An Infrastructure for Experience Centered Agile Prototyping of Ambient Intelligence

José Luís Silva
Departamento de Informática/CCTC
Universidade do Minho
4710-057 Braga, Portugal
jlsilva@di.uminho.pt

José C. Campos
Departamento de Informática/CCTC
Universidade do Minho
4710-057 Braga, Portugal
jose.campos@di.uminho.pt

Michael D. Harrison
School of Computing Science,
Newcastle University, Newcastle upon Tyne, NE17RU United Kingdom
michael.harrison@ncl.ac.uk

ABSTRACT
Ubiquitous computing poses new usability challenges that cut across design and development. We are particularly interested in "spaces" enhanced with sensors, public displays and personal devices. How can prototypes be used to explore the user's mobility and interaction, both explicitly and implicitly, to access services within these environments? Because of the potential cost of development and design failure, the characteristics of such systems must be explored using early versions of the system that could disrupt if used in the target environment. Being able to evaluate these systems early in the process is crucial to their successful development. This paper reports on an effort to develop a framework for the rapid prototyping and analysis of ambient intelligence systems.

Categories and Subject Descriptors
D.2.m [**Software Engineering**]: Miscellaneous – *Rapid prototyping.* H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems – *Artificial, augmented, and virtual realities.*

General Terms
Design, Human Factors.

Keywords
Ubiquitous and Context-Aware Computing, Modelling, 3D virtual environments.

1. INTRODUCTION
Agile development techniques provide early feedback about the implications of a design concept through exploration of a concrete prototype. Formal techniques on the other hand provide the means to analyze the implications of a proposed design before implementation. These two techniques are often considered to be in opposition to one another. In ubiquitous computing systems both techniques are required. Analysis techniques are required to ensure that the right information is provided to users when they need it, to ensure that the context will have an appropriate and perceivable effect on users, to prove that similar interactions that take place within the environment will be consistent with each other. Concrete prototypes will be required to ensure that the system will have a texture and an impact on users as expected. This paper presents and illustrates a proposal for a system that will combine the two techniques, enabling the development of formal specifications that can be used for analysis, and providing the means to move rapidly from the specification of the system to a prototype that exhibits properties of the design that can be explored by users.

The paper builds on previous work concerned with the development of prototypes from formal descriptions (e.g. ICO[4] and Marigold[27]). While the work described here builds on this research, distinctive features of ubiquitous computing systems are of particular concern and require a new approach. In ubiquitous systems people are immersed within the environments created by the system. Their interactions are affected by the context used by the system to predict what the user "means" by the action that has been taken.

It is proposed that the ubiquitous system is modeled using colored Petri nets (CPN). The CPN model makes use of a generic notion of context and focuses particularly on user interaction with the system. Analysis involves a combination of animation and model checking. The Petri net, along with a description of the objects that make up the virtual environment, are used to create a 3D world using a 3D application server (OpenSimulator), see figure 1. The approach supports rapid simulations of proposed designs based on CPN models.

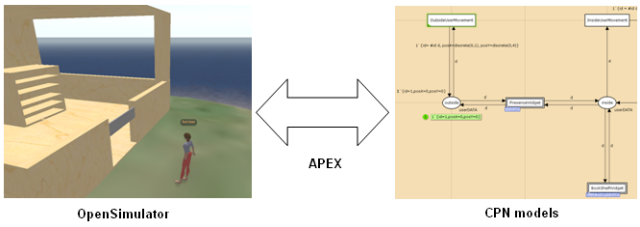

Figure 1 - Proposed process

Modeling techniques make it possible to describe, analyze and reason about systems before implementation. Of particular interest here are techniques supporting reasoning about context-aware aspects of the design. Simulation enables the exploration of mobile devices in 3D environments as well as the visualization of communication between devices, visualizing state changes using 3D animations.

EICS'09, July15–17, 2009, Pittsburg, Pennsylvania, USA.

2. THE APEX SYSTEM

In order to support the rapid prototyping of ubiquitous computing systems, we have started the development of the APEX platform (Agile Prototyping for usEr eXperience).

