Chapter 1

Introduction

Understanding how users behave has been (and certainly *is*) a longstanding subject of study in a really wide range of disciplines in science. Often, behavior needs to be measured, usually by directly asking the users. When interacting with computers, though, the intention of the user is mostly hidden. What is more, direct user feedback is notoriously unreliable most of the time. For instance, feedback regarding feelings, opinions, threats, etc. is strongly biased toward an individual perception; and hence it is hardly generalizable.

Fortunately, despite of the heterogeneity and dynamism inherent in user behavior, some actions are common to many individuals, and hence they can be recognized automatically. This kind of information can provide useful hints when designing interactive systems, which is the foremost motivation of this thesis, as discussed in this chapter.

Chapter Outline

1.1 Preamble: On User Behavior

Behavior refers to the actions or reactions of an object or organism, usually in relation to the environment. Behavior can be (sub)conscious, (c)overt, and (in)voluntary. In Human-Computer Interaction (HCI), behavior is the collection of responses exhibited by people, which are influenced by a diversity of factors; e.g., culture, attitudes, emotions, values, and/or genetics.

According to humanism, each individual has a different behavior. Observations about individual differences can thus inform the design of interfaces that are tailored to suit specific needs [Hwang et al., 2004]. Nevertheless, humans often show certain behaviors recurrently. In fact, some actions can be recognized automatically and therefore can provide useful hints when designing interactive systems. For example, when browsing a web page, if many users highlight the same text paragraph and copy it, then that text is supposed to be interesting, and hence the webmaster could consider giving it more prominence, e.g., by typesetting it in boldface.

Additionally, user behavior is not static but rather *dynamic* per se: preferences and attitudes change frequently over time. This fact can easily invalidate methods or theories that were developed not so many time ago, because of the temporary dependence of the evaluations that once supported them—for instance, think of the findings on electronic mail usage analysis reported thirty years ago by Hersh [1982]. Instead, measuring natural behavior gives a much more accurate picture of a user's immediate experience rather than asking him after a task is complete [Hernandez, 2007]. This way, behavioral (or biometric or interaction-based) measurements are theoretically more accurate than relying on explicit user feedback. They are indeed theoretically more accurate because, similar to everyday life body language, a certain behavior does not indicate always and universally the same inner state [Gellner et al., 2004]. So, depending on the task or its context, we can safely rely on this kind of measures or, on the contrary, acknowledge their limitations and combine them with other data sources.

1.1.1 Historical Background

According to behaviorism, behavior can be studied in a systematic and observable manner with no consideration of internal mental states [Cherry, 2006]. So, intentions are evidenced by exertions: users first focus and then execute actions. But, can behavior be measured? If not, then it could not be scientifically analyzed. Fortunately, this is not the case. In fact, instrumentation, i.e., automatic recording of user behavior within a system, has a long history in psychology. Its use in simple systems such as operant chambers (c.f. the Skinner box) helped to advance the study of animal (and, later, human) learning, revealing new patterns of behavior. Instrumentation was a key milestone in

HCI, since the field draws on cognitive psychology at its theoretical base. Over the last 25 years researchers have used instrumentation to better understand users and, consequently, to improve applications [Kim et al., 2008]. Computers are now found in most aspects of our daily life, and for some it is hard to even imagine a world without them.

Today, user interfaces (UIs) are one of the main value-added competitive advantages of computer applications, as both hardware and basic software become commodities. People no longer are willing to accept products with poorly designed UIs. So much so that notions of software products have been revisited with generalized psychology and physiology concepts in mind. For example, the standard ISO/TR 16982:2002 addresses technical issues related to human factors and ergonomics, to the extent necessary to allow managers to understand their relevance and importance in the design process as a whole.

