Essay - Ubiquitous computing

Kim Syversen

November 29, 2014

Introduction

Users should not be aware of that they are using computers. This is the essence of Mark Weisers vision of ubiquitous computing [13]. Computers exists in different forms and can be integrated into the physical objects we use in our daily life; they become augmented physical objects, and a device in ubiquitous computing. This does not mean that the device is perceived as having high usability, or that users becomes unaware of using a computer. Prototyping is a natural choice when exploring what is usable and what is not. To ease at least some of the parts of prototyping, it is possible to use model driven development and create models of common functionality which applies to a multitude of devices.

The requirements for usability in ubiquitous computing is different than they are for traditional computing. To enable comparison of evaluation results rigorously and quantitatively, there is need for a framework. A framework for evaluation help creating design guidelines.

This leads to the following thesis: *Usability in ubiquitous computing could be improved trough prototyping, supported by model-driven techniques and rigid evaluation.*

The first section discusses a study where it was used Smart-Its to develop augmented artifacts. It also discusses evaluation, and a possible reason for why there was no evaluation from the user's point of view. The second section discusses the development approach behind Smart-Its and explains why more devices should be available for prototyping. The third section discusses how model driven development might help creating user interfaces to similar, yet different devices. The fourth section discusses various challenges and possible technologies that could to some extent solve or mitigate prototyping, thus creating a lower threshold for prototyping and exploring possibilities.

Prototyping with Smart-Its and evaluation

The use of Smart-Its indicates that prototyping is easy. With Smart-Its, one can create embedded interactive systems by augmenting physical everyday objects in relatively little time [5]. There have been both case studies and developer workshops which indicates good results, both from the developer's perspective and easiness to develop rapid prototypes.

A weakness is that the user is not sufficiently included in the evaluation. None of the case studies or workshops seem to include any evaluation of usability aspects or how the prototypes was perceived by the user. One case study included users in a more direct situation, but the evaluation and results of the study is not mentioned [5]. The development of Smart-Its are conducted in two phases, where the first phase consists of creating the hardware and customizing software [5]. The second phase consists of implementing the details for the particular use. Both phases includes evaluation, but does not seem to include evaluation of the usability. There could be a reason that the users is not sufficiently included. According to [10], there is a lack of an accepted framework for user evaluation of ubiquitous computing applications that enables rigorously comparing and quantifying results. A framework is proposed, which hopes to help creating ubiquitous computing design guidelines. If the usability is to be improved, the evaluation must include the users.

Increasing the number of devices that can be prototyped

The flexibility of Smart-Its allows combining them with different everyday objects [5]. Designing hardware and software is relatively easy. Both the hardware and software is assembled by a modular principle. By integrating Smart-Its in everyday objects such as tables and using other devices such as ambient displays, they could form an interactive system. One scenario could be that a table can sense when the plate is laid down, indicating it is time for a meal, and the display in the room would then mute the sound and turn black while the plate is there. Together these form a smart space which could be explored together in a more real context.

However, while the ideas are good, it is limited to experimenting with Smart-Its. The threshold for experimenting with other types of artifacts should be lowered. A study where ambient displays were integrated into the organizational environment, they had to use highly interdisciplinary approaches and employ designers and architects to develop these displays [12]. Another study focusing on proxemic interactions, used a self-built proximity toolkit [6]. By borrowing the idea of utilizing modules for both software and hardware, the number of possible artifacts for easier prototyping could be increased. It has been pointed out that researchers interested in ubiquitous computing often has a software background and is not interested in the hardware part [5]. If the hardware is not carefully considered, the computer will probably not disappear in the background. Studies such as [12, 6] has experimented with large artifacts that could be compared to boards and small artifacts that to some extent could be compared with tabs. If prototyping with other kinds of devices (such as tabs, pads, and boards mentioned in [13]), it should be possible to construct and extend them by a modular approach, as with Smart-Its. This mitigates the hassle with creating it from the ground and reduces development time. Thus, easing the threshold to experiment with them.

Model Driven Development and mobile computing systems

Ubiquitous computing includes mobile devices with different types of user interfaces. Technology that previously existed in other devices is becoming available on mobile devices, and they are becoming more advanced in terms of hardware and software. This enables creating new types of devices, and these has different user interfaces and requirements. The different devices might accept input in different ways and have different forms.

Model driven development might help the development and enable adjustments for supporting similar, yet different devices [8]. The advantages with model driven development is many, some of them are easier maintenance, and faster time to market. The idea is that on the conceptual level, the model can describe concepts that exists on all devices such as communication and interaction. On the logical level, the different concepts would be mapped together. In the terms of code, this could be skeleton code and implemented patterns. In the physical level, an implementation of the concepts is conducted to fit the specific device. Different restrictions for the devices can be set trough modelling rules. It is important that the model is not too general or too specific, else it cannot support the variety of devices [8].

According to [9]: "*pervasive computing offers new beginnings for the adventures and the restless – a rich open space where the rules have yet to be written and the borders yet have to be drawn*". Some years later [8] states that the most multichannel approaches do not use model driven development. It is pointed out in [8] that there is a need for user interface guidelines and standards for mobile solutions. This statement is supported in [10] where the a proposed framework hopes to help creating design guidelines.

It has been conducted studies of different aspects for mobile devices and it exists different projects with different goals [8]. One of them seems to be a more comprehensive platform for creating adaptive mobile interfaces. The development approach used in [12, 6, 5] indicates that the approach for creating user interfaces is very different.