The APEX system is being designed with the following characteristics: 1) it should include an adequate notation for describing models of the ubiquitous system taking into account context, modularity and usability; 2) it should generate Virtual Reality prototypes from these models in a (semi-) automatic way; 3) the fidelity of these prototypes should be such as to support the evaluation of the eventual human (usability and social) impact of the design.

A key criterion for the success of the framework will be its fidelity as a tool for evaluation of proposed systems. The framework will plug into virtual reality (VR) simulations of the environments, and allow for the integration of users' portable devices so that a user or users can be immersed within the environment using a mixture of reality (physical mobile device) and virtuality (virtual sensors and virtual public displays). Developers and users will be able to use simulations to navigate environments and to develop an impression of what it will be like to use the final systems once fielded. Simulations must therefore be sufficiently rich and textured to address usability requirements (e.g., experience requirements) that depend on the target environments.

3. BACKGROUND

The literature concerned with prototyping ubiquitous systems (see [8] for a good overview) focuses primarily on the development of prototypes of isolated devices. The systems are typically to be explored outside the context of the fully integrated system in its proposed setting, see Abowd et al.'s paper in [8] for a useful discussion of this contrast.

The integrated system is crucial to understanding user impact in the type of system of concern here. Yet, typically, integration with the built environment is impossible at the early stages. The displays, devices and sensors form an integrated whole contributing to the texture of the target environment. Fielding such systems for testing purposes is, in many cases, not feasible (consider a hospital or an airport). Additionally, developing such high-fifelity prototypes might imply commitment to design decisions that would be expensive to reverse [12].

A software framework is needed to enable rapid prototype development. Existing platforms for context aware or ubiquitous computing (e.g., [5,6,12,13]) are more focused towards building experimental systems than in supporting the design process. Cogtool [15] has many of the properties of the kind of system envisaged but it is targeted to the prototyping of single devices with explicit interactions. We need to explore substitutes for the physical characteristics of the built environment, including use of virtual environments supporting virtual displays and virtual sensors, see [3,19].

3D Application Servers such as SecondLife™ (secondlife.com), OpenSimulator (opensimulator.org) or There (there.com) provide a fast track to developing virtual worlds. They are a natural choice for the development of the type of prototype we are aiming at. A number of alternatives were compared regarding configurability, development support features, and cost. OpenSimulator was chosen because of its availability, the ability to create objects, and to attach behaviors to them. Its backend, which can be programmed, makes it highly configurable and extensible.

Several possibilities were considered regarding the modeling of ubiquitous environments. These included: Hybrid high-level Nets (HyNets) [26], Communicating Sequential Processes (CSP) [10], Flownets [22], ASUR++ [16], Interactive Cooperative Objects (ICO) [4] and CPN [14]. In the end the choice was made to use CPN. This was due mostly to the substantial set of tools available, making it easier to do our own development. While the language lacks the features of, for example, HyNets or ASUR++, we believe it provides enough expressive power to suit our purposes, and CPN tools provides a rich modeling, simulation and analysis environment that fits out purposes. Notice that virtual reality modeling languages such as VRIXML [7] have not been considered. The goal it not the modeling virtual reality environments, but to generate them from models of ubiquitous environments.

Concerning context studies, the 5W1H [25] and Context Toolkit [9] approaches have been explored, and are being used as the basis for integrating context in the modeling approach proposed.

A number of systems aiming at the simulation of ubiquitous environments can be found in the literature. Systems such as UbiWise[3], TATUS [20] or UbiREAL [24] focus on supporting developers of new devices or software. The simulation acts as a test bed for the devices/software being developed. That is not the focus of the current work. Rather, the goal is to support the design of the environment itself, with a particular emphasis on how users will experience it.

Systems such as 3DSim [2] or UbiWorld [22] place a higher emphasis on user-environment interaction. Their vision is similar to ours. We want to start from models of the ubiquitous environment to integrate with development processes. UbiWorld envisages the use of programming languages. We are unaware of any further research or development addressing the prototyping of ubiquitous environments, from a user experience perspective, in the context of a development process.

