Factoring user experience into the design of ambient and mobile systems

Michael D. Harrison, Christian Kray, Zhiyu Sun and Huqiu Zhang

Informatics Research Institute, Newcastle University, NE1 7RU, UK.
Michael.Harrison@ncl.ac.uk

Abstract

The engineering of ubiquitous computing systems provides important challenges. Not least among these is the need to understand how to implement designs that create a required experience for users. The paper explores a particular class of such systems for built environments. In particular it is concerned with the capture of experience requirements and production of prototypes that create experience. The aim is to develop methods and tools for such environments to enable the creation of particular sorts of experience in users. An approach that combines the use of scenarios, personae and snapshots with the use of prototypes and models is described. The technique aims to elicit an understanding of the required experience of the system and then create a design that satisfies the requirements.

1. Introduction

While a wide variety of experimental ubiquitous computing systems have been developed in recent years, relatively little effort has been aimed at the problems of engineering the interaction of these systems. This paper addresses a class of such systems that involve public displays, hand held devices and location sensors. The systems that are of interest may be used to deploy services to users of built environments (office, leisure complex, hospital, airport or museum). Such systems enhance the user's experience of the environment by offering information about it and the services available within it. The systems envisaged here are always on in the background, and provide services to the user according to their context and location.

The success of these systems depends on a number of factors, including software and hardware reliability and usability. The user's experience of these systems is particularly important but what experiencing a system in a particular way might mean is difficult to express and then to implement in a system. Examples of experience in a built environment might include: *place* (feeling that you know where things are); *absence of anxiety*; *safety* or *security*.

These experiences can be valuable in making the environment more attractive to users. They can also enhance in users an awareness of issues such as safety or security and therefore modify their behavior. Weiser and Brown [27] in early discussions of ubiquity highlighted the importance of experience when they used the term “calm technology”. Their vision of ubiquitous systems was that it would create an experience akin to lack of anxiety and feeling in control. Forlizzi and Battarbee [10] go further and make distinctions between types of experience relating to “fluent”, “cognitive” and “expressive” interactions. The issue to be addressed is how to elicit, model and implement these experience requirements. A particular concern is what role that formal modeling could play in such a process.

Many factors affect the experience that users have of built environments. These include the texture and physical characteristics of the environment and where information displays are situated. The paper uses the same example throughout which is based on an airport. In each space within the airport there is a public display that displays messages about flights that are relevant to passengers that occupy the space at any one time. Each passenger carries a mobile phone and receives messages on this phone that are specifically relevant to their flight and location. The design deliberately adopts a simple set of techniques for deploying information to users. It should be noted from the outset that many schemes are feasible for combining public displays with private information, see for example [13,16]. The scheme used here is illustrative of a range that could equally be addressed by the techniques described. A prototype is described as well as a formal model used to explore experience requirements and the creation of designs producing the required experience for users.

The structure of the paper is as follows. Section 2 introduces the main issues associated with experience relevant to the paper. Section 3 discusses issues of experience elicitation and proposes a set of feasible experience properties of the example. Section 4 comments on experience articulation. It identifies the problems associated with expressing an experience requirement so that an engineer can use it to produce a design. Section 5 discusses experience prototyping. It describes prototypes that were developed as a basis for exploring features of the airport environment. Section 6 describes a specific model of the airport system. It discusses the role that modelling and analysis techniques might play.

2. Factoring in experience

Before building a system that creates a given experience it is necessary to understand what experience is appropriate. It is then necessary to express the experience in a form that can be used by designers and engineers. It is only possible to be sure of the experience that is created in a design when the system is in-situ in its proposed setting. However it is usually infeasible to explore the role of a prototype system in this way, particularly when failure of the system might have safety or commercial consequences. A prototype running in a busy airport will have unacceptable consequences if it fails. It may have safety or commercial consequences

if crucial information is not provided clearly in a timely way. At the same time, deploying a system that is close to product when many downstream design commitments have already been made will be expensive to redesign. Exploring and assessing usability and experience of prototypes, however close to product, in its target environment is therefore unlikely to be acceptable or cost effective. Techniques are required to enable early system evaluation.