Possibly in collaboration, it can be created good models which can be shared and utilized to faster develop. When discussing extending models, it helps to speak of the same model. It could lead to a preferred way to develop user interfaces for many similar, yet different devices when prototyping. Thus helping lowering the threshold.

Possible mitigations for various challenges in ubiquitous computing

A distributed system consists of devices which communicate and coordinate their actions to achieve a common goal. For a system to be more invisible and act as a whole, it should be distributed. According to [9], there are four key constraints related to mobility and distributed systems: "*unpredictable variation in network quality, lowered trust and robustness of mobile elements, limitations on local resources and a concern for battery power consumption*". These constraints are related to requirements for ubiquitous computing mentioned in [13], and includes both software and hardware challenges.

Challenges with creating distributed systems is mentioned in [9] (also challenges Smart-Its platform [12] as well). By utilizing semantic web, it could promote higher availability for distributed systems. This is because the semantic web will be decentralized [2]. However, it will require support by artificial intelligence.

Limitation on local resources and excessive battery power consumption could to some extent be mitigated by cyber foraging, as pointed out in [9]. Mobile devices can simply send their requests to hidden computers which is integrated in the environment and acts like surrogates. These computers receive requests, processes, and return results. Management of energy could to some extent be mitigated by an operating system which manages memory in a more conservative way [9]. The applications could adjust their resource usage based on available technology in the environment, which can be advertised. Advertisement could be realized with semantic web [2]. Since the environments can be very different, they may not be a need for all the functionality a mobile device can provide, and therefore it would be a good idea to advertise what functionality is available in the environment, so it is possible for the devices to save resources.

Mobile computing have challenges with location [3, 13]. Accurate positioning can be difficult by using the mobile network and GPS at places with weak signals, especially indoors. Alternative technologies that could be supplied for indoor positioning is mentioned in [4] (Bluetooth Service Discovery Protocol and eSquirt IR and RFID). They support different ranges and should be carefully considered before implemented.

Discovery challenges is mentioned in [4]. It is important that the process with registering devices in an environment should be automatic (it should be invisible for the user). When registered, there is then a challenge for how the devices to communicate and understand each other. Discovery and communication could be made more feasible with the use of techniques from semantic web [2]. Devices (agents) could send messages and understand each other by exchanging ontologies described with RDF. It is stated in [2] that RDF have turned out to be able to describe the majority of data machines is processing. This means that physical augmented objects, and other type of objects and their attributes could be described with RDF.

Sensing-challenges is discussed in [1]. A framework [10] is partly based on these. This framework is not complete, and will refined as it is used in practice. Two other studies which includes evaluation trough an acceptance model [7, 11] have some similarities with this framework because both includes aspects of trust, context and usefulness.

To in a larger degree realize ubiquitous computing, there are both software and hardware related challenges that must be solved [9, 5]. By mitigating challenges which hinders prototyping and exploring of different ways to integrate devices in the environment, prototyping could become easier. Thus it helps lower the threshold to create design guidelines which [8, 10] notes the importance of.

Conclusion

The meaning of this essay was to discuss challenges which keeps the threshold for prototyping higher than it should be, and possible ways to mitigate.

The thesis for the essay is: *Usability in ubiquitous computing could be improved trough prototyping, supported by model-driven techniques and rigid evaluation.* It is defended by suggesting direct or indirect ways to promote creation of design guidelines for ubiquitous computing, which helps increasing the usability.

The first section discusses Smart-Its, a platform which makes development easier. If the user was more included in the evaluation, it could promote creating design guidelines. Second section suggests that the number of devices that can easily be prototyped should be extended. This let both developers and users experiment with different kinds of devices, thus promoting creating design guidelines. Third section suggests how model driven development could be used to create user interfaces, promoting easier prototyping of user interfaces. The last section discusses various challenges that hinders the more practical usage of prototypes, and possible mitigations.

My personal conclusion is that while this essay only consider some challenges, it exists a lot of other challenges to realize ubiquitous computing fully. By gradually attacking challenges, I believe the threshold for prototyping can be lowered. It will probably not be done in a few years, because it requires much interdisciplinary work to create a system that covers all aspects of our human life. After all, when actually using computers, we should not be aware of that.

References

[1] Victoria Bellotti et al. *Making Sense of Sensing Systems: Five Questions for Designers and Researchers.* Apr. 2002.

[2] Tim Bernes-Lee, James Hendler, and Ora Lassila. *The Semantic Web.* May 2001.

[3] Gergely Biczók et al. *Navigating MazeMap: indoor human mobility, spatio-logical ties and future potential.* 2014.

[4] W. Keith Edwards. *Discovery Systems in Ubiquitous Computing.* 2006.

[5] Hans Gellersen et al. *Physical prototyping with Smart-Its.* 2004.

[6] Saul Greenberg et al. *Proxemic Interactions: The New Ubicomp?* 2011.

[7] John Krogstie and Anne-Cecilie Haugstvedt. *Mobile Augmented Reality for Cultural Heritage: A Technology Acceptance Study.* 2012.

[8] John Krogstie et al. *Research areas and challenges for mobile information systems.* 2004.

[9] M. Satyanarayanan. *Pervasive Computing: Vision and Challenges.* Aug. 2001.

[10] Jean Scholtz and Sunny Consolvo. *Toward a Framework for Evaluating Ubiquitous Computing Applications.* 2004.

[11] Keng Siau, Shang Gao, and John Krogstie. *Adoption of mobile information services: An empirical study.* 2014.

[12] Norbert A. Streitz et al. *Designing Smart Artifacts for Smart Environments.* Mar. 2005.

[13] Mark Weiser. *The Computer for the 21st Century*. Sept. 1991.