4. PROPOSED APPROACH

The core problem is being able to test design ideas for ubiquitous environments during the early phases of design. This implies that analysis must be performed without the need to fully develop and install the envisaged system on location. Our proposal is to use a 3D Application Server as the basis for a prototyping framework that will enable agile development of simulations of the ubiquitous environment. This framework should enable different degrees of fidelity for the prototypes, from simple desktop simulations, to fully immersive experiences in a CAVE environment (see [21]).

The first step in developing such a framework was the selection of an appropriate modeling language with a focus on modeling for rapid prototyping, and the modeling of user context. Several modeling approaches were studied and compared and the choice was made to use CPN, largely because of the quality of tool support available (see above).

Since no one language addresses both the architectural and behavioral aspects adequately (from a prototyping perspective) a combined approach has been favored. Behavioral aspects are modeled in CPN models. These models identify the components of the system, and their behavior. Detailed architectural information (location and physical attributes of the components,

and of the space) is defined interactively when building the prototypes using OpenSimulator tools.

The framework is currently under development. It will enable the simulation of environments, enhanced with sensors and displays, in which the user can move and interact, both explicitly and implicitly, to access services. The framework is basically a module loaded by a 3D application server. This module is responsible for generating the ubiquitous environment simulation, and for using the CPN models to drive it. Events in the simulation (context changes or user interactions) are forwarded to the CPN models, and the responses of the CPN (transitions fired) are reflected back in the simulation. Interaction with the envisaged system is supported both through the virtual environment and through personal devices connected to the system via Bluetooth.

A 3D application server such as OpenSimulator, besides streamlining prototype development, has the added advantage of making it easy to deploy the prototypes to distributed users. When combined with a desktop computer to navigate the world, it is also a low cost approach since hardware and software requirements are low. The desktop approach however, makes it harder to create a sense of presence. Presence will be addressed by connecting the system developed on a desktop to external devices and to a CAVE system. This will enable more realistic prototypes of the prototyped environments with the aim of making it easier to create a sense of "being there".

5. LIBRARY CASE STUDY

To better explain and demonstrate the approach an illustrative example is used. A library, composed of books with RFID tags (or some other technology to locate them within the library), bookshelves, users, entry/exit gates and screens. When a registered user arrives at the entry gate, a screen displays what book(s) have been requested and the entry gate opens. It is assumed that the books have been chosen at the user's desk before going to the library.

The system guides the user to the required book sending directions to the PDA of the user via Bluetooth. Sensors recognize the position of the user in real-time. As the user approaches the book a light with a color previously notified is turned on. This is to cater for the situation where many users look for books in nearby locations. When the user takes the book, the light in the book is turned off.

When a user approaches the exit gate, the list of requested and returned books in this 'session' is displayed on the exit screen. When the user is allowed to leave, the door is opened and the user can go out. In the case of failure the door remains closed.

To help understand the models a plan of the library with coordinate positions is presented in figure 2. The figure presents the simplified library composed of an entry/exit gate and only one bookshelf. Two sensors, referred to as *contextWidgets*, detect the presence of library users. They are responsible for detecting changes in the environment; in this case study one *contextWidget* is present at the gate and one at the bookshelf.

5.1 CPN Modeling

In this section a description of the CPN model of the library developed in CPN Tools [1] is presented. The language used in the CPN models is CPN ML, a variant of Standard ML (see [17] for a good overview).

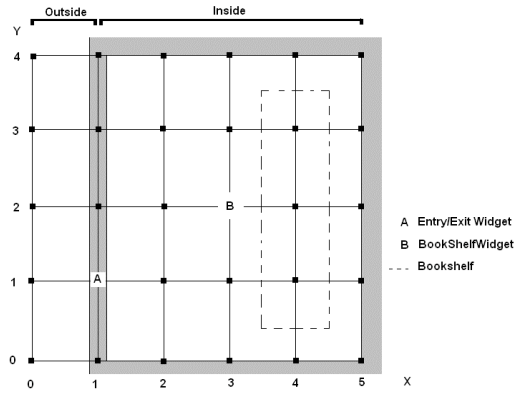

Figure 2 – Plan of the library

The model uses several data types which are defined in the declaration field of CPN Tools. For instance, type *userDATA* is used to represent the user features (e.g. identification and position).