Interaction design is often associated with the design of UIs in a variety of media, but focuses on the aspects of the interface that define and present its behavior over time, with a focus on developing the system to respond to the user experience and not the other way around. Designing interactive systems is about designing technology to maximize aspects of the interaction toward some goal [Bongard, 2010]. Interactivity, however, is not limited to technological systems. People have been interacting with each other as long as humans have been a species [Sinclair, 2011]. Therefore, interaction design can be applied to the development of any software solution, such as services and events. Ultimately, the design process must balance technical functionality and aesthetics to create a system that is not only operational but also usable and adaptable to changing user needs. Therefore, it is necessary to consider a multidisciplinary point of view to understand the role of human beings in computer science.

Finally, to close this very succinct historical context[1], we should mention the contributions to HCI of notable organizations such as the Interaction Design Foundation and ACM SIGCHI in USA or AIPO in Spain. Organizations like these are providing an international discussion forum through conferences, publications, workshops, courses and tutorials, websites, email discussion groups, and other services. For many of us, HCI is therefore enjoying a privileged position compared to other fields in computer science.

1.2 Implicit Interaction

Often, in HCI, behavior needs to be measured. Otherwise, how could we figure out if an application is really being used as intended? It is clear that user feedback is invaluable and, as such, usually behavioral data are gathered by directly asking the users. When interacting with computers, though, the intention of the user is mostly hidden [Hofgesang, 2006]. The activation of automatic goals,

[1] [Carroll, 2009] is a must-read in this regard.

and the physical traits of stimuli in our environment all influence our thoughts and behavior considerably, and often without our awareness.

What is more, direct user feedback is notoriously unreliable most of the time. For instance, feedback regarding feelings, opinions, threats, etc. is strongly biased toward an individual perception; and hence it is hardly generalizable—unless the size of the user sample is fairly substantial, of course, which is rarely the case in HCI studies (see, e.g., [Henze, 2011] for a quantitative comparison). Moreover, this kind of feedback must be acquired through some in-lab based methods, e.g., surveys, usability tests, cognitive walkthroughs, etc., and therefore requires to invest both time and money, which are often finite resources that eventually should be optimized.

In addition, to learn a user's interests reliably, intelligent systems need a significant amount of training data from the user. The cost of obtaining such training data is often prohibitive because the user must directly label each training instance, and few users are willing to do so [Goecks and Shavlik, 2000; Zigoris and Zhang, 2006]. Meanwhile, users expect a system to work reasonably well as soon as they first use the system. Thus, it is supposed that systems should work well initially with less (or none) explicit user feedback.

The social psychologist John A. Barg (1955–) stated that *one of the functions of consciousness is to select behaviors that can be automated and become unconscious.* In this context, researchers have elucidated new ways of expanding this notion to computers. As such, many different definitions (that largely overlap each other) have been independently proposed worldwide and thus are diffusely spread in the literature. For instance, implicit interaction is related to some extent to the following terms:

- Ubiquitous Computing [Weiser, 1993]
- Calm Technology [Weiser and Brown, 1996]
- Proactive Computing [Tennenhouse, 2000]
- Ambient Intelligence [Hansmann, 2003]
- Attentive Interface [Vertegaal, 2003]
- Perceptual Interface [Wilson and Oliver, 2005]

In the literature, implicit interaction is found to be cited, among others, as:

- Untold Feedback [Tan and Teo, 1998]
- Subsymbolic Behavior [Hofmann et al., 2006]
- Subconscious Awareness [Yoneki, 2006]
- Passive Actions [Grimes et al., 2007]
- Implicit Intentions [Kitayama et al., 2008]

Consequently, as pointed out by Oulasvirta and Salovaara [2004], the topic now seems to be in a state of conceptual balkanization, and it is difficult to get an overall grasp of the field. This fact poses an additional difficulty when defining the topic precisely. From my research, however, I would probably recommend (as being most adequate) the definition of Schmidt [2000]:

> *An action performed by the user that is not primarily aimed to interact with a computerized system but which such a system understands as input.*

Implicit interactions are thus those actions that the user performs with little (or no) awareness. And, unsurprisingly, humans have an abundance of experience with implicit interactions; we successfully employ them in a daily basis without conscious thought. For example, we laugh when someone tells a joke that we like. In doing so, we are communicating to that person that we appreciate such a joke. Humans constantly exchange information about their environment, and so can do computers. Figure 1.1 depicts a framework that summarizes quite well a modern view of implicit interactions in HCI.

