A Context-Based Information Assistant
Human-Centered Computing Course, Fall 1999

Jason I. Hong
525 Soda Hall
UC Berkeley Computer Science Division
Berkeley, CA 94720-1776

+1 510 643 7354
jasonh@cs.berkeley.edu
http://www.cs.berkeley.edu/~jasonh/

Abstract

Finding the right piece of information at the right time can be quite a complicated task. Conventional information retrieval techniques can help us find relevant information, but interacting with these systems is often an awkward experience, especially for novices. The challenge of designing new techniques for finding contextually useful information appropriate to the task at hand is the motivation for this research.

We believe that an autonomous software agent can help us in this endeavor. By observing our communication with other people, and by using contextual information about us, we believe that an autonomous software agent can be developed that can proactively search, filter, organize, and present information useful to us in our current activity. We call such a software agent a Context-based Information Agent (CIA).

In this paper, we outline a low-fidelity evaluation we performed to explore some of the relevant issues; a high-level design; a first cut at implementing a speech-based information agent; related work; and a further look into the design space.

Introduction

Inter-networked ubiquitous computing devices are providing us with fantastic opportunities for creating, capturing, storing, analyzing, and sharing information. However, finding the right information at the right time can be difficult and time consuming. Conventional search engines only partially address this problem. One drawback is that they require the user to explicitly enter keywords to initiate a search. The interaction can be awkward at times, since users cannot always be at a computer to type in keywords. It can also be downright frustrating when the computer presents thousands of results, most of which are wildly out of context.

The most significant drawback to conventional search engines, though, is that they take full command of our most precious resource: *our attention.* It is difficult to have any kind of meaningful interaction with other people while we are in front of our desktop computers, staring at the monitor, and sifting through the results.

We contend that both natural human-to-human communication and situational context can be used as the basis for searching and retrieving information. Towards these ends, we describe an autonomous software agent that can, independently and in parallel with you, search, filter, organize, and proactively present useful information based on your current activities and on your current context. We call such a software agent a Context-based Information Assistant (CIA).

At a high level, the challenge here is threefold. The first challenge is to determine what kinds of human-human communication and situational context is useful in driving searches, and then capturing and managing this input. The second is to determine how to use this input to gather useful information effectively. The final challenge is to determine how to present the results in a contextually useful way without distracting the user from the task at hand.

To further explore these issues, we performed a low-fidelity evaluation, described in the next section. The remaining sections detail a high-level design for the CIA; a first-cut at implementation; related work; a second look into the design space; and a brief summary.

Low-Fidelity Prototype

To test the desirability and usefulness of the CIA, we ran a low-fidelity prototype, in which a person acted as the software agent. During a weekly research meeting, a tape recording was made of the conversation (with permission). After the meeting, the recording was used to make informed searches on the web. The results of this search were assembled and presented to the meeting members, along with an anonymous questionnaire for feedback.

In the presentation of the results, each of the links was grouped by time and topic of conversation (see Figure 1). Under each topic, the results were further grouped by what the link represented: an answer to a question, a paper or web reference explicitly made, or web pages related to the current conversation. The rationale for these three categories is based on our observations of what people said during the meeting. One thing we noticed is that people often made explicit references to web pages or papers they had read. For example, they would say things like "[t]here's a paper from Xerox PARC about tagging physical objects" or "[t]here's that Krugman paper about babysitting co-ops." Another observation we made is that people sometimes asked questions that no one knew the answer to. For example, one question was "[d]o you happen to know what percent of the world population lives in urban areas?"

