporting - making applications mobile. In *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications*, pages 82-84, Santa Cruz, California, December 1994. IEEE Computer Society Press.

[Bro98] Peter J. Brown. Triggering information by context. *Personal Technologies*, 2(1), March 1998.

[CK99] P. Couderc and Anne-Marie Kermarrec. Enabling context-awareness from network-level location tracking. In *Proceedings of First International Symposium on Handheld and Ubiquitous Computing, HUC'99*, pages 67-73, Karlsruhe, Germany, September 1999. Springer Verlag.

[CM00] Paul Castro and Richard Muntz. Managing context data for smart spaces. *IEEE Personal Communications*, 7(5):44-46, October 2000.

[CMD98] Keith Cheverst, Keith Mitchell, and Nigel Davies. Design of an object model for a context sensitive tourist GUIDE. In *Proceedings of Workshop on Interactive Applications of Mobile Computing (IMC'98)*, Rostock, Germany, November 1998. Neuer Hochschulschriftverlag.

[DA99] Anind K. Dey and Gregory D. Abowd. Towards a Better Understanding of context and context-awareness. Technical Report GIT-GVU-99-22, Georgia Institute of Technology, College of Computing, June 1999.

[DA00] Anind K. Dey and Gregory D. Abowd. CybreMinder: A context-aware system for supporting reminders. In *Proceedings of Second International Symposium on Handheld and Ubiquitous Computing, HUC 2000*, pages 172-186, Bristol, UK, September 2000. Springer Verlag.

[DCM+99] Nigel Davies, Keith Cheverst, Keith Mitchell, and Adrian Friday. Caches in the air: Disseminating tourist information in the GUIDE system. In *Proceedings of Second IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, Louisiana, February 1999. IEEE Computer Society Press.

[DFS+99] Anind K. Dey, Masayasu Futakawa, Daniel Salber, and Gregory D. Abowd. The Conference Assistant: Combining Context-Awareness with Wearable Computing. In *Proceedings of the 3rd International Symposium on Wearable Computers (ISWC '99)*, pages 21-28, San Francisco, CA, October 1999. IEEE Computer Society Press.

[DSF+99] Anind K. Dey, Daniel Salber, Masayasu Futakawa, and Gregory D. Abowd. An architecture to support context-aware applications. Technical Report GIT-GVU-99-23, Georgia Institute of Technology, College of Computing, June 1999.

[Fit93] George W. Fitzmaurice. Situated information spaces and spatially aware palmtop computers. *Communications of the ACM*, 36(7):39-49, July 1993.

[HHS+99] Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, and Paul Webster. The anatomy of a context-aware application. In *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 59-68, Seattle, WA, August 1999. ACM Press.

[HLP+99] Andrew C. Huang, Benjamin C. Ling, Shankar Ponnekanti, and Armando Fox. Pervasive computing: What is it good for? In *Proceedings of the ACM International Workshop on Data Engineering for Wireless and Mobile Access*, pages 84-91, Seattle, WA, August 1999. ACM Press.

[HNR97] Richard Hull, Philip Neaves, and James Bedford-Roberts. Towards situated computing. In *Proceedings of the First International Symposium on Wearable Computers*, Cambridge, Massachusetts, October 1997. IEEE Computer Society Press.

[JD99] Rui José and Nigel Davies. Scalable and flexible location-based services for ubiquitous information access. In *Proceedings of First International Symposium on Handheld and Ubiquitous Computing, HUC'99*, pages 52-66, Karlsruhe, Germany, September 1999. Springer Verlag.

[Lae99] Kristof Van Laerhoven. Online adaptive context awareness, starting with low-level sensors. Licentiaats thesis at the University of Brussels, Brussels, Belgium, May 1999.

[Leo98] Ulf Leonhardt. Supporting location-awareness in open distributed systems. PhD thesis, Imperial College of Science, Technology and Medicine, University of London, May 1998.

[LKA+96] Sue Long, Rob Kooper, Gregory D. Abowd, and Christopher G. Atkeson. Rapid prototyping of mobile context-aware applications: the Cyberguide case study. In *Proceedings of the Second Annual International Conference on Mobile Computing and Networking*, pages 97-107, White Plains, NY, November 1996. ACM Press.