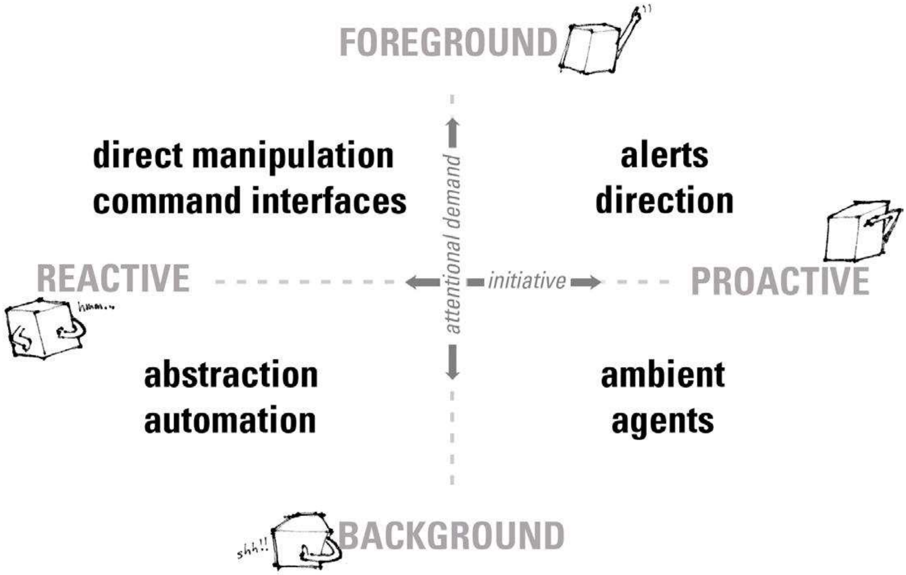

Figure 1.1: The implicit interaction framework [Ju and Leifer, 2008]. © Massachusetts Institute of Technology. Reproduced with permission.

As previously pointed out, the concept of implicit interaction is somewhat historically related to the ubiquitous computing (et al.) mantra: "the most profound technologies are those that disappear" [Weiser, 1999]. However, implicit interaction has a subtle but fundamental differentiation factor: is the user who takes the initiative to interact with the system. Therefore, ultimately the role of implicit interaction consist in leveraging as much information as possible derived from a natural user input, without requiring the user to be aware of the data the system needs to operate. This definitely has the capacity to make computers more useful and tailored to our needs.

1.2.1 Putting It All Together

The increasing use of technology—especially concerning to mobile devices and the Web—is changing our daily lives, not only in the way we communicate with each other or share information, but also *how* we relate to the environment. This entails new opportunities to transfer knowledge from one domain to another, by understanding that: *a*) implicit interactions offer a valuable source of information, and *b*) they can help to better manage user expectations.

By unobtrusively observing the user behavior we are able to learn functions of value. We can collect automatically generated training samples during a normal use, allowing for a collection of large datasets if deployed over the Web. This is interesting for many reasons. First, typical interactions with an application can involve many impasses, depending on the expertise of the user toward the application. Second, if such an application is intended to be used by an unknown user population, then it is very likely to involve ill-structured goals and tasks, and substantial influences from the content that is encountered while interacting [Card et al., 2001]. Third, classical approaches have relied on very simple measures such as time spent on a task or average number of clicks alone. These measures do not, however, provide any trace of the moment-by-moment cognition that occurs between regular interactions. If we are interested in developing detailed models of such cognition—for instance, to better understand how people's goals evolve, how people perceive and process the contents of an application, how and why they make decisions, and so on—then progress will be accelerated by having more detailed data of that cognition [Card et al., 2001].