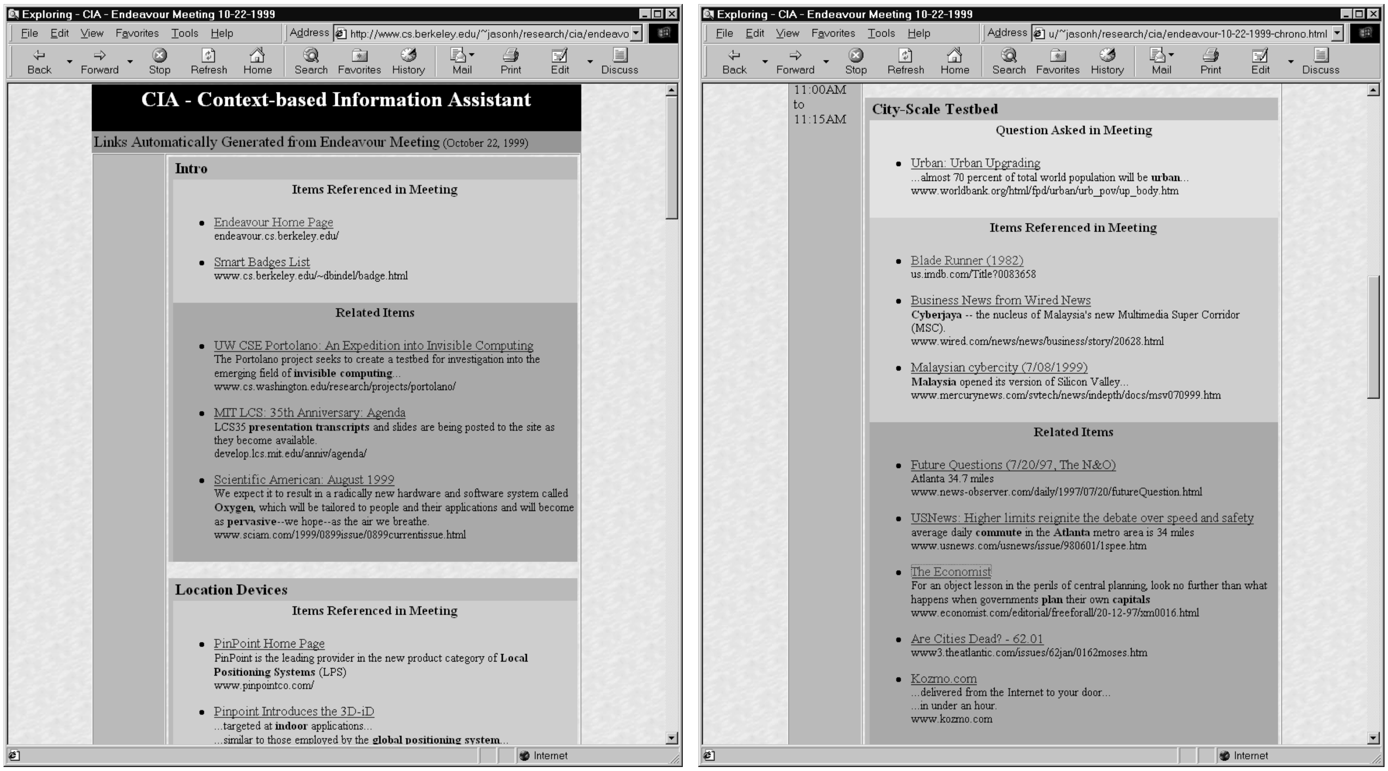

Figure 1 – Low-fidelity Prototype of Results Generated by CIA

From the questionnaire, we learned that people really liked the idea of having an autonomous agent help them find information, and that organizing the results by time, topic, and link type was very useful. There would be additional value if the agent could work in real-time, and if more types of context, such as the speaker, could be used to index and access the results. In general, people were mostly comfortable about being recorded, as they perceived a benefit to having an agent assist them in their activities, especially one that needed little or no explicit management. However, there needs to be clear boundaries between what the agent can and cannot do, and that people should always have the option of turning the agent on and off.

In analyzing the recording, we also observed some interesting changes in people's behavior. For example, when a comment was made that people did not want recorded, someone immediately said "[s]trike that from the record!" Another comment, made after someone asked a question that no one knew the answer, was "[t]hat would be a good thing for our intelligent agent to find out." These comments suggest some intriguing design directions for the CIA, indicating that the agent should have some method for direct interaction.

High-Level Design

Using the knowledge gained from the low-fidelity evaluation, we have developed a first-cut high-level design, as well as a first-cut block diagram depicting a simple version of the system (see Figure 2). The high-level design (top part of Figure 2) consists of five different subsystems: Interaction, Query Extraction, Context, Meta-Search, and Presentation. The Interaction subsystem takes all of the user input and converts it into standard data formats. The data is forwarded to the Query Extraction subsystem, which tries to process the data and generate meaningful queries. These queries are forwarded to the Meta-Search subsystem, which uses the queries to search a range of information sources. These information sources can include web search engines, local databases, local files, and digital libraries. The Context subsystem is used to expand certain queries, as well as filter out certain results. The results are then forwarded to the Presentation subsystem, which assembles and organizes the results. Here, the Context subsystem is used to help determine the importance of individual results, as well as to determine the best method for presenting the information. For example, if someone is speaking, then the system should not draw attention to itself while they are speaking.

