An Empirical Study on Immersive Prototyping Dimensions*

Samuel Moreira[1], Rui José[2,3], and José Creissac Campos[1,4]

[1] Departamento de Informática/Universidade do Minho
[2] Departamento de Sistemas de Informação/Universidade do Minho
[3] Centro Algoritmi
[4] HASLab / INESC TEC
pg17627@alunos.uminho.pt, rui@dsi.uminho.pt, jose.campos@di.uminho.pt

Abstract. Many aspects of the human experience of ubiquitous computing in built environments must be explored in the context of the target environment. However, delaying evaluation until a version of the system can be deployed can make redesign too costly. Prototypes have the potential to solve this problem by enabling evaluation before actual deployment. This paper presents a study of the design space of immersive prototyping for ubiquitous computing. It provides a framework to guide the alignment between specific evaluation goals and specific prototype properties. The goal is to understand the potential added-value of 3D simulation as a prototyping tool in the development process of ubiquitous computing environments.

Keywords: 3D environments, prototyping, ubiquitous computing.

1 Introduction

Ubiquitous computing technologies provide exciting new opportunities for enhancing physical spaces to support the needs and activities of people within them. However, many aspects of the human experience of ubiquitous computing in built environments can only be explored in the context of the target environment. In evaluating these systems it is not only necessary to explore conventional properties of usability, but also properties of the environment that contribute to the experience of its users. Fielding such systems for testing purposes, however, is in many cases not feasible because of the potential disruption to the target environment. Consider, for example, an emergency evacuation scenario. Additionally, developing the system to a deployable state can imply commitment to design decisions that will be expensive to reverse. Nevertheless, the potential impact of a system in user practice, justifies that its design should be explored as early as possible [18]. It should be possible to use prototypes to explore the

* This work is funded by the ERDF through Programme COMPETE and by the Portuguese Government through FCT - Foundation for Science and Technology, project ref. FCOMP-01-0124-FEDER-015095.

M. Kurosu (Ed.): Human-Computer Interaction, Part I, HCII 2013, LNCS 8004, pp. 421–430, 2013.

consequences that different design decisions might have, while promoting the identification of new solutions.

Simulated 3D environments offer an interesting solution to immersive prototyping [5,12,22,20]. 3D Application Servers and game engines provide a fast track to developing virtual worlds that replicate the type of environments that needs to be prototyped. The use of these 3D Application Servers as the basis for a immersive prototyping framework enables agile development of simulations of the ubiquitous environment. However, to be successful, immersive prototyping requires a thorough alignment with the key properties of the target environment, both at the technical and social level, and a strong focus on the specific evaluation goals and they can be met while considering the specific limitations of immersive prototyping. With this in mind, we have carried out a study of the design space of immersive prototyping based on 3D simulation, in order to define a framework to guide the alignment between specific evaluation goals and particular prototype properties.

A similar study was carried out by Ostkamp et al. [17]. In it the authors were interested in studies about public displays. They introduced the AR-Multipleye, a system that visually highlights items on a personal device that is pointed towards a public display. They, additionally, carried out an evaluation of the existing approaches according to a set of criteria to classify highlight methods for public displays.

In our case, the key issue that was addressed is "what are the relevant dimensions that prototypes should exhibit to better support evaluation of the envisaged design?". The paper presents the two groups of characteristics identified as a result of this work. The first relates to the immersive prototyping ubiquitous systems, and includes topics such as Fidelity of immersion, Embodied interaction support or Hybrid prototyping. The second addresses the different perspectives on evaluation (from evaluation centered on the system and its functional qualities, to evaluation centered on the user's experience of the system), and the methods to gather feedback about user experience. A discussion on how the framework was applied to the development of a prototype used to aid the design of a concrete ubiquitous environment ends the paper.