Implicit interaction, as observed, requires no training and provides context for actions. As such, a wise knowledge of the limits, capabilities, and potential of implicit interaction in HCI provides an interesting theoretical basis for a systematic approach to analyzing, optimizing, and enhancing computer applications.

1.3 Aims and Goals of the Thesis

The central hypothesis of this research work is that *1*) there is a lot of information inherently encoded in user interactions, which *2*) can be measured and from which it is possible to extract meaningful knowledge, and therefore *3*) can be leveraged in a wide spectrum of applications and tasks. Virtually every chapter of the thesis is devoted to this notion, aiming to answer the same question: *How can implicit interaction be of help in computing systems?*

Other questions we try to answer include the following[2]. How can we exploit the potential of computer-based support to augment our daily activities? How

[2] See also `http://www.ercim.eu/EU-NSF/DC.pdf`

can we build systems in the face of uncertainty and partial knowledge? When do we try to predict the user and when do we let the user choose? How do we convey the system boundaries to the user?

This thesis is approached with a double-fold intent: *a*) researching on what characteristics can be inferred or leveraged from how users behave when interacting with computers, and *b*) deriving applications and implications to improve the utility of the systems that are meant to be used by people in a regular basis. There is a challenge, thus, in the way we can exploit this potential, in order to rethink how current technology may drive the dynamic environment of interactive systems. Through an exploratory research well beyond the classical (now interdisciplinary[3]) scope of HCI, this thesis will try to expand the body of knowledge on implicit interaction to related communities that rely to some extent on the user intervention, such as Cognitive Science, Infographics, Interactive Pattern Recognition, or Visual Design communities. This way, by exploring the role of implicit interactions in different domains and from different perspectives, not only a global vision of their importance is acquired; but specific solutions and working perspectives are proposed, discussed, and evaluated at different levels of understanding, depending on the specific task and the available resources. To do so, every chapter of this thesis has been conceived as a self-contained unit that in turn relates to the central topic of the thesis: the role of implicit interaction in HCI.

1.3.1 Organization and Contributions

This work has been divided into five illustrative scenarios, each one corresponding to a main chapter of this thesis, which are indeed the main contributions of the author to the field of implicit interaction. A brief overview of them is now advanced, although the reader can find a more detailed description in 'Thesis Overview' on page 9.

Chapter 2 showcases what probably is the most direct application to begin dealing with implicit interactions: visualization. An open source tool to understand browsing behavior is thoroughly described, providing also a real-world case study as an evidence of its utility. Most parts of this tool have been used to build other systems that helped to achieve the goals of this thesis. Chapter 3 presents a methodology designed to model the user in context, i.e., to find homogeneous groups of what a priori are different interaction behaviors, and also to automatically identify outliers. In addition, a novel revisitation of the K-means algorithm is presented to classify human actions in an unsupervised way. Chapter 4 discusses the problems when the focus of interaction changes from application to application, either unconsciously (e.g., a pop-up notification) or on purpose (e.g., multitasking). A technique to regain context is introduced in the domain of parallel browsing, and some directions are

[3] According to A. Oulasvirta, HCI has become so absurdly diverse and multi-multi-disciplinary that it is more aptly called *hyper-disciplinary*.

given to extend the same notion to mobile and desktop applications. Chapter 5 provides a novel approach to automatically redesign interface widgets. An appealing feature of such approach is that the method operates unobtrusively for both the user and the application structure. Although this is still ongoing work, with about a year of existence, the motivation of the technique has been empirically validated. Chapter 6 discusses the role of implicit interactions in Interactive Pattern Recognition applications, where the system and the user are expected to collaborate seamlessly. Four applications are examined: handwriting transcription, machine translation, grammatical parsing, and image retrieval. Finally, Chapter 7 wraps up the general conclusions of the thesis, remarking the main implications for design when implicit interaction is considered, and stating possible directions for further research. Last but not least, Appendix A enumerates the publications derived from this thesis.