There are currently many research problems in the area of context awareness. The research community has not yet completely answered many questions, including what kinds of context are useful, how to represent the context, and, most importantly, how to use the context for searching, filtering, and presenting the information. We hope to contribute some insights into these questions by performing some more low-fidelity prototypes, as well as by experimenting with a variety of capture devices, algorithms, and schemas.

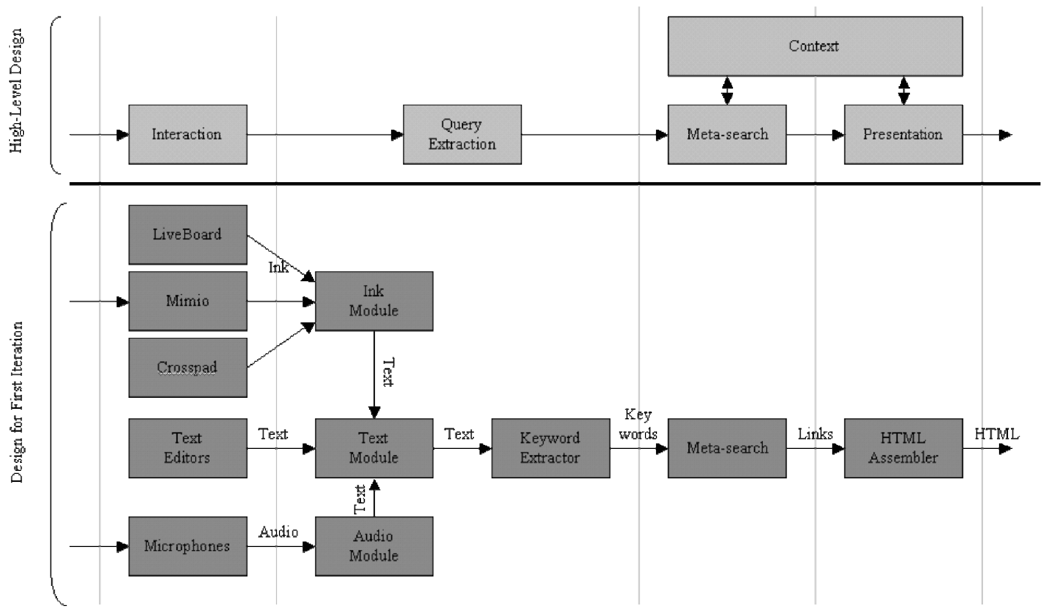

Figure 2 – Preliminary Design for CIA

First Iteration at Implementation

Using this high-level design, we implemented a first cut of the CIA. This implementation is speech-based, using IBM ViaVoice 98 in dictation mode[1], and uses the Google web search engine as the back end (See Figure 3 next page). When started, one portion of the system waits for words from the speech-recognition system. When a word is recognized, it simply adds it to a list of words. At the same time, another portion of the system is continually looking at a sliding window of these words, extracting keywords from them, and sending them to a web search engine. We decided to use a sliding window, since traditional information retrieval techniques don't seem to work as well on continuous input like speech. The search results are parsed and sent into a results pool. The results pool uses simple heuristics to rank results. For example, the first hit from a search is given a higher score than the subsequent ones, and results that have been in the pool for a longer time are penalized. After the results are sent to the results pool, the pool is queried for the best hits so far, which are displayed on screen.

Unfortunately, the speech recognition technology is not quite up to par. There are two separate problems. The first is that words are often mis-recognized. This problem can be addressed by retrieving the N-best matches instead of just the best match. The second problem is that the words that are recognized often aren't "interesting" keywords that could be used for searching.

[1] We tried using an alternative, the Nuance speech recognition toolkit, but this toolkit was constrained by requiring the designer to provide a speech grammar.