Once an experience has been understood, it should be expressed in a form that supports construction of an environment that creates the experience. This paper addresses a number of questions. How are the experience requirements for such a system established? How are they articulated so that a system can be designed to implement them? How can models or prototypes be used to check whether the required experiences are created in the design before committing to a final implementation?

The paper explores available methods for experience elicitation, noting the role that scenarios play not only in capturing features of an experience but also providing a basis for visualizing what a proposed new design would be like in the context of that experience. The paper also explores the role that snapshot experiences play in deriving properties that can be applied to models of the proposed design. Snapshot experiences can also be used to inspire or to construct further scenarios that can also be explored as a basis for visualization.

3. Experience Elicitation

McCarthy and Wright [22] and Bannon [2] have argued that while GUIs lead to an emphasis on technology as tools, systems such as those described in this paper require thought about how people live with the technology. This change has also been described as a shift from understanding use to understanding presence [15]. Existing methods of user-centered design do not help engineers understand which designs are likely to lead to feelings of resistance, engagement, identification, disorientation, and dislocation amongst users.

Experience can be understood through a variety of mechanisms. It can be understood through narrative by:

- Asking people to tell stories about experiences they have had of an existing system.
- Exploring alternative worlds in which the experience would have been different.

Many authors (for example [14]) discuss the use of scenarios based on these narratives. Personae are also used as a filter for understanding the scope of experience requirements.

Scenarios alone are not sufficient to provide clear experience requirements for which the route to implementation is clear. They may be biased towards the current system. They may lead unacceptably to a proposed solution that fails to capitalize on the opportunities that the new technology affords and is instead an incremental development of the old one. The collection of scenarios and personae

are unlikely to be sufficiently inclusive to provide a complete picture of the experience of the system. However, scenarios provide rich descriptions that are extremely valuable in the experience elicitation process. At the same time scenarios provide a medium that can later be used with proposed paper designs or prototypes to "visualize" effectively what the design requirements are.

Other techniques are required to complement scenario orientated techniques. It is necessary to augment some of the limitations of scenarios to obtain a richer understanding of what experience is required. Cultural probes provide an orthogonal perspective [12]. They can be used to elicit snapshot experiences. These are understood as fragments of experience provided by users that can be used to help understand how users experience an existing system. The aim is that these snapshots should be used to establish what is required of a new design. Eliciting snapshots involves subjects collecting material: photographs, notes, sound recordings, that they believe capture important features of their environment. These snippets may make sense as part of a story. The information gleaned may help understand characteristics of the current system that cut across a range of scenarios. In the example (see Section 1) the purpose of the ambient and mobile system is to notify passengers about the status of their flights, wherever they are in their passenger journey. Passengers might be asked to identify snapshot experiences of the existing airport environment. They may be invited to take photographs or make audio-video recordings and to produce commentaries or annotations of these snapshots explaining why the snapshots are important. The following are plausible examples:

- **S1:** photographs of the main display board with comments such as:
  - "I like to be in a seat in which I can see this display board";
  - "I wish that the display board would tell me something about my flight - it disturbs me when it simply says wait in lounge",
  - "How can I be sure that it is up-to-date?";
- **S2:** photographs of signposts to the departure gate with annotations such as: "I wish I had better information about how far it was and whether there were likely to be any delays on the way";
- **S3:** tape recordings of helpful announcements and tape recordings of unhelpful announcements, with annotations such as "These announcements do not happen often enough and announcements for other flights distract me";

This information requires organization to ensure that subsets of facilities are not neglected. Snapshot experiences may be used to trigger further narratives. The analyst might enquire of a user who has generated a snapshot: "Can you think of situations where this particular feature has been important?" By these means they may inspire a scenario that would not otherwise have been gathered. They can also be converted into properties that the new design should satisfy. Hence the comment relating to S1: "How can I be sure it is up-to-date" could lead to a number of properties:

- **P1:** when the passenger moves into the location then flight status information is presented to the passenger's hand-held device within 30 seconds

- **P2:** information on public displays should reflect the current state of the system within a time granularity of 30 seconds

In future sections these properties are considered in more detail.