1.3.2 Importance and Application Fields

Software applications in general and interactive systems in particular imply somewhat the understanding of their users. As previously discussed in Section 1.2, virtually any user-driven system can gain some benefit from implicit interaction analysis. Just to name a few of the possible application fields:

Usabiliy Testing Both remote and in-lab usability experiments are the primary source to evaluate the success of computer applications. Here, implicit interaction can help to unobtrusively analyze natural behaviors.

Data Mining If the experiments depicted above are, e.g., deployed over the Web, one can obtain vast quantities of data samples and perform readily prospective studies.

Performance Evaluation Related to the previous examples, a baseline control sample could be compared to a variety of test samples in real time, without interfering with the user experience.

Interface Analysis Determine which elements in the layout do attract the user interaction the most; again, without asking the users on purpose.

Gesture Recognition Use implicit features to convey meaning when drawing a picture (e.g., identify symmetries) or when handwriting (automatically isolate words or characters).

Usage Elicitation On the Web, spider bots behavior may greatly distort human usage patterns, hence it is critical to deal only with interaction data from real users.

Interaction Research Understanding human movement is a key factor to improve input devices as well as envision novel interaction techniques.

Behavior Prediction Usage data can presage not only how interfaces are likely to be used, but also which elements add value (or not) to the application.

Information Visualization Visualizing what users have done is a great aid to understand exactly how users behave and perform actions.

Biometrics Model behavior according to the usage of mouse, keyboard, eye-gaze, or other input devices for identifying users unequivocally.

Collaborative Filtering Discover usage profiles, involving the collaboration among multiple methods, viewpoints, data sources, and so on.

User Modeling Acquire information about a user (or a group of users) so as to be able to adapt their behavior to that user (or that group).

Multimodal Interfaces Leverage additional feedback signals that sometimes are unconsciously submitted to improve the utility of the system.

Self-Adapting UIs Use interaction data for re-arranging layout elements based on how users interact with them.

1.4 Thesis Overview

The following sections below introduce the contents that shall be later covered in the chapters of the thesis. It is worth mentioning that all systems developed in the context of this thesis are either web-based or closely related to the Web. The main reason is because currently people use web browsers more than any other class of desktop software on a daily basis. This situation has created a previously unparalleled level of user experience in a software niche [Edmonds, 2003]. Moreover, regarding to test new research methods and techniques, three reasons back up the need for driving research through web-based systems: *1*) the initial development time can be shorter, so the system is available to users earlier, *2*) continuous improvement is possible, without having to update or reinstall software, and *3*) real-world usage data can be obtained during the application life cycle.

1.4.1 Interactive Usability Evaluation

Besides conventional features such as performance and robustness, usability is now recognized as an important quality attribute in software development. Traditionally, usability is investigated in controlled laboratory conditions, by recruiting a (hopefully representative) user sample and often performing video recordings and surveys that are later reviewed. This requires an important investment in time and money, not to mention that processing user interaction data is, at a minimum, cumbersome. This chapter discusses the role of implicit interaction when performing usability tests on websites; concretely, *a)* which kind of data can be gathered by observing the overt behavior of users, without relying on explicit feedback, *b)* how this data can be presented to the usability evaluator, and *c)* which questions can be answered by inspecting such data.

1.4.2 Behavioral Clustering

Behavioral clustering is a broad term that refers to the task of automatically labeling and classifying user behavior. Overall, clustering is a relevant method to identify sub-populations in a dataset, so that they can be represented by more compact structures for, e.g., classification and retrieval purposes. To this end, implicit interaction can provide current clustering methods with additional information. First, on the Web, fine-grained interactions can reveal valuable information (e.g., related to cursor movements, hesitations, etc.) that is not available in typical access logs. Second, in a general context, user behavior has an intrinsic *sequential* nature, which is not considered on current clustering analysis, that can be exploited to simplify the structure of the data. This chapter proposes two approaches to solve both drawbacks: *1*) a novel methodology to model websites, i.e., finding interaction profiles according to how users behave while browsing, and *2*) a novel clustering algorithm to deal with sequentially distributed data, whose suitability is illustrated in a human action recognition task.