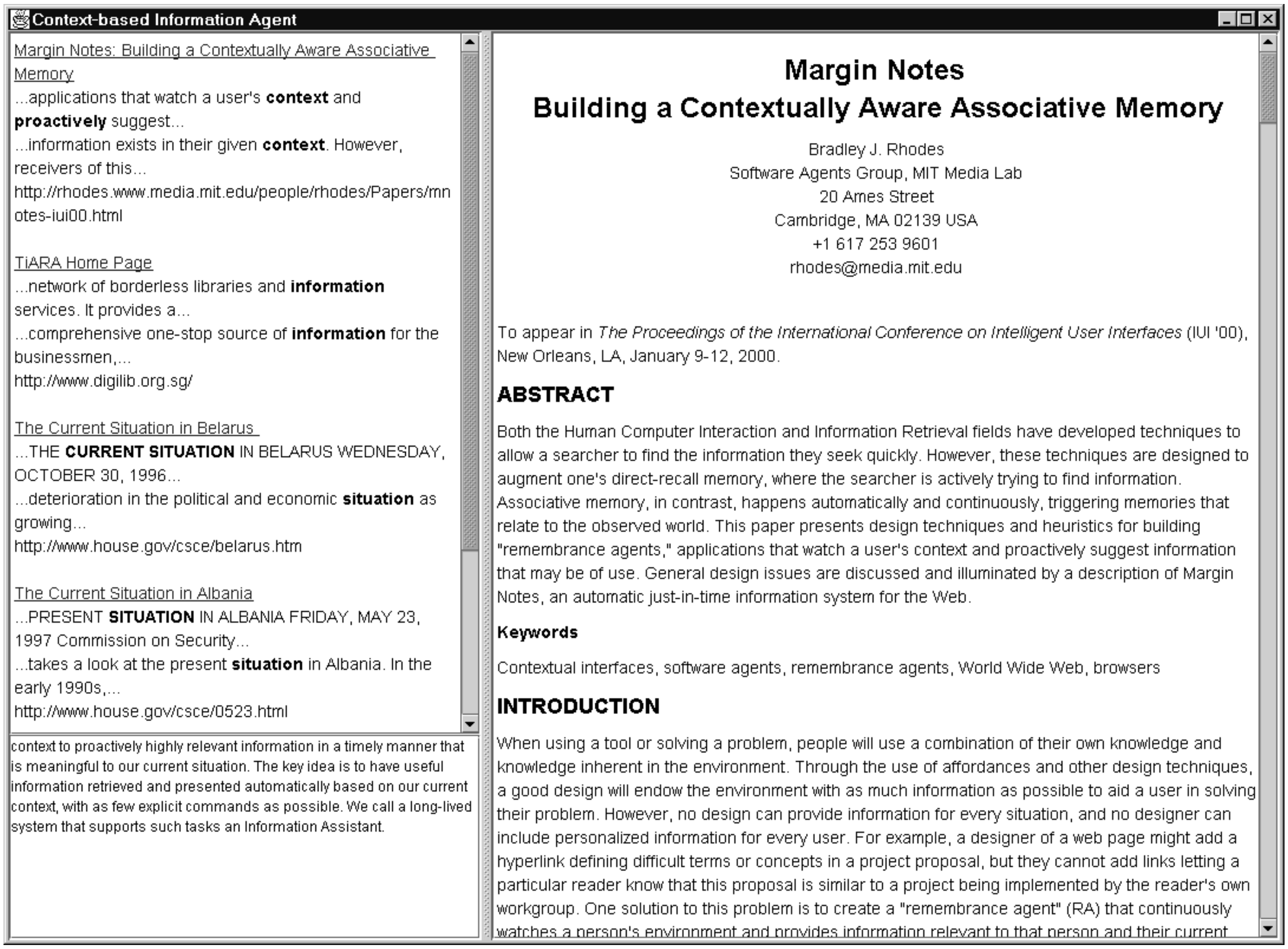

Figure 3 – First cut at Implementation. The top-left panel is the search results. The bottom-left panel is a transcript of recognized words. The panel on the right is a web browser.

To address the problem of “interesting” keywords, we created a webcrawler that would crawl local web pages, extract words, and use auto-pronunciation tools to add the extracted words to the dictionary. The conjecture is that local web pages reflect our vocabulary. For example, a personal home page is highly likely to contain words that its owner would say. Pages that are “close” in terms of link distance are also likely to contain such words. By gathering these words, a dictionary can be created that can improve speech recognition.

This is the current state of the system. Unfortunately, the system often does not return the crawled words. As mentioned above, retrieving the N-best matches seems to be a good approach, but we have not yet determined how to do so or if it is possible to do so in ViaVoice.

An alternative path to take is to perform another low-fidelity Wizard of Oz evaluation. A basic implementation exists for the query-extraction, meta-search, and presentation subsystems. It is possible that we could simulate “improved” speech recognition by having a human listen in on a conversation and type in a transcript.