4. Experience articulation

As discussed in the previous section, scenarios and snapshots together capture experience characteristics of the system. The question is how this information can be used along with prototypes and models to produce an implementation that can create the desired experience. Experience provides an imprecise basis for implementation requirements. It becomes necessary to explore the proposed system experimentally: "I will know what it is when I have got it". Buchenau and Suri [6] describe a process of probing using scenarios and approximate prototypes. For example their method might involve asking people to carry dummy devices around with them to visualize how it would feel. Their approach ("experience centred design") enables imagination of the experience that users would have with the design. The quality and detail tends to vary: from "mocking up", using prototypes that simply look like the proposed device but have no function, to more detailed prototypes that are closer to the final system. The design intended to create the experience emerges through a process of iteration. Articulation of the required experience is encapsulated in the design that emerges from the process. To explore and to visualize the proposed design effectively it is important that prototypes can be developed with agility. It should be possible to try out ideas and to dispose of prototypes that are not effective. It should be possible to use a context that is close to the proposed target environment. These early prototypes help envision the role of the "to-be-developed" artefact within the user's activity. Prototypes can also be used to "probe" or to explore how valid and representative the scenarios are. This can be used as a basis for generating a discussion about alternative or additional scenarios.

Snapshot experiences can be a valuable aid to analysts. They can form the basis for properties that the system should satisfy. The conversion from snapshots to properties relies on the experience and practice of the analyst. Such properties should be independent of specific implementation details. Whereas scenarios can be explored with prototypes, properties require the means to explore the design exhaustively. This can be done, as in heuristic evaluation, through the expertise of a team of analysts exploring a description of the design systematically. It can also be automated through model checking as will be discussed in a later section. The same model that is appropriate for experience requirements checking can be used to analyze other properties that relate to the integrity and correctness of the system.

- **P3:** when the passenger enters a new location, the sensor detects the passenger's presence and the next message received concerns flight information and updates the passenger's hand-held device with information relevant to the passenger's position and stage in the embarkation process.

- **P4:** when the passenger moves into a new location then if the passenger is the first from that flight to enter, public displays in the location are updated to include this flight information
- **P5:** when the last passenger on a particular flight in the location leaves it then the public display is updated to remove this flight information

5. A stimulus for experience recognition

The physical characteristics of the environment in which the proposed system is embedded are critical to an understanding of the experience that the system will create for users. These characteristics might include the texture of the environment, ambient light and color, the positioning of public displays, the activities that passengers are engaged in (for example pushing luggage trolleys) and how this intrudes on their ability to use mobile phones and look at public displays. Given the potential cost of premature commitment to systems in the target environment how can scenarios and snapshot experiences be used earlier in the development process as a means of understanding the experience that users might have with a proposed design?

5.1. The role of scenarios

Walkthrough techniques such as cognitive walkthrough [18] can be applied to a proposed design in the early stages of the design development. These techniques require sufficient detailed scenario narratives to make it possible to analyze individual actions. In the context of the airport, analyzing actions would involve assessing how effectively the displays and mobile phones resource the actions described in the scenario. Similarly, walkthrough techniques may be used to explore the experience of a proposed system if the analyst can use the scenario to visualize in sufficient detail what the system would "feel like" in its proposed setting. The problem with this approach is that it depends on the imagination of the analyst – what would it really feel like for a particular persona [14], perhaps a frequent flyer who is nevertheless an anxious traveler, to be involved in this story with the proposed design embedded in a given airport. The advantage of using such visualization techniques is that they can be used at very early design stages. A further development would be to ask potential users to visualize scenarios in the context of a description of the proposed design, perhaps using mock-ups of the displays or very approximate, perhaps non-functional, artifacts to help them visualize the scenario in the proposed target environment [6]. Here they would imagine the scenario, perhaps sitting in a meeting room, but would be able to hold or see some of the proposed artifacts that are designed to be embedded in the intended environment. Such a visualization approach is not concerned with the details of the actions involved in the scenarios, rather it would provide an impression of aspects that require further analysis.