2 Methodology

In order to establish the relevant analysis dimensions we performed a review of the research literature on the topic of ubiquitous computing immersive prototyping. Most of the papers are related to the rapid development and evaluation of ubiquitous systems in the early stages of the development life cycle. Examples include 3DSim [12], TATUS [15], the work of O'Neill et al. [16], UBIWISE [1], the work of Reynolds [19] or APEX [20,21]. Others papers, as UbiWorld [5] and the work of Pushpendra et al. [22], are focused in creating immersive environments for users, and testing their applications, using CAVEs and other immersive technologies. VARU [6], CityCompiler [11], UbiREAL [14], and the work of Brandherm et al.[2], focus their study in hybrid prototyping approaches, integrating services (e.g. Internet services) and devices in their ubiquitous systems. A few papers are more concerned with the analysis of user behavior when confronted with different situations (this is the case of Siafu [10] and the work of Maly et al. [9]), while Topiary [7] and the work of Li et al. [8] are more concerned with the context awareness behavior of ubiquitous applications.

The papers were analyzed in search of codes for two groups of characteristics of ubiquitous computing that we initially defined as: (1) Properties of the simulation; and (2) Requirements for evaluation and evaluation objectives.

Open Coding [23] was used to analyze the contents of the papers. Each paper was read in order to identify phrases or paragraphs containing references to the two groups of characteristics of ubiquitous computing aforementioned. A code was assigned to each piece of text identified. At this stage, the goal was to generate as many codes as possible without much consideration of how they related with each other. The MAXQDA10 tool was used to aid the open coding process. A total of 33 different codes were identified: 20 in the first group, and 13 in the second. The number of code instances identified was 220.

An affinity diagram was then created to synthesize the data. The goal here was to find the key dimensions, based on the natural relationships between codes. In a brainstorming session we grouped similar properties into logical groups. As we analyzed more codes, we discussed whether to place each of them in one of the existing groups, as also the possibility of creating more groups or the creating of subgroups.

3 Dimensions

This section presents the results of the study. A total of eleven main dimensions was identified. Seven in the first group of characteristics (Prototyping) and four in the second (Evaluation). Two of the dimensions in the latter group are further divided in sub-dimensions, creating a total of thirteen. For each group we present the identified dimensions and provide illustrative examples from the literature.

3.1 Prototyping

The first group characterizes the relevant features of the immersive prototyping of ubiquitous systems. The seven dimensions are described below.

Fidelity of immersion can be described as the degree to which a virtual environment represent (in terms of appearance, sound, etc.) the real world, making the user feel immersed in the virtual environment. Techniques to immerse users within virtual environments go from the use of head-mounted displays to the use of CAVEs [3] (see, for example [22,5]), or other CAVE-derived techniques as presented in [5], the ImmersaDesk and the Infinity Wall [4]. An example of immersion is the case of immersive video inside a CAVE. This approach eases the evaluation and prototyping of mobile applications before its actual deployment, providing a high fidelity recreation of a user's experience [22].

3D modeling and simulation is a means to build virtual environments and/or devices. This is typically achieved through the use of game engines or 3D application servers. A key factor is to make the virtual environment realistic. It should be noted however, that creating a realistic simulation extends beyond its physical and graphical qualities. For example, [1] points out that creating a realistic simulated wireless device, implies being realistic in terms of connection latency, bandwidth, screen size, and battery life. According to [16] the use of game engine allows for a greater flexibility in the type of sensors that are used. Half-Life, Unreal, and Quake are the most used game engines. In [20] the OpenSimulator 3D application server is used. According to the author one advantage of using a 3D application server is to enable the remote and simultaneous connection of many users over the internet.

Embodied interaction support refers to the ability of the simulation to enable the reproduction of interactions that we use every day in the real world. Embodied interaction can be achieved through the use of interactions technologies such as motion tracking and gesture, or speech recognition. Users may, for example, interact with the virtual environment through the use of 3D gestures to point to devices and room objects [12], allowing for a more interactive and immersive experience. In [11] a scenario is built where a camera captures the size and location of human shadows and, based on that, triggers appropriate events (e.g. displaying a video).

