J.B. Morris

Dr. John Dionisio

CMSI 370

21 October 2014

Abstract

Ubiquitous devices are meant to bring the intelligence of computing devices to items used in our everyday lives. Utilizing their numbers, these artifacts can gather massive amounts of information and communicate with users in a multitude of different levels. However, this represents a huge shift from the traditional methods of user interaction and requires new methods and innovative approaches to make them practical. This paper addresses multiple scholarly articles to discern the guiding principles of ubiquitous computing and the challenges that they create.

Introduction

Ubiquitous computing utilizes the increasing utility derived from smaller and more efficient devices to create augmented environments that individuals can interact with in order to make everyday tasks easier and more completely integrated with a computerized system. One crucial aspect of ubiquitous devices is to create an "Internet of things" that includes elements such as appliances, lighting, security systems, wearable technology, and so on. These devices would be intelligent: remembering the behavioral patterns of the users, reminding users when tasks need to be completed, and recognizing their current state in a given environment. Implementing technology of this nature requires a new and different approach to human-computer interaction that would utilize an array of sensors to detect input from the user through mediums such as audio, motion, infrared, gesture, or other unconventional mediums. The challenge presented to ubiquitous computing is to interact with the user in such a way that not only feels natural and unobtrusive but does so in an implicit manner without the complete reliance on screens and displays. In this paper, I will analyze the paradigm shift in the implementation of user interfaces in Ubiquitous computing and the methodologies that are used in their creation.

Prior Work

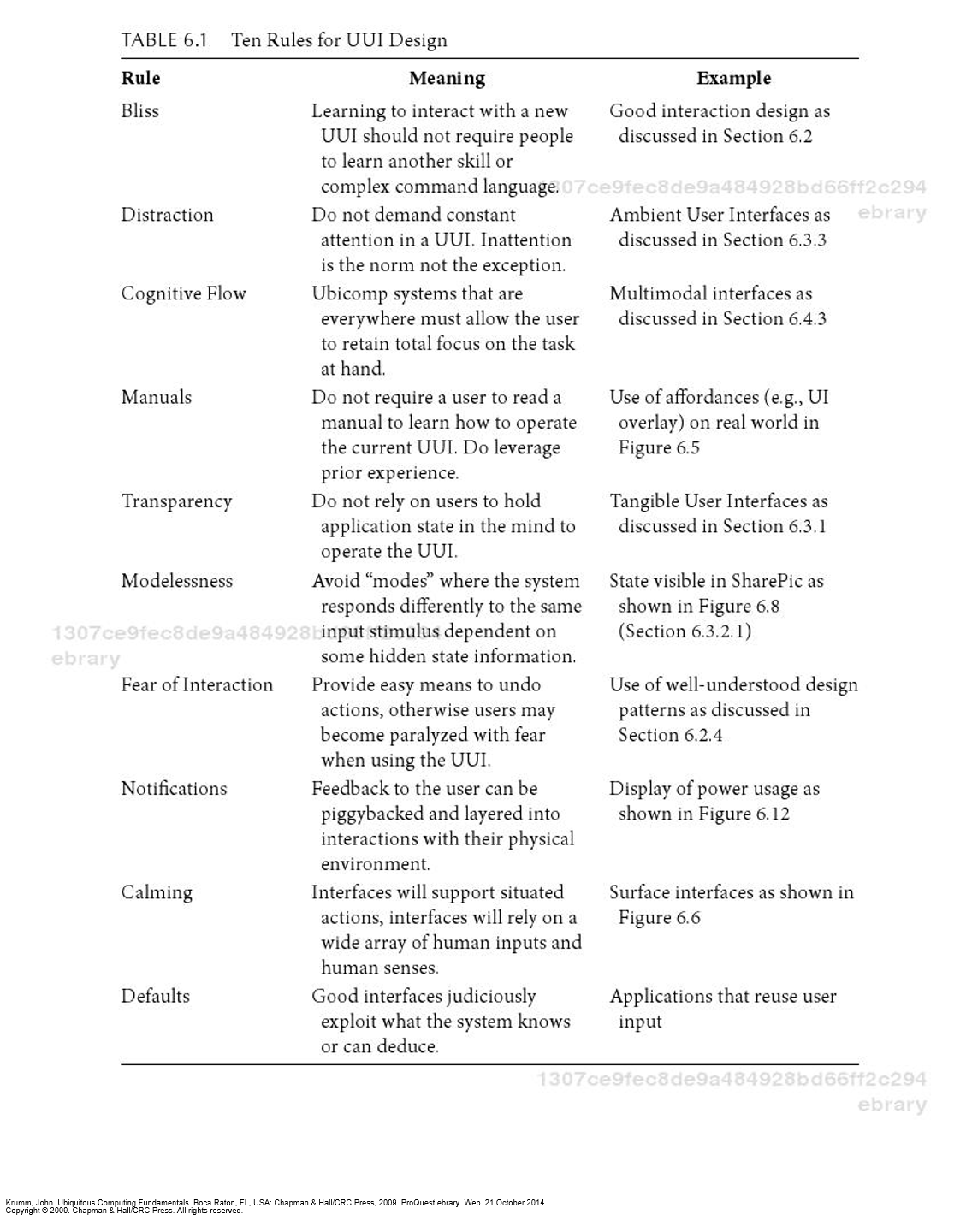
*Ubiquitous Computing Fundamentals* by John Krummsets standards for a ubiquitous computing user interface with a framework of required elements as well as a diverse set of general methods one could use to implement such an interface. Krumm focuses on the need for the system to remain in the peripheral. Ubiquitous computing must take place in as many aspects of the environment as possible with as little disturbance to the users in that environment as possible.