Providing an environment in which a "passenger-to-be" can envisage the experience of the proposed technology would involve transplanting working prototypes either to a different context or to simulate the proposed context. For example, some of the features associated with the proposed system are similar to a system designed to provide office commuters with train departure information. To explore this analogy a large display was sited in a common area in the office, and a database was created containing information about workers' railway tickets. A blue-tooth sensor detected the presence of enabled mobile phones in the common area. Relevant information about the departure times of the next few trains was displayed for those people who were in the common room who had railway tickets and were registered with enabled phones. Particular train information was removed from the display when the last commuter for whom the train was relevant left the common room. The system was developed using publish subscribe middleware, by scraping the train destination information from www.livedepartureboards.co.uk. It was then possible to explore how users would experience this environment by configuring their mobile phones appropriately for the trains for which they had tickets and exploring how well the system worked in various situations. The question that such an activity raises is whether much can be learned about the experience of office workers using this system that can be transferred to the airport environment.

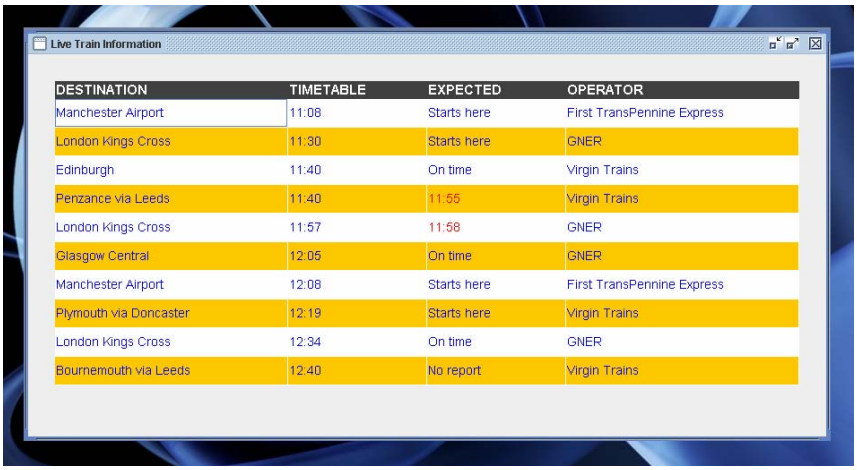

Figure 1: The real train departure display

In reality the two contexts are very different and therefore the experience is likely to be very different. Office workers move out of their workspace to the common area for a cup of coffee or specifically to see whether their preferred train is currently on time. For an air traveler the primary purpose of being in the airport is to travel. They may be working at their laptops or making phone calls but these are secondary activities. Only the most general usability considerations can be ad-

dressed at issues associated with the stability of the display and the way the display is updated.

It is clear that prototyping a similar system (the train information system) in a different setting is not likely to provide much useful information about the experience of the airport. Another possible solution is to explore a simulated environment for the prototype system. A virtual environment was created that bore some resemblance to the office space within a CAVE environment (an alternative that was not explored was to consider the virtual environment on a desk-top to stimulate the experience of the office system with the public display). The departure information was displayed in a virtual room using a virtual display located on one of the walls in the room. The basis of the proposed target system: the sensor software, the use of the publish-subscribe middleware, was the same as the implemented system but it provided a virtual display, and a virtual sensor was triggered by the presence of a real mobile phone in the virtual common room (Figure 2).