Controlled Environment Manipulation. Ubiquitous systems' simulation can be molded to best serve the objectives of the designers and developers. We can define the behavior of the system and its objects, by programming them, by the use of models, or we can manually control/influence this behavior. The most common method, for expressing behavior is programming it through the use of scripts [1,20]. Another approach to attach behavior and functionalities to the system and its objects, is through the use of models [20,16]. Wizard of OZ can also be used to give behavior to the system [8] . The need to use people to realize the tests, and the fact that these tests are never realized in the exact same circumstances, are problems associated with the technique [22].

Context driven behavior happens when the system/prototype is able to capture the state of the environment and its relevant data, adjusting its behavior to that data. (e.g., a door opens, when a user gets close to it). This feature is present in many systems [7,15,5,16,1]. Approaches to gather context data include the use of sensors or other devices such as GPS systems [8], systems with information about networks, or specialized tools to extract information from the virtual environment [12,9]. Sensors, in particular, are very common in ubiquitous systems. According to [19], sensors can be classified as active or passive, i.e., they can detect values internally or from the virtual environment, respectively. Sensors can act as listeners for the system, enabling it to react to the environment [12] and store relevant sensor information for later use [22].

Multi-user Support. Enabling multiple users to explore the ubiquitous system allows for faster testing and assessment of the behavior of the system. This can be achieved by supporting the connection of multiple real users, or supporting the use of bots in the system. Supporting multiple real users enables evaluation of their behavior and their interactions in the system, but also evaluation of the system's behavior. In [20,16,15], this is an important dimension to integrate in the development of the ubiquitous system. Supporting the use of bots (i.e. AI expert software systems), enables the configuration of environments featuring multiple user using a limited numbers of real users, or to systematically explore an environment (e.g. to automatically identify unwanted behaviors) [16,15].

Hybrid prototyping takes advantage of a combination of simulated and real components to generate a mixed reality which can be used to assess the envisaged system. In [11], a mix of physical miniature prototyping and virtual prototyping is used. Two basic types of hybrid prototyping were found: one focusing on devices, another on services. Virtual devices enable testing specific systems (e.g., smartphones or sensors), and their integration in the ubiquitous environment, without actual physical deployment. In [1], images of the device's physical interface are used to create the virtual device. The embedding of sensors in virtual devices is addressed in [2]. A emulation framework allowing simulated hardware devices to interact with emulated software is described in [19]. Hybrid prototyping of services provides higher realism, accuracy and precision, since it can use real services. The most common cases are the integration of internet services, or the use of Bluetooth or similar protocols to integrate real devices [20,1]. Many systems tend to create their own communication components, using protocols such as TCP-IP or UPnP [12], or resorting to proxies [15], while other systems integrate existing network simulators into their framework [19] (thus reducing cost while providing users and developers an enhanced experience).

3.2 Evaluation

Evaluation is a key motivation in the immersive prototyping of a system. This second group of dimensions characterizes the different perspectives on ubiquitous systems evaluation. Two types of interests could be identified. Evaluation focused on the system and its developers, and evaluation focused on the users. Additionally, codes related to how to conduct experiments and collect data were also found.

System-centric evaluation is focused on evaluating the prototypes and the supporting frameworks, and is divided in two sub-dimensions.

Developer-Centric Evaluation. This evaluation is mainly concerned with knowing how easy it is for developers to develop accurate ubiquitous environments. This can be accomplished by collecting their feedback while performing a predefined prototyping task [16,20,7]. Other possibility is to use developers as test users, in order to determine if they can identify problems in ubiquitous environment.

Environments-Centric Evaluation. Immersive prototypes have the goal of creating virtual environments that are accurate enough replicas of real environments. These virtual environments must have the same properties that the real environments have, in order to give to the users a more realistic experience. To assess these environments, users that regularly explore the real environment should supply feedback to the developers, for them to know if the environment is accurate enough. At a more fundamental level, environments also need to be tested and evaluated regarding how the models react to user interactions or to context changes, in order to check if the prototype is behaving correctly.

User-centric evaluation focuses on how the users react to the ubiquitous system. Evaluating the users' behavior and their feelings when interacting with it, or evaluating if they can interact with the system efficiently and perform the tasks they were assigned.