[Maa97] Henning Maaß. Location-aware mobile applications based on directory services. In *Proceedings of the Third Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 23-33, Budapest, Hungary, September 1997. ACM Press.

[MS00] Natalia Marmasse and Chris Schmandt. Location-aware information delivery with ComMotion. In *Proceedings of Second International Symposium on Handheld and Ubiquitous Computing, HUC 2000*, pages 157-171, Bristol, UK, September 2000. Springer Verlag.

[Nel98] Giles John Nelson. Context-aware and location systems. PhD thesis, Clare College, University of Cambridge, January 1998.

[Nor99] Donald A. Norman. *The Invisible Computer: why good products can fail, the personal computer is so complex, and information appliances are the solution*. The MIT Press, 1999.

[NSN+97] Brian D. Noble, M. Satyanarayanan, Dushyanth Narayanan, James Eric Tilton, Jason Flinn, and Kevin R. Walker. Agile application-aware adaptation for mobility. In *Proceedings of the Sixteenth ACM Symposium on Operating Systems Principles*, pages 276-287, Saint Malo, France, October 1997. ACM Press.

[OA00] Robert J. Orr and Gregory D. Abowd. The Smart Floor: A mechanism for natural user identification and tracking. In *Proceedings of the 2000 Conference on Human Factors in Computing Systems (CHI 2000)*, The Hague, Netherlands, April 2000. ACM Press.

[OS98] Reinhard Oppermann and Marcus Specht. Adaptive support for a mobile museum guide. In *Proceedings of Workshop on Interactive Applications of Mobile Computing (IMC '98)*, Rostock, Germany, November 1998. Neuer Hochschulschriftverlag.

[OS00] Reinhard Oppermann and Marcus Specht. A context-sensitive nomadic exhibition guide. In *Proceedings of Second International Symposium on Handheld and Ubiquitous Computing, HUC 2000*, pages 127-142, Bristol, UK, September 2000. Springer Verlag.

[Pas97] Jason Pascoe. The Stick-e note architecture: Extending the interface beyond the user. In *Proceedings of the 1997 International Conference on Intelligent User Interfaces*, pages 261-264, Orlando, FL, January 1997. ACM Press.

[Pas98] Jason Pascoe. Adding generic contextual capabilities to wearable computers. In *Proceedings of the Second International Symposium on Wearable Computers*, Pittsburgh, Pennsylvania, October 1998. IEEE Computer Society Press.

[PCB00] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The Cricket location-support system. In *Proceedings of the Sixth Annual ACM International Conference on Mobile Computing and Networking*, Boston, MA, August 2000. ACM Press.

[PJ96] Charles E. Perkins and David B. Johnson. Mobility support in IPv6. In *Proceedings of the second annual international conference on Mobile computing and networking*, pages 27-37, White Plains, NY, November 1996. ACM Press.

[PMR98] Jason Pascoe, David Morse, and Nick Ryan. Developing personal technology for the Field. *Personal Technologies*, 2(1), March 1998.

[PRM99] Jason Pascoe, Nick Ryan, and David Morse. Issues in developing context-aware computing. In *Proceedings of First International Symposium on Handheld and Ubiquitous Computing, HUC'99*, pages 208-221, Karlsruhe, Germany, September 1999. Springer Verlag.

[PSG00] Thai-Lai Pham, Georg Schneider, and Stuart Goose. Exploiting location-based composite devices to support and facilitate situated ubiquitous computing. In *Proceedings of Second International Symposium on Handheld and Ubiquitous Computing, HUC 2000*, pages 143-156, Bristol, UK, September 2000. Springer Verlag.

[RLU94] Mike Rizzo, Peter F. Linington, and Ian A. Utting. Integration of location services in the open distributed office. Technical Report 14-94, Computing Laboratory, University of Kent, Canterbury, UK, August 1994.

[SAT+99] Albrecht Schmidt, Kofi Asante Aidoo, Antti Takaluoma, Urpo Tuomela, Kristof Van Laerhoven, and Walter Van de Velde. Advanced interaction in context. In *Proceedings of First International Symposium on Handheld and Ubiquitous Computing, HUC'99*, pages 89-101, Karlsruhe, Germany, September 1999. Springer Verlag.