There were a number of problems with this approach. Though it had the effect of creating some of the features of the proposed real world it lacked textural realism. In reality common rooms contain people as well as the bustle and noise of these people and their activities. These issues could be crucial to an assessment of the appropriate experience. The CAVE environment made it possible for potential users to explore the virtual space and to see the display from various angles as they would if they were in the real world. To achieve this exploration "natural" mechanisms for navigation around the space are required. A wand was used for navigation in the prototype. In practice this was not satisfactory because it adds encumbrance to the user, potentially interfering with their use of the mobile phone. An alternative approach, currently under exploration is to use body movement as a means of navigation. Another problem with these techniques is that they can provoke nausea in the subject. Simulation sickness can entirely affect the experience of the user in the virtual environment in such a way that its value as a means of exploring experience requirements is compromised.

An alternative approach that would be effective to overcome some of these problems and create an improved simulation of the experience that a user would have is described in [26]. Here "immersive video" is used as a means of exploring features of the design of a system. Their approach uses a video of the existing environment that has been treated using computer enhancement to create the artifacts (for example the public displays) that are proposed. The film represents the scenario. At stages in the scenario the appropriate triggers are generated to modify the subject's mobile phone. The advantage of this technique is that it provides a richer environment with better atmospheric texture including ambient sound and the movement of other people. The approach is called immersive video because all three sides of the CAVE contain video perspectives, though the film is not stereoscopic. The problem with the approach is that the exploration is limited to a fixed sequence. Users have some interaction capabilities with the immersive video and they have more limited means to explore the virtual world that has been created. The filmed scenario constrains where they can move.

A combination of these approaches promises to provide useful feedback on user experience before deployment of the completed system.

5.2 The role of the snapshots

The information that is gathered through snapshot experiences can be used by the analyst to elicit further scenarios. Hence a snapshot can be used as basis for visualizing the experience that the proposed design would create. Alternatively, as illustrated through S1, the comments that are associated with the snapshots can be used as a basis for discovering properties that the design should satisfy such as P1-P2. These properties can be used systematically but informally in the way that usability heuristics [24] are used. Usability inspection techniques typically involve a team of analysts to question the representation of the design. Alternatively, these properties may be made formal and applied to a model of the proposed design as a further stage of the analysis. While satisfaction of the properties is the goal of this formal modeling, counter-examples where the properties fail may provide valuable information that can be used as the basis for further scenarios.