Evaluating User Experience. User experience can be characterized by how well a person feels, when she interacts with the system. Through user experience evaluation, developers can know if the system that they are building will create a positive impact in people's lives. User experience evaluation techniques are widely used in many of the studies that were analyzed. Particularly in [7,20,9], a big importance is given to analyzing and comparing user experience and behavior, allowing redesign of the systems depending on the users' feedback.

Evaluating Usability. Usability is also a key goal on the process of developing ubiquitous systems. The more common approach to usability testing is through observation and recording users while they perform tasks. Others usability test methodologies are described in [13]. Maly et al. [9] built a framework for testing the usability of applications in virtual environments. The method consists in conducting specific tasks to evaluate usability. The approach builds on usability testing methodologies for desktop applications, combined with the evaluation of user behavior in ubiquitous environments.

Controlled experiments enable carrying out interaction tests under varied environment settings and context changes. A possibility is to replicate the exact same experiment with different users. All experiments will have the same system configurations, e.g., the events generated by sensors or the way the system adapts to context changes must be the same for any user that interacts with the ubiquitous system. Increasingly, ubiquitous systems are being developed to function and adapt to different scenarios [12,5,20]. A possibility, to assess what can happen when the system is deployed in different scenarios is to change one or several ambient settings in each experiment. These manipulations can go from re-positioning objects and avatars, to the manipulation of actuators and devices

Table 1. Relation between each evaluation dimension and each prototyping dimension

	Developer evaluation	Environments evaluation	Evaluating user experience	Evaluating usability
Fidelity of immersion	2	3	3	2
3D modeling and simulation	2	2	2	2
Embodied interaction support	2	1	3	3
Controlled env. manipulation	3	2	1	1
Context driven behavior	3	3	2	2
Multi-user support	2	2	1	1
Hybrid device prototyping	2	2	2	2
Controlled experiments	3	3	1	1

such as, lights, temperatures or displays [12,20,14]. The more common examples were the manipulation of lights and temperatures. In [12], the authors evaluate the suitability of the Philips iPronto device to new environments and their adaptability to different interactions.

Data collection is an increasing concern in the evaluation of ubiquitous systems. Developers can gather user feedback, either by allowing the user to freely explore virtual environments, or by making him or she follow or perform a list of tasks and storyboards [5]. Video recording or user observation are examples of methods to gather data about user behavior/performance, while performing tasks. The use of sensors to collect user performance, and save this data in log files, is another method that can be used. Conducting a series of interviews with users, or using surveys, are methods used to collect user feedback after the completion of the experiment.

3.3 Discussion

The relationships between the four evaluation dimensions and the dimensions related to the development of ubiquitous systems is presented in Table 1. The table should be read having in consideration that the primary point of analysis are the several types of evaluation. Evaluation dimension are assessed against each prototyping dimension, and also to the controlled experiments dimension. With this we want to highlight which dimensions are more critical for each evaluation dimension. The scale of values chosen to measure the relationship was: (1) - little influential, (2) - influential, and (3) - very influential. The values in Table 1 are derived from the analysis of the papers. They reflect the percentage of codes collected for each of the evaluation dimensions, when compared with the percentage of codes for each of the prototyping dimensions in the papers.

From Table 1 several conclusions can be reached. Developer-centric evaluation is more concerned with how to give behavior to the system, and how it reacts to change (be it context changes or user interactions). The ability to support multiple users with the purpose of realizing experiments is also an influential aspect of developer centric evaluation. Nevertheless, the other dimensions are also influential in this type of evaluation. Regarding the assessment of environments, the more realistic is the environment the better is the ability to evaluate the envisaged design. The realization of controlled experiments in the virtual environment, and how ubiquitous applications or smart objects react to changes are also among the most influential dimensions to assess environments. Allowing multiple users to interact with the environment, and supporting the use of virtual/real devices/services are the remaining influential dimensions in environment evaluation. For the user to have a good user experience, he should feel able to use most of the interactions that he usually uses in reality. The environment should be as realistic as possible, so that the user feels as embedded as possible in the environment. Regarding usability, the way users interact with the system and how much they feel immersed in the virtual environment are the more important dimensions to make user more connected with the environment, thus providing them a better way to accomplish their tasks. The possibility of interaction with virtual or real devices/services, and the way the ubiquitous system reacts to the user, are other influential dimensions to usability evaluation.