[SAW94] Bill Schilit, Norman Adams, and Roy Want. Context-aware computing applications. In *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications*, pages 85-90, Santa Cruz, California, December 1994. IEEE Computer Society Press.

[SBG98] Albrecht Schmidt, Michael Beigl, and Hans-W. Gellersen. There is more to context than location. In *Proceedings of Workshop on Interactive Applications of Mobile Computing (IMC'98)*, Rostock, Germany, November 1998. Neuer Hochschulschriftverlag.

[Sch95] William Noah Schilit. A system architecture for context-aware mobile computing. PhD thesis, Columbia University, May 1995.

[SDA99] Daniel Salber, Anind K. Dey, and Gregory D. Abowd. The context toolkit: Aiding the development of context-enabled applications. In *Proceeding of the CHI 99 Conference on Human Factors in Computing Systems: The CHI is the Limit*, pages 434-441, Pittsburgh, PA, May 1999. ACM Press.

[SKM+93] M. Satyanarayanan, James J. Kistler, Lily B. Mummert, Maria R. Ebling, Puneet Kumar, and Qi Lu. Experience with disconnected operation in a mobile computing environment. In *Proceedings of USENIX Mobile & Location-Independent Computing Symposium*, pages 11-28, Cambridge, Massachusetts, August 1993. USENIX Association.

[ST93] Mike Spreitzer and Marvin Theimer. Providing location information in a ubiquitous computing environment. In *Proceedings of the Fourteenth ACM Symposium on Operating Systems Principles*, pages 270-283, Asheville, NC, December 1993. ACM Press.

[STE99] Scott Stillman, Rawesak Tanawongsuwan, and Irfan Essa. A system for tracking and recognizing multiple people with multiple cameras. In *Proceedings of Second International Conference on Audio-Vision-based Person Authentication*, Washington, DC, April 1999.

[STW93] Bill N. Schilit, Marvin M. Theimer, and Brent B. Welch. Customizing mobile applications. In *Proceedings of USENIX Mobile & Location-Independent Computing Symposium*, pages 129-138, Cambridge, Massachusetts, August 1993. USENIX Association.

[VB94] Geoffrey M. Voelker and Brian N. Bershad. Mobisaic: An information system for a mobile wireless computing environment. In *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications*, pages 185-190, Santa Cruz, California, December 1994. IEEE Computer Society Press.

[War99] Andrew Martin Robert Ward. Sensor-driven computing. PhD thesis, Corpus Christi College, University of Cambridge, May 1999.

[Wei91] Mark Weiser. The computer for the 21st century. *Scientific American*, 94-104, September 1991.

[Wei93] Mark Weiser. Some computer science issues in ubiquitous computing. *Communications of the ACM*, 36(7):75-84, July 1993.

[WHF+92] Roy Want, Andy Hopper, Veronica Falcão, and Jonathan Gibbons. The Active Badge location system. *ACM Transactions on Information Systems*, 10(1):91-102, January 1992.

[WJH97] Andy Ward, Alan Jones, and Andy Hopper. A new location technique for the active office. *IEEE Personal Communications*, 4(5):42-47, October 1997.

[WL98] Jay Werb and Colin Lanzl. A positioning system for finding things indoors. *IEEE Spectrum*, 35(9):71-78, September 1998.

[WSA+95] Roy Want, Bill N. Schilit, Norman I. Adams, Rich Gold, Karin Petersen, David Goldberg, John R. Ellis, and Mark Weiser. An overview of the PARCTAB ubiquitous computing experiment. *IEEE Personal Communications*, 2(6):28-43, December 1995.

[WSA+96] Roy Want, Bill N. Schilit, Norman I. Adams, Rich Gold, Karin Petersen, David Goldberg, John R. Ellis, and Mark Weiser. The ParcTab Ubiquitous Computing Experiment. In Tomasz Imielinski and Henry F. Korth, editors, *Mobile Computing*, chapter 2, Kluwer Academic Publishers, 1996.

[YS00] Hao Yan and Ted Selker. Context-aware office assistant. In *Proceedings of the 2000 International Conference on Intelligent User Interfaces*, pages 276-279, New Orleans, LA, January 2000. ACM Press.