*Pervasive Systems and Ubiquitous Computing* by Alessandro Genco, however, gives more attention to the methods of communicating to the user. Genco introduces sensor equipment such as data gloves, cameras, and other wearables to allow the system to gather as much data from the user as possible. In addition, Genco's system utilizes the user's emotional state as a determining factor in its processes. Ubiquitous computing in this way relies on constant awareness of the user and interpreting human behavior.

Methods

The shift in the user interface from a graphic user interface (GUI) to a ubicomp user interface (UUI) presents one of the largest challenges to ubiquitous computing. The goal for a truly streamlined ubicomp system would be to have it integrated seamlessly and naturally into the environment. It is important to consider the dramatic change in human-computer interaction (HCI) that is brought about by, not only the distancing from a graphic user interface, but the nature of the new computing system in general. Krumm highlights the importance of designing an overall experience as opposed to the individual subsystems that make up the environment. This experience must be "calm and invisible" and "should remain unnoticed until required" according to him (Krumm 240) . Ubicomp user interfaces, therefore, must utilize a wide range of inputs such as spatial information, identity, user models, temporal data, environmental data, social data, resource availability, computing data, activity, schedules and agendas, and other data that can be discerned explicitly or implicitly (243).

Explicit data collection allows users to directly manipulate the system through whatever medium is available. In a classic graphic user interface, the most common methods of explicit data collection are achieved through the keyboard and mouse. Other common applications include touch screens, remotes, and styluses. With these tools, a user can issue commands and provide inputs to the system whenever necessary. In a ubicomp user interface, users may provide explicit input through speech, gaze, or any form of human movement. Implicit data collection, however, deals more with the contextual awareness of any artifact within the system or the system as a whole. To accomplish this, a ubiquitous system can gather information from persons, the sensed environment, and the computational environment (243). A series of sensors can be placed in an environment or on the user to gather information about the state of the user or the state of the artifacts or other objects within the environment.

Krumm presents a series of criteria for ubicomp user interfaces in the table below. 

Directly addressing interaction design principals in ubicomp user interfaces, Krumm addresses user centered design (UCD), systems design, and genius design. User centered design focuses on the user's desires and needs while involving him or her in every step of the computing process. Systems design utilizes the combination of components and their interaction to solve a problem. This approach requires a designer to consider the entirety of the environment the system will be implemented in, and not simply the components and artifacts. Genius design relies completely on proven designers to create systems that reflect their design skill.

The implementation of ubicomp user interfaces is analyzed more closely in *Pervasive Systems and Ubiquitous Computing*. Alessandro Genco highlights the importance of designing ubicomp user interfaces that take advantage of explicit and implicit user input simultaneously. Much like a conversation, messages are conveyed through a number of different methods and interpreted as such. To effectively create intelligent environments, they must be intelligent enough to interpret users' inputs on a variety of levels while being context aware. This introduces a completely new dimension to user input and, much like conversation, leaves more room for potential misinterpretation.