The model is composed of seven components: *library*, *Widget*, *MovementAtDoor*, *Screen*, *Door*, *BookShelfWidget* and *PDA* (figures 3 to 7 present some of them). Each component describes different behaviors, together they describe the whole system. Arcs in the models have an associated variable of the same type as the places they connect. They can change the value of the tokens that transit on then. For instance, the arc from the transition *OutsideUserMovement* to the place *outside* (figure 3) changes the value of the token using three attributes. In one of them the variable *posY* (user position in the Y-axis) gets a random value between 0 and 4, calculated by the *discrete* function.

CPN models support hierarchy, allowing the modeler to combine small CPN components into a large net. Labels (see the small rectangles below some transitions in figure 3) indicate the presence of sub models with internal detail.

5.2 Behavioral description

The library model (figure 3) is triggered by a token of type *userDATA* being placed at the *outside* place indicating the presence of a new user outside the library. To simplify the explanation, the model shows only one user (one *userDATA* token). By increasing the number of these tokens more users can be represented and simulated. Initially only the *OutsideUserMovement* transition will be enabled. When the transition fires a token is consumed by it and placed back in the outside place, indicating the movement of the user outside the library. User positions are defined simply as a set of coordinates from (0, 0) to (5, 4).

In order for the user to move inside the library (figure 3 *inside* place) the *contextWidget* (*PresenceWidget*, figure 4) must detect the user's presence at the entry gate (user in position (1,1)) enabling the *Moving* transition. The detection of the user at position (1, 1) is represented in the model by a condition in the *userDetectedOutside* transition (*#posX d=1 andalso #posY d=1*). In the *PresenceWidget* the token moves from the *outside* place to the *AtGate* place once the *userDetectedOutside* transition is executed. Within the PresenceWidget a further component is defined (*Moving*) which is further refined in the *MovementAtDoor* component (Figure 5). This behavior captures what happens when the token makes the transition to the *inside* or *outside* place.

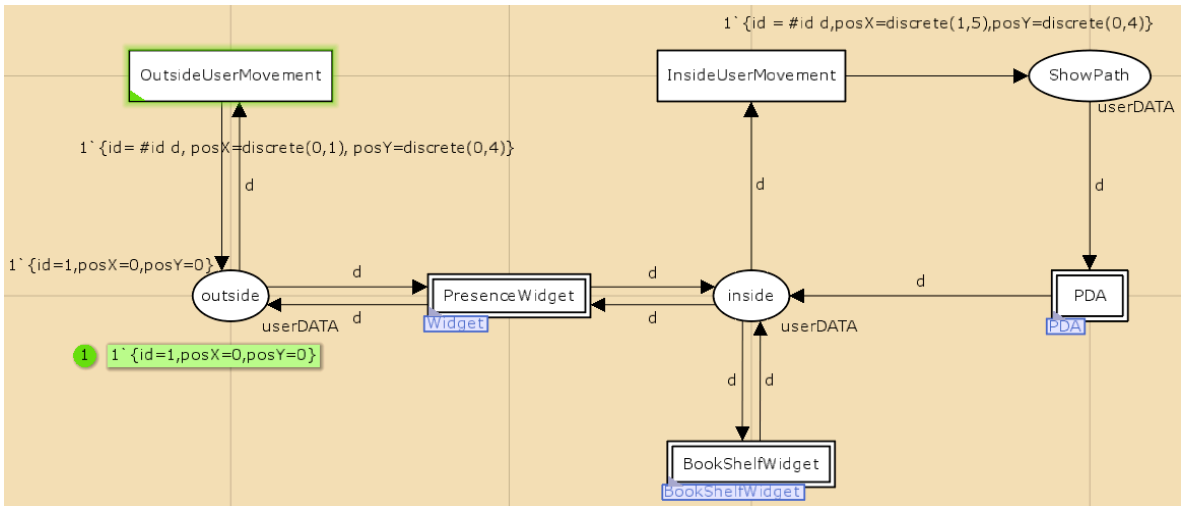

Figure 3 - Library model

In the case of *MovementAtDoor* when a user arrives at the entry gate the *Screen* (figure 6) must show relevant information and the *Door* (figure 7) must go to the opened state if the user has access authorization. When the user passes the door (*GoingInSide* transition) or away (*GoingOutSide* transition) the screen display's default information (*figure 6*) and the door goes back to the closed state (*figure 7*).