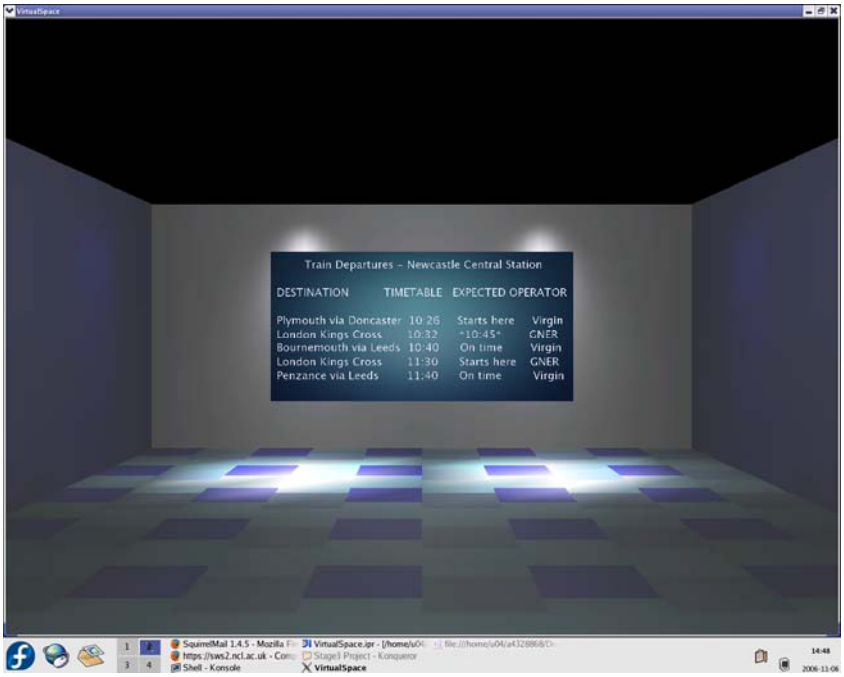

Figure 2: Virtual display of train departure information

6. Modeling the system

Snapshot experiences can be converted into properties to be used as a complement to scenario driven design. Instead of visualizing the design through the scenario, the model of the system is checked to ensure that the property holds. The

approach comes full circle when sequences of states of the model that are generated as a result of properties not being true are themselves used as the basis for further scenarios. Campos, Harrison and Loer [7,20] have explored techniques for using properties to analyze models of interactive systems in the context of usability analysis, in particular the mode complexity of the design. They use model checking techniques to discover whether a property is true in general of the proposed model or to find counter examples that do not satisfy these properties. Model checkers typically generate sequences of states of the model as counter examples. Domain experts can use a bare sequence of states to create a plausible narrative to form the basis for a scenario. This scenario can then be used to visualize the design as described in Section 5.1. This process that combines models, prototypes, snapshot experiences, properties, traces and scenarios is depicted in Figure 3. The figure reflects an iterative process in which models and prototypes are developed in parallel keeping models and prototypes consistent.

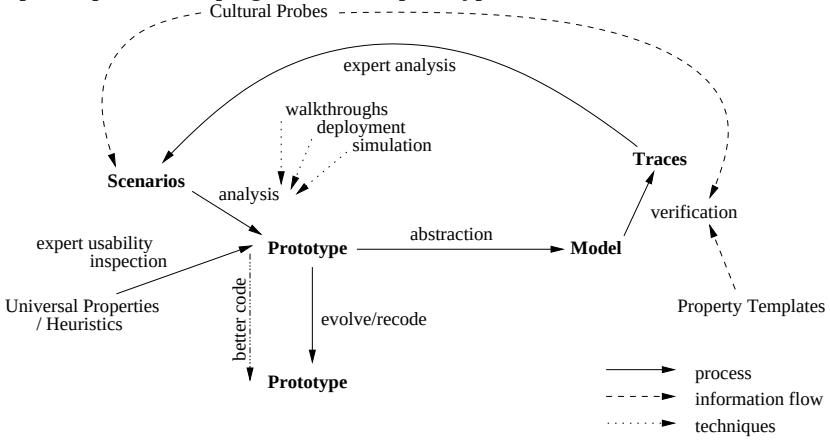

Figure 3 The formal process of experience requirements exploration.

It is possible to enter the diagram from a usability perspective or from a system modeling perspective. Traces are used by specialists to construct scenarios on one side of the diagram and these scenarios are evaluated using prototypes. On the other side of the diagram properties are derived from snapshot experiences and these properties are used to check models of the system. The diagram suggests the variety of evaluation techniques that can be applied to the scenarios and the prototypes.

It is envisaged that a variety of models may be developed to check the properties of the system. The airport model described in this section reflects preoccupations surrounding properties P1-P5 which were in turn based on snapshot experience S1. They focus on timing related properties. An alternative model could have been created to explore possible physical paths in the environment. This model could have been used to analyze properties relating to the snapshot experience S1: “I like to be in a seat in which I can see this display board” and to the snapshot experience S2: “I wish I had better information about how far it was and whether

there were likely to be any delays". Loer and Harrison [19] include location in a model of a plant involving pipes, pumps and valves. They use this model to explore the control of the plant by a hand-held PDA and the potential confusions that arise as a result of location. It is envisaged that a similar model to [19] which employs SMV [23] could be used to model locational characteristics of the airport.