1.4.3 Human Multitasking

We use different applications to multi-task the activities we do every day, even when browsing the Web; e.g. it is not unusual having multiple tabs or browser instances open at a time. People thus may cognitively coordinate simultaneous tasks through multiple windows or multi-tabbing, having many applications open at the same time and switching between them in any order. This chapter addresses how to reduce the overall cognitive load involved in switching among multiple windows during the course of typical information work. The chapter provides directions for designing mobile applications, where interrupted tasks usually have a high resumption cost. A method was implemented to illustrate a means to assist web browsing: using mouse movements as an indicator of attention, a browser plugin highlights the most recently interacted item as well as displaying (part of) the mouse path. An empirical study shows that this technique can help the user to resume and complete browsing tasks more quickly.

1.4.4 Adaptive User Interfaces

Adaptive systems accommodate the UI to the user, but doing so automatically is a non-trivial problem. Adaptation should be predictable, transparent, and discreet, so that changes introduced to the UI do not confuse the user. Also, adaptation should not interfere with the structure of the application. This chapter presents a general framework to restyle UI widgets, in order to adapt them to the user behavior. The value of this methodology comes from the fact that it is suited to any application language or toolkit supporting structured data hierarchies and style sheets. As discussed, an explicit end user intervention is not required, and changes are gradually applied so that they are not intrusive for the user. The method is also extended as a technique to foster creativity, by suggesting redesign examples to the UI developer.

1.4.5 Interactive Pattern Recognition

Mining implicit data from user interactions provides research with a series of interesting opportunities in order to create technology that adapts to the dynamic environment of interactive systems. This chapter presents an iterative process to produce a user-desired result, in which the system initially proposes an automatic output, which is partially corrected by the user, which the system then uses to suggest a suitable hypothesis. Such iterative (and interactive and predictive) paradigm is the core of the MIPRCV project, a Spanish consortium of 10 universities and 7 research groups, which the author has been involved with since 2009. The main contribution of the author to the project has been the development (and later evaluation with real users) of interactive systems that implement the aforementioned paradigm, namely: *1*) Interactive Handwritten Transcription, *2*) Interactive Machine Translation, *3*) Interactive Grammatical Parsing, and *4*) Interactive Image Retrieval. According to user-simulated experiments and a series of real-world evaluations[4], results suggest that this paradigm can substantially reduce the human effort needed to produce a high-quality output.

Bibliography of Chapter 1

J. Bongard. Class notes on Human Computer Interaction. Available at http://cs.uvm.edu/~jbongard/2012_HCI/CS228_Class_02.pdf, 2010. Retrieved July 12, 2012.

S. K. Card, P. L. Pirolli, M. V. D. Wege, J. B. Morrison, R. W. Reeder, P. K. Schraedley, and J. Boshart. Information scent as a driver of web behavior graphs: Results of a protocol analysis method for web usability. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI)*, pp. 498–505, 2001.

J. M. Carroll. *Encyclopedia of Human-Computer Interaction*, chap. Human Computer Interaction (HCI). The Interaction Design Foundation, 2009.

K. Cherry. What is behaviorism? Available at http://psychology.about.com/od/behavioralpsychology/f/behaviorism.htm, 2006. Retrieved July 27, 2012.

A. Edmonds. Uzilla: A new tool for web usability testing. *Behavior Research Methods, Instruments, & Computers*, 35(2):194–201, 2003.

[4]Excepting Grammatical Parsing, all prototypes were empirically tested with real users.

M. Gellner, P. Forbrig, and M. Nelius. Results of mousemap-based usability evaluations – towards automating analyses of behavioral aspects. In *Proceedings of the 8th ERCIM Workshop UI4ALL: "User interfaces for all"*, 2004.