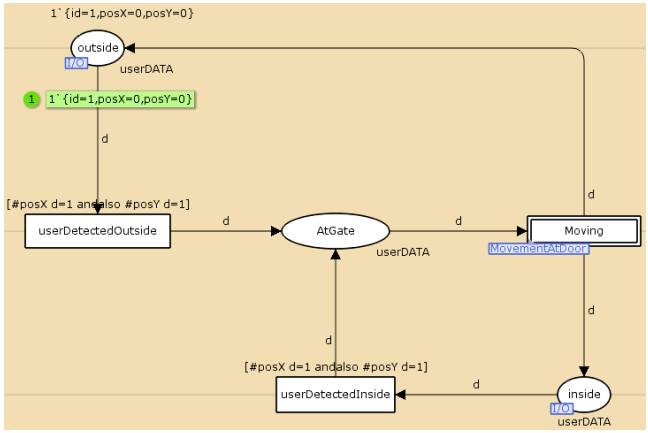

Figure 4 - Widget component

In the *MovementAtDoor* component not all places are visibly connected. *Fusion* places are used in this component. For instance, the *AtGate1* place has a label (*Fusion Screen*) which coincides with the *AtGate1*, place of the *Screen* component (figure 6). When the user is at gate, i.e. a token is at the *AtGate1* place the *ShowInfo* transition of the screen model becomes enabled if the condition is satisfied and if each place connected to the transition holds at least one token. This enables the *ShowInfo* transition to be executed, moving the *userDATA* token to the *informationShowed* place and the *ScreenDATA* token to the *Showing* place. Next, in the *Door* component (figure 7), the *OpeningDoor* transition of the Door model is executed, thereby the gate is open and the screen shows the relevant information.

Once in the *DoorOpened* state (fusion place of *Door* and *MovementAtDoor* components) the user can move (*Movement* transition) into the library (*goingInSide* transition) or move outside (*goingOutSide* transition). The screen and door components respectively show default information on the screen (*ScreenDATA* token to the *Default* place) and close (*DoorDATA* token to the *Close* place). If the user is inside (*#posX d>1* condition) the userDATA token moves to the *inside* place otherwise it moves to the *outside* place (see figure 5).

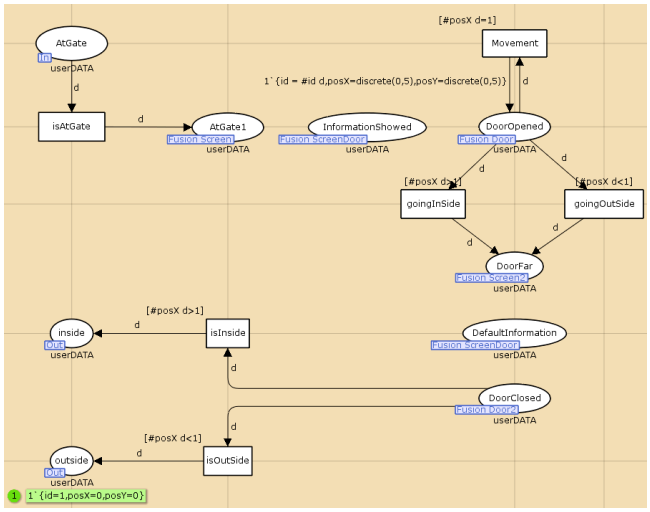

Figure 5 - MovementAtDoor component

This model can be used to analyze a number of properties, for example that given specified assumptions the user will find the book and exit the library. This can be achieved via the state exploration features of CPN Tools, by writing adequate query predicates using temporal logic formulae over the possible markings and transitions in the model.