Alternatively a model could be developed to address stochastic properties of the proposed design to address further properties using techniques such as those described by [9, 17]. Examples of properties that might be explored using such models are:

- **P6:** any service that is offered to a subscriber will only be offered if there is a high probability that there is enough time to do something about the service
- **P7:** the message is most likely to be the next message

P6 may have arisen as a result of a comment: "What is the use of being told about a service if there is no time to benefit from the service before the flight departs". P7 on the other hand could be a property generated by the engineer as a compromise, recognizing that the user requirement that it should be guaranteed to be the next message cannot be satisfied in practice.

It is envisaged that generic models could be developed to make the process of construction of models easier. The airport model shares generic characteristics with other ubiquitous systems designs to deploy information about services in built environments (for example systems involving rooms, public displays and sensors). Such an approach is already being used in the analysis of publish-subscribe protocols [3, 11]. The properties may also be based on property templates that are generic forms of frequently occurring snapshot experiences. These templates could be a supported in a way that is similar to that described in [20] in the context of usability properties. The challenge is to develop a model at an appropriate level without unnecessarily biasing the design consideration. The models should allow a proper formulation and application of appropriate properties.

Traces, providing the basis for scenarios, are important in the investigation of experience requirements. However some properties, for example those that relate to quantifiable aspects of the design cannot produce meaningful scenarios. Consider the following:

- **P8:** no matter how many services a user is subscribed to, the flight information service will be dispatched both to the user's device and to the local display within a defined time interval

6.1 Characteristics of the airport model

The model captures the timing behavior of the airport system. It follows previous work [21] on timing aspects of scheduling in a dynamic control system (a paint shop) using `uppaal` [5] to model the interactive system. The airport model contains a process that describes the activity within a room, including the mechanism for sensing the arrival and departure of passengers. This process updates the room based display to show flight information for those passengers that are in the room. A further process describes the passenger that receives specific messages relating

to flight and location in the airport. The passenger moves from room to room. There is also a process that dispatches messages regularly. In what follows a more detailed description of the system will be given.

In the `uppaal` diagrams that follow, circles represent states. States can be named (for example `dispstart`) and can be identified as initial states by two concentric circles. Arcs between states represent possible transitions that can occur. Transitions may be guarded. Hence in Figure 4 one transition from the unnamed state can only occur if the clock `t` is greater than or equal to a value defined by the constant `workload` and the variable `j` is non zero. An arc can specify a communication. Hence `mchan!` is an output signal representing the next message to be sent to waiting processes. This transition can only proceed if there is a process waiting to receive using `mchan?`. A transition can also specify that the state is to be updated.

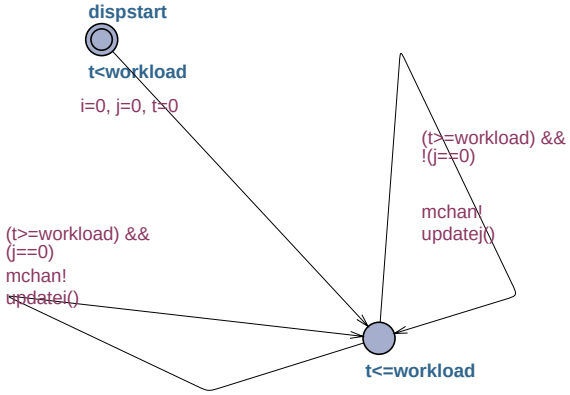

Fig. 4 The dispatcher process

Hence in the arc from `dispstart, i=0, j=0, t=0` specifies that variables `i` and `j` are set to `0` and the clock `t` is also set to 0. Finally, functions may be used to specify more complex updates. In general, for reasons of space, these functions will not be described in detail. In the case of the process of Figure 4, `updatei()` and `updatej()` are functions that among other things update `i` and `j` respectively.

The dispatcher (Figure