J. Goecks and J. Shavlik. Learning users' interests by unobtrusively observing their normal behavior. In *Proceedings of the 5th International Conference on Intelligent User Interfaces (IUI)*, pp. 129–132, 2000.

C. Grimes, D. Tang, and D. M. Russell. Query logs alone are not enough. In *Workshop on Query Log Analysis at the 18th International Conference on World Wide Web (WWW)*, 2007.

U. Hansmann. *Pervasive Computing: The Mobile World*. Springer, 2nd edition, 2003.

N. Henze. Analysis of user studies at MobileHCI 2011. Available at http://nhenze.net/?p=865, 2011. Retrieved October 2, 2011.

T. Hernandez. But what does it all mean? understanding eye-tracking results. Available at http://eyetools.com/articles, 2007. Retrieved November 8, 2009.

H. M. Hersh. Electronic mail usage analysis. In *Proceedings of the 1982 conference on Human Factors in Computing Systems (CHI)*, pp. 278–280, 1982.

P. I. Hofgesang. Methodology for preprocessing and evaluating the time spent on web pages. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pp. 218–225, 2006.

K. Hofmann, C. Reed, and H. Holz. Unobtrusive data collection for web-based social navigation. In *Proceedings of the 4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, 2006.

F. Hwang, S. Keates, P. Langdon, and J. Clarkson. Mouse movements of motion-impaired users: A submovement analysis. In *Proceedings of the 6th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS)*, pp. 102–109, 2004.

W. Ju and L. Leifer. The design of implicit interactions: Making interactive systems less obnoxious. *Design Issues*, 24(3):72–84, 2008.

J. H. Kim, D. V. Gunn, E. Schuh, B. C. Phillips, R. J. Pagulayan, and D. Wixon. Tracking real-time user experience (TRUE): A comprehensive instrumentation solution for complex systems. In *Proceedings of the 26th annual SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pp. 443–452, 2008.

D. Kitayama, T. Teratani, and K. Sumiya. Digital map restructuring method based on implicit intentions extracted from users' operations. In *Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication (ICUIMC)*, pp. 45–53, 2008.

A. Oulasvirta and A. Salovaara. A cognitive meta-analysis of design approaches to interruptions in intelligent environments. In *Proceedings of Extended Abstracts on Human factors in computing systems (CHI EA)*, pp. 1155–1158, 2004.

A. Schmidt. Implicit human-computer interaction through context. *Personal and Ubiquitous Computing*, 4(2):191–199, 2000.

K. Sinclair. Creating interactions in building automation. Available at http://www.automatedbuildings.com/news/aug11/columns/110725014808emc.html, 2011. Retrieved October 2, 2011.

A.-H. Tan and C. Teo. Learning user profiles for personalized information dissemination. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pp. 183–188, 1998.

D. Tennenhouse. Proactive computing. *Communications of the ACM*, 43(5):43–50, 2000.

R. Vertegaal. Attentive user interfaces. *Communications of the ACM*, 46(3):31–33, 2003. Editorial note.

M. Weiser. Some computer science issues in ubiquitous computing. *Communications of the ACM*, 36(7):74–84, 1993.

M. Weiser. The computer for the 21st century. *Mobile Computing and Communications Review*, 3(3):3–11, 1999.

M. Weiser and J. S. Brown. The coming age of calm technology, 1996.

A. Wilson and N. Oliver. Multimodal sensing for explicit and implicit interaction. In *Proceedings of the 11th International Conference on Human-Computer Interaction (HCII)*, 2005.

E. Yoneki. Sentient future competition: Ambient intelligence by collaborative eye tracking. In *Proceedings of the European Workshop on Wireless Sensor Networks (EWSN)*, 2006.

P. Zigoris and Y. Zhang. Bayesian adaptive user profiling with explicit & implicit feedback. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 397–404, 2006.