4 Case Study

The motivation to carry out the study described above appeared in the context of using immersive protoyping to support the design of a specific ubiquitous systems. The bar of a art gallery in Guimarães, Portugal has been equipped with public displays featuring the Instant Places system[1]. By default, users can interact with the system via their smartphones. The goal now is to enrich the system with new interaction capabilities. To study the viability of different alternatives prototypes will be developed in the APEX framework. The developers of the APEX frameworks, and the designers and developers of Instant Places, felt the need to identify what were the relevant prototyping dimensions that should be considered, and how they related to specific evaluation goals.

Considering that the main goal of the prototype to be developed is evaluating a number of new interaction techniques, and their influence the experience of being in the bar, the most relevant dimensions are: the Fidelity of immersion and support for embodied interaction. However, other features like Context driven behavior and Hybrid prototyping should also be considered as relevant. Indeed Hybrid prototyping is also quite relevant since the goal is to integrate Instant Places into the prototype, adding only the new interactions techniques. In the current case, it was decided to experiment with a table top interface. Hence, a prototype of the space was built (see Figure 1) that integrates virtual screen connected to the Instant Places service. Users are able to interact with the service both through their physical smartphones, or through a virtual table top interface in the simulated environment. The prototype was tested with users and while, due to space constraints it is not possible to discuss the results herein, they indicate that the table top interface was found useful.

[1] `http://www.instantplaces.org/` (Accessed: 29/1/2013)

Fig. 1. The art gallery prototype

5 Conclusions

This paper has presented a set of dimensions that address the immersive prototyping of ubicomp systems and their evaluation. Starting from a review of the sate of the art the relevant dimensions were identified. They characterize both the features that can be used to build immersive prototypes of ubiquitous environments, and the types of evaluation that can be carried out. By establishing a relation between these two groups, it becomes easier to decide which type of prototype to use given specific evaluation needs. Identifying the dimensions that should be the focus of a prototyping exercise provides two main advantages: it helps both reduce the costs of the process, and it helps focus the prototype on those features that will provide better results.

The approach is being used in a concrete example, where a public space has been equipped with ubicomp technology. In order to explore the impact of introducing new technology in the environment (e.g. tabletop interfaces) an immersive prototype has been developed. We are currently comparing the experience of using the prototype with the experience of being in the actual space.

References

1. Barton, J.J., Vijayaraghavan, V.: UBIWISE, A Ubiquitous Wireless Infrastructure Simulation Environment. Technical Report HPL-2002-303, HP Laboratories, Palo Alto (October 2002)
2. Brandherm, B., Ullrich, S., Prendinger, H.: Simulation framework in second life with evaluation functionality for sensor-based systems. In: UbiComp 2008 Workshop W2 – Ubiquitous Systems Evaluation, USE 2008 (2008)
3. Cruz-Neira, C., Sandin, D.J., DeFanti, T.A.: Surround-screen projection-based virtual reality: the design and implementation of the cave. In: SIGGRAPH 1993, pp. 135–142. ACM (1993)
4. Czernuszenko, M., Pape, D., Sandin, D., DeFanti, T., Dawe, G.L., Brown, M.D.: The immersadesk and infinity wall projection-based virtual reality displays. SIGGRAPH Comput. Graph. 31(2), 46–49 (1997)