5.3 Virtual world modeling and simulation

Once the basic components of the model having been identified, the virtual world representation of the envisaged system can be constructed. This is done using the tools provided by OpenSimulator, by defining which objects are present along with their physical characteristics including position in space.

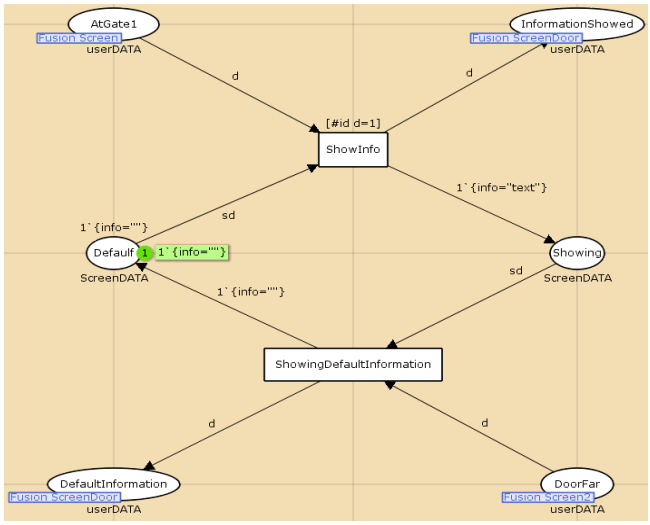

Figure 6 - Screen component

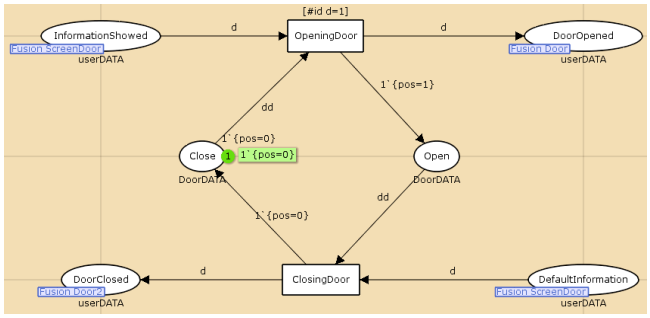

Figure 7 – Door component

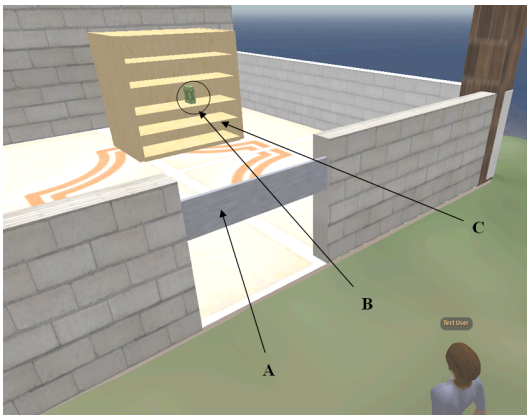

Figure 8 – Library simulation, components: A-Entry/exit gate; B- Book; C- Bookshelf

Figure 8 presents a view of a first attempt at modeling the simplified virtual library. The following components can be identified, the entry/exit gate, the bookshelf and the book. The *contextWidget* component is transformed into an object that informs the system of the proximity of a user.

In order to run the simulation, and have the environment adequately respond to users navigating it, it is necessary to connect the virtual world to the CPN model. It is that model that provides the semantics governing the behavior of objects in the virtual world simulation. To achieve this we are developing a C# *IRegionModule* that is loaded by the OpenSimulator server.

COMMS/CPN [11] is a Standard library provided by the CPN tools to establish communication between CPN models and external processes. This library implements TCP/IP communication functions (e.g. *openConnection*, *acceptConnection*, *send* and *receive*). Several packages are available for connecting CPN Tools and Java/C processes. Because OpenSimulator modules are developed in *C#* we are currently working on a C#/CPN communication package. Currently it is already possible to perform bidirectional exchanges of information between OpenSimulator and CPN tools.