With regards to design principles, Genco presents learnability, flexibility, and robustness as general principles for modern interfaces in general. More specifically, he details gerneralizability—a subset of learnability meaning the ability for an interface to be accessible to a user based on similar situations he or she may have encountered—and recoverability—a subset of robustness that refers to an interface's ability to allow a user to easily recover from errors. With the addition of ubiquitous computing, adaptive aspects are added to the programs guidelines. The interface itself must adapt to the user based on his or her behavior derived from three modules: perceptual processing, behavioral processing, and cognitive comprehension (Genco 21). The goal of this model is to acquire information about the user's mood and mental state through things like facial expressions, posture, and other perceivable indicators to effectively adapt to them.

Discussion

The major shift in design principals in ubiquitous computing is derived from its very nature. The shift from a desktop centered interface—and graphic user interface in general—creates a need for a different means of input and communication. Graphic user interfaces exist under the complete control of the computer that creates the user environment while ubiquitous computing takes place in preexisting environments that vary greatly from one another. The user interface must, therefore, be very adaptable. Additionally, these means of human-computer interaction must also be as unobtrusive as possible, with computing elements and artifacts placed throughout an environment, one must avoid overwhelming a user. Passive means of communicating with a system allow for interaction without any investment on the part of the user, allowing one to focus on a particular task as the system addresses other concerns in the background. Passive communication, however, relies on inferences and could lead to greater numbers of errors when it comes to using them as user input.

Above all, however, ubiquitous computing must be massively intelligent for it to function properly. In order to interact with users indirectly, it requires the ability to interpret human behavior accurately. Doing so presents the challenge of articulating a series of sensors to record a user's behavior as well as creating a means of interpreting that recorded data. As opposed to graphic user interfaces which interpret very direct commands, a ubiquitous user interface must draw from inferences. This introduces an interesting element to the idea of mental models as it frames things such as smart houses and other intelligent environments as entities that think on their own for you rather than responding to you.

Conclusion

Ubiquitous computing distances individuals from the familiar desktop based interfaces and attempts to add the convenience of computing to real world environments. The dramatic shift in environment—that is, a virtual one to a physical one—forces developers to form radically different ways of interacting with the interfaces. The demands of a system like this are high as the system itself must be intelligent, aware, familiar, and unobtrusive at the same time. Essentially, the goal is to create a computer that is always around when one needs it and invisible when one doesn't while being intelligent enough to know when those times are. The intellect, in my opinion, represents the greatest challenge. Users can slowly adapt to ubiquitous environments as more and more technology begins to link to mobile devices and tablets, but stripping the graphic interface and central computing environment requires the system to think for the user. Without an extreme level of intellect supported by a massive amount of sensory technology, ubiquitous computing user interfaces that fit the aforementioned criteria will have to make sacrifices in their guidelines in order to function.

Works Cited

Genco, Alessandro, and Sorce, Salvatore. Pervasive Systems and Ubiquitous Computing. Ashurst, GBR: WIT Press, 2010. ProQuest ebrary. Web. 21 October 2014.

Goodwin, Steven. *Smart Home Automation with Linux*. New York: Apress, 2010. 1-303. Print.

Krumm, John. Ubiquitous Computing Fundamentals. Boca Raton, FL, USA: Chapman & Hall/CRC Press, 2009. ProQuest ebrary. Web. 21 October 2014.

Poslad, Stefan. Ubiquitous Computing : Smart Devices, Environments and Interactions. Hoboken, NJ, USA: John Wiley & Sons, 2009. ProQuest ebrary. Web. 21 October 2014.

Scholtz, Jean, and Sunny Consolvo. "Towards a Discipline for Evaluating Ubiquitous Computing Applications." (2004): 1-18. Intel Research. Web. 1 Oct. 2014. <http://intel-research.net/Publications/Seattle/022520041200\_232.pdf>.

Yang, Laurence Tianruo, Syukur, Evi, and Loke, Seng. Handbook on Mobile and Ubiquitous Computing. London, GBR: CRC Press, 2012. ProQuest ebrary. Web. 21 October 2014.