5. Disz, T., Papka, M.E., Stevens, R.: Ubiworld: An environment integrating virtual reality, supercomputing, and design. In: Heterogeneous Computing Workshop, pp. 46–57 (1997)
6. Irawati, S., Ahn, S., Kim, J., Ko, H.: VARU Framework: Enabling Rapid Prototyping of VR, AR and Ubiquitous Applications. In: Virtual Reality Conference, VR 2008, pp. 201–208. IEEE (2008)
7. Li, Y., Hong, J.I., Landay, J.A.: Topiary: a tool for prototyping location-enhanced applications. In: UIST 2004, pp. 217–226. ACM (2004)
8. Li, Y., Landay, J.A.: Rapid prototyping tools for context-aware applications. In: CHI 2005 workshop – The Future of User Interface Design Tools (2005)
9. Maly, I., Curin, J., Slavik, P., Kleindienst, J.: Framework for visual analysis of user behaviour in ambient intelligence environment. In: Ambient Intelligence, pp. 85–107. InTech (2010)
10. Martin, M., Nurmi, P.: A generic large scale simulator for ubiquitous computing. In: Mobile and Ubiquitous Systems, pp. 1–3. IEEE Computer Society (2006)
11. Nakanishi, Y.: Virtual prototyping using miniature model and visualization for interactive public displays. In: Proceedings of the Designing Interactive Systems Conference, pp. 458–467. ACM (2012)
12. Nazari Shirehjini, A.A., Klar, F.: 3DSim: rapid prototyping ambient intelligence. In: Proc. Joint Conference on Smart Objects and Ambient Intelligence, pp. 303–307. ACM (2005)
13. Nielsen, J.: Usability Engineering. Morgan Kaufmann Publishers Inc., San Francisco (1993)
14. Nishikawa, H., Yamamoto, S., Tamai, M., Nishigaki, K., Kitani, T., Shibata, N., Yasumoto, K., Ito, M.: UbiREAL: Realistic Smartspace Simulator for Systematic Testing. In: Dourish, P., Friday, A. (eds.) UbiComp 2006. LNCS, vol. 4206, pp. 459–476. Springer, Heidelberg (2006)
15. O'Neill, E., Klepal, M., Lewis, D., O'Donnell, T., O'Sullivan, D., Pesch, D.: A testbed for evaluating human interaction with ubiquitous computing environments. In: First International Conference on Testbeds and Research Infrastructures for the DEvelopment of NeTworks and COMmunities, pp. 60–69. IEEE Computer Society (2005)
16. O'Neill, E., Lewis, D., Conlan, O.: A simulation-based approach to highly iterative prototyping of ubiquitous computing systems. In: 2nd International Conference on Simulation Tools and Techniques, pp. 56:1–56:10. ICST (2009)
17. Ostkamp, M., Bauer, G., Kray, C.: Visual highlighting on public displays. In: 2012 International Symposium on Pervasive Displays, pp. 2:1–2:6. ACM (2012)
18. Reilly, D., Dearman, D., Welsman-Dinelle, M., Inkpen, K.: Evaluating early prototypes in context: Trade-offs, challenges, and successes. IEEE Pervasive Computing 4(4), 42–50 (2005)
19. Reynolds, V., Cahill, V., Senart, A.: Requirements for an ubiquitous computing simulation and emulation environment. In: InterSense 2006. ACM (2006)
20. Silva, J.L., Ribeiro, Ó.R., Fernandes, J.M., Campos, J.C., Harrison, M.D.: The APEX framework: Prototyping of ubiquitous environments based on petri nets. In: Forbrig, P. (ed.) HCSE 2010. LNCS, vol. 6409, pp. 6–21. Springer, Heidelberg (2010)
21. Silva, J.L., Campos, J.C., Harrison, M.D.: Formal analysis of ubiquitous computing environments through the apex framework. In: ACM Symposium on Engineering Interactive Computing Systems (EICS 2012), pp. 131–140. ACM (2012)
22. Singh, P., Ha, H.N., Olivier, P., Kray, C., Kuang, Z., Guo, A.W., Blythe, P., James, P.: Rapid prototyping and evaluation of intelligent environments using immersive video. In: MODIE Workshop at Mobile HCI 2006, September 12-15 (2006)
23. Strauss, A.L., Corbin, J.M.: Basics of Qualitative Research: Techniques and Procedures for developing Grounded Theory. Sage Publications Inc. (1998)