6. CONCLUSION AND FUTURE WORK

This paper has reported on ongoing work that aims to provide prototyping support in the early stages of developing ubiquitous computing systems. The main expected contribution of the work is the development of a framework to support the simulation. Prototypes will consist of virtual environments in which the users can navigate, and interact with available (virtual) devices.

The desktop approach makes it possible to generate prototypes rapidly, and to test them more economically. However, creating a sense of presence in such a setting might prove difficult. To address this, we plan to link the prototyping framework to a CAVE system. This will enable a more immersive experience of the prototype, making it easier to create a sense of "being there".

Developing a solution to connect external devices to the virtual environments generated is another feature that will enable a more realistic user experience. In this case we want to enable users of the prototypes (either at a desktop or at a CAVE) to interact with them via their own portable devices (e.g. [18]) increasing the ease of use and facilitate the interaction with the simulation.

An obvious issue arising from the use of models and simulations is the extent to which their results would correspond to an equivalent real-world evaluation of a fully functioning system. Model validity is not conditioned to a perfect match, but assuming that the results will never be exactly the same, it is crucial that we understand in which aspects of the evaluation the matching is more appropriate and in which aspects it is more limited.

With the above in mind, a study will be carried to evaluate user experience within generated prototypes. It is envisaged that both free and guided exploration of the environments constructed will be used. Approaches will be selected to determine user reaction when exposed to the environment to establish the fidelity of the prototype environment. An existing ubiquitous computing system will be used as a case study. The goal of the exercise will be twofold: 1) to assess the capabilities of the infrastructures as a tool for experience centered agile prototyping for ambient intelligence; 2) to explore the fidelity of these prototypes (whether desktop or virtual reality) as a factor in evaluating the eventual human (usability and social) impact of the design.

This will entail an analysis of the effort involved in creating the prototypes, and a comparative analysis of user experience in the real system and in the prototypes, and will be an important contribution in informing designers on which aspects of the evaluation to focus when using models.

In parallel the models originating the prototypes will be analyzed using reasoning tools, the results of the two styles of analysis compared, and complementarities explored.

7. ACKNOWLEDGMENTS

This work is supported by PhD Grant SFRH/BD/41179/2007 from Fundação para a Ciência e Tecnologia (FCT, Portugal).

8. REFERENCES

[1] A.V. Ratzer, L. Wells, H.M. Lassen, M. Laursen, J.F. Qvortrup, M.S. Stissing, M. Westergaard, S. Christensen, K. Jensen 2003. CPN Tools for Editing, Simulating, and Analysing Coloured Petri Nets. In Applications and Theory of Petri Nets 2003, pp 450-462. LNCS vol. 2679. Springer.

[2] Ali A. Shirehjini N., Klar F. 2005, 3DSim: rapid prototyping ambient intelligence, Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies, October 12-14, 2005, Grenoble, France

[3] Barton, J.J and Vijayaraghavan 2003. UBIWISE, a simulator for ubiquitous computing systems design. Tech. Rep. HPL-2003-93, HP Laboratories, Palo Alto.

[4] Bastide, R., Schyn, A., Winckler, M., Nedel, L.P., Freitas, C., Navarre, D. and Palanque, P. 2005. A formal description of multimodal interaction techniques for immersive virtual reality applications. In Human-Computer Interaction - INTERACT 2005, pp 170-183. LNCS vol. 3585. Springer.

[5] Braubach, L., Pokahr, A., Moldt, D., Bartelt, A., Lamersdorf, A. 2002. Tool-Supported Interpreter-Based User Interface Architecture for Ubiquitous Computing. In Interactive Systems, pp. 89-103. LNCS vol. 254. Springer, 2002.

[6] Buchenau, M. and Suri, J.F. 2000. Experience prototyping. Proceedings DIS'00, Brooklyn, New York. ACM Press pp. 424-433.