Related Work

The CIA is most closely related to the Remembrance Agent (RA) [12]. The RA is an autonomous agent that is constantly monitoring what is being typed into an Emacs editor window. The agent looks at what is being typed, and is continually searching the user's personal text files as well as other Remembrance Agents to find relevant documents. A follow-up project is Margin Notes [11], a web proxy that modifies incoming web pages and adds suggestion boxes that are linked to pre-indexed personal information.

A related system is Letizia [7], an autonomous agent that monitors what web pages are being explored, and performs a breadth-first search from the current page to find nearby pages that are likely to be relevant. These pages are then displayed in a separate window, in a manner similar to channel surfing. While interesting in its approach, Letizia has some of the same problems as the Remembrance Agent, namely that it draws too much attention to itself and lacks rigorous evaluation.

In some respects, the CIA is conceptually similar to the Forget-me-not memory prosthesis developed by Lamming and Flynn [6]. The Forget-me-not is a computer system designed to support human memory. A ParcTab [16] was instrumented to automatically capture certain kinds of information, such as where you are, who is with you, what time it is, and for simple types of events, what happened. For example, the Forget-me-not could record that at 10:30 you received mail from Bob, and that at 10:35 you called extension 511 from phone extension 518. The Forget-me-not system also provided a way to filter the information, displaying all events that pass the filter in an iconic manner.

In the area of user interaction, the CIA is related to work in creating multimodal interfaces, that is interfaces that simultaneously support alternative input modalities, such as using pen and speech together. The difference is that research in multimodal interfaces is trying to find better ways for people to explicitly interact with computers, whereas we are looking at ways for leveraging communication between humans, using what people say and write to each other as input. Our goal is to have a person explicitly interact with the CIA only minimally.

Two conceptually related systems are VOIR [2] and XLibris [15], which use a novel technique for generating hypertext links. The key observations are that links are expensive to manually construct in a corpus, and that identifying useful and relevant links while reading takes up too much effort. Instead, links are generated on the fly based on user interaction. In the case of VOIR, links are automatically generated from a user interest profile. In the case of XLibris, links are automatically generated by text that is highlighted while reading. In both cases, the user never explicitly searches: documents are instead found opportunistically.

There has been a fair amount of work in the area of context awareness. Most of the research applications developed so far rely solely on position for context [1, 9, 14, 17, 18]. A recent direction has been to recognize objects, either through computer vision, through physically tagging the objects [16], or using a mixture of these two techniques [3, 10]. One new innovation has been the Context Toolkit [13], which introduces a layer of abstraction that mediates between sensors and applications, allowing applications to use different kinds of sensors without having to change any source code. However, there is still a great deal of work that needs to be done in the field of context awareness. In developing the CIA, we hope to expand the research community's understanding of what kinds of context are useful for applications to use, how to capture and interpret this context, and how to use this context in a meaningful manner.

The CIA will also draw upon a fair amount of research in artificial intelligence. Some promising directions to investigate are decision-theoretic methods and Bayesian modeling [4, 5]. We believe that these techniques can be used to help in searching and filtering information, as the CIA will need to have some model of the user and some model of the current context, as well as a foundation for making decisions about which pieces of information are important and which are not. The difference between these applications and the CIA is that the CIA is looking to find related information based on content instead of trying to assist the user accomplish a task with the interface.

Another Look at the Design Space

Based on the low-fidelity evaluation, conversations, brainstorming, and related work, we have made a first cut at partitioning the design space into three different parts: input, search, and presentation. The input portion deals with understanding what kinds of human-human communication and situational context are useful to capture, as well as how to capture it in a non-intrusive manner. For example, in the Remembrance Agent [12], the communication is direct keyboard input. In XLibris [14], the act of highlighting, a form of communication to oneself, is used to find more information. Both these systems take good approaches, in that they observe and use information already provided, minimizing the amount of direct interaction to the agent.

The search portion deals with understanding how to use the input effectively, and what information sources to use. For using the input, the simplest approach is to spot keywords and use those in the search. This is the method used by the systems described above. An alternative approach would be to see if the richer input could be used to formulate better queries. The low-fidelity evaluation revealed two examples: detecting questions and detecting explicit references made during a meeting. For what information sources to use, the design decision is to determine the scope of the information. Will the information be retrieved from personal files, from group files, or from files globally accessible?

The presentation portion deals with taking the results of the search and presenting it to the user in a contextually useful and non-intrusive manner. There are at least two high-level design decisions to be made here. First, the results can be shown either synchronously or asynchronously. For example, the agent could show the results as you do your work, or it could do it periodically, such as every morning. Second, the results need to be presented in a manner that is in context of the task, in a way that minimizes the amount of attention needed.

Summary

Most search engines require users to be sitting at a desktop computer, explicitly typing in keywords. Furthermore, it is often the case that far too many results, completely meaningless to our current context, are presented to us. Most importantly, however, is that conventional search engines require the full attention of the user, hampering human-to-human communication.

In this paper, we describe an alternative approach, an autonomous software agent that uses human-human communication and situational context to help drive searches as we are performing a task. We describe a first cut at creating a speech-based implementation, and another look at the design space for such agents.

The URL for this project is http://www.cs.berkeley.edu/~jasonh/research/cia

References

1. Abowd, G.D., Atkeson, C.G., Hong, J., Long, S., Kooper, R., and Pinkerton, M. Cyberguide: A Mobile Context-Aware Tour Guide. ACM Wireless Networks 3, 421-433.
2. Golovchinsky, G. What the Query Told the Link: The Integration of Hypertext and Information Retrieval. In the Proceedings of Hypertext 97, Southampton UK, 1997.
3. Hecht, D. Embedded Data Glyph Technology for Hardcopy Digital Documents. Proceedings of the SPI Color Hard Copy and Graphic Arts III. SPIE, Feb, 1994.
4. Horvitz, E., Breese, J., Heckerman, D., Hovel, D., and Rommelse, K. The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users. In the Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI, July 1998, Morgan Kaufmann Publishers, pp. 256-265.
5. Horvitz, E. Principles of Mixed-Initiative User Interfaces. In the Proceedings of CHI99, Pittsburgh, Pennsylvania. ACM Press.
6. Lamming, M, and Flynn, M. "Forget-me-not" - Intimate Computing in Support of Human Memory. Technical Report EPC-1994-103. Proceedings of Friend21, '94 International Symposium on Next Generation Human Interface.
7. Lieberman, H. Autonomous Interface Agents. In the Proceedings of CHI97, Atlanta, Georgia. ACM Press.
8. Mimio. http://www.virtual-ink.com/
9. Mynatt, E. D., Back, M., Want, R., Baer, M., and Ellis, J. Designing Audio Aura. In the Proceedings of CHI98, Los Angeles, CA. ACM Press.
10. Rekimoto, J., and Nagao, K. The World through the Computer: Computer Augmented Interaction with Real World Environments Virtual and Augmented Realities. In the Proceedings of the ACM Symposium on User Interface Software and Technology – UIST'95, p. 29-36.
11. Margin Notes: Building a Contextually Aware Associative Memory (html), Bradley J. Rhodes, The Proceedings of the International Conference on Intelligent User Interfaces (IUI '00), New Orleans, LA, January 9-12, 2000.
12. Rhodes, B., and Starner, T. The Remembrance Agent: A Continuously Running Automated Information Retrieval System. In the Proceedings of The First International Conference on The Practical Application of Intelligent Agents and Multi Agent Technology (PAAM '96), London, UK, April 1996. pp. 487-495. http://rhodes.www.media.mit.edu/people/rhodes/research/Papers/remembrance.html
13. Salber, D., Dey, A., and Abowd, G. The Context Toolkit: Aiding the Development of Context-Enabled Applications. In the Proceedings of CHI99, Pittsburgh, Pennsylvania. ACM Press.
14. Schilit, W. N. A Context-Aware System Architecture for Mobile Distributed Computing. PhD Thesis, Columbia University, May 1995.
15. Schilit, W.N., Golovchinsky, G., and Price, M. Beyond Paper: Supporting Active Reading with Free Form Digital Ink Annotations. In the Proceedings of CHI '98, 1998. ACM Press.
16. Roy Want, Ken Fishkin, Beverly Harrison, Anuj Gujar. Bridging Real and Virtual Worlds with Electronic Tags. In the Proceedings of CHI99, Pittsburgh, Pennsylvania. ACM Press.

17. R.Want, A. Hopper, V. Falcao, J. Gibbons. The Active Badge Location System, ACM TOIS Vol. 10. N0. 1, Jan 1992 Pages 91-102
18. R.Want, B.Schilit, N. Adams, R. Gold, D. Goldberg, K. Petersen, J.Ellis, M. Weiser. An Overview of the Parctab Ubiquitous Computing Experiment. IEEE Personal Communications, December 1995, Vol 2. No.6, pp28-43