[7] Cuppens E., Raymaekers C., Coninx K., 2004. VRIXML : A User Interface Description Language for Virtual Environments, Proceedings of Developing User Interfaces with XML: Advances on User Interface Description Languages, satellite workshop of Advanced Visual Interfaces 2004, Gallipoli, IT, May 25 2004

[8] Davies, N., Landay, J., Hudson, S., Schmidt, A. 2005. (Eds.) IEEE Pervasive Computing. Special issue on rapid prototyping. Volume 4, Issue 4, October 2005.

[9] Dey A.K. Abowd G.D. Salber, D. 1999. The context toolkit: Aiding the development of context-enabled applications. In Proc. of CHI '99, pp 434-441. ACM.

[10] Donk O.A., Zwiers J., Van Schooten, B.W. 1999. Modelling interaction in virtual environments using process algebra. In TWLT 15: Interactions in Virtual Worlds, pp 195-212, 1999.

[11] G. Gallasch and L. M. Kristensen 2001. COMMS/CPN: A communication infrastructure for external communication with Design/CPN. In Third Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools, pp 75-91. Department of Computer Science, University of Aarhus, Denmark, 2001.

[12] Garlan, D., Siewiorek, D.P., Smailagic, A & Steenkiste, P. 2002. Project Aura: toward distraction-free pervasive computing. IEEE Pervasive Computing. 1(2):22-31, Apr-June 2002.

[13] Harter, A., Hopper, A., Steggles, P, Ward, A & Webster, P. 2001. The anatomy of a context-aware application. Wireless Networks 1. pp 1-16.

[14] Jensen, K., Kristensen, L.M., Wells, L. 2007. Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. Int. J. Software Tools and Technology Transfer, 9(3), 213-254, May 2007.

[15] John, B.E., Salvucci, D.D. 2005. Multi-Purpose Prototypes for Assessing User Interfaces in Pervasive Computing Systems. IEEE Pervasive Computing 4(4), 27-34, October 2005.

[16] L. Nigay, E. Dubois, P.D. Gray 2002. Asur++: A design notation for mobile mixed systems. Berlin: Springer. Lecture Notes in Computer Science, 2411:123-139 (2002).

[17] L.C. Paulson 1996. ML for the Working Programmer (2nd edition). Cambridge University Press.

[18] M. Gutierrez, F. Vexo, D. Thalmann 2004. The Mobile Animator: Interactive Character Animation in Collaborative Virtual Environments. In IEEE Virtual Reality Conference 2004 (VR 2004), pp. 125-132, IEEE.

[19] O´Neill, E., Lewis, D., McGlinn, K. and Dobson, S. 2007. Rapid user-centred evaluation for context-aware systems. In Interactive Systems, pp 220-233. LNCS vol. 4323. Springer.

[20] O'Neill E. 2005. TATUS: A Ubiquitous Computing Simulator. PhD thesis, The University of Dublin, Trinity College, 2005.

[21] Slater, M. and Steed, A. 2000. A virtual presence counter. Presence: Teleoperators and Virtual Environments, 9(5):413-434, 2000.

[22] Smith, S., Duke, D. and Massink, M. 1999. The hybrid world of virtual environments. Computer Graphics Forum, 18(3):287-307, 1999.

[23] T.L. Disz, M.E. Papka, and R. Stevens 1997. UbiWorld: An Environment Integrating Virtual Reality, Super-computing, and Design. In Proc. 6th Heterogeneous Computing Workshop, pp 46-57, 1997.

[24] Tamai M., Nishigaki K., Kitani T., Shibata N. Yasumoto K., Ito M. Nishikawa H., Yamamoto S. 2006. Ubireal: Realistic smartspace simulator for systematic testing. In UbiComp 2006: Ubiquitous Computing (2006), pages 459-476, 2006.

[25] W. Jang, S. Woo 2004. Unified context representing user-centric context: Who, where, when, what, how and why. In Int. Workshop ubiPCMM05, pp 1-4. CEUR.

[26] Wieting, R. 1996. Hybrid high-level nets. In The 1996 Winter Simulation Conference. pp 848-855. ACM Press.

[27] Willans, J.C. and Harrison, M.D. 2001. A toolset supported approach for designing and testing virtual environment interaction techniques. International Journal of Human Computer Studies. 55(2) pp. 145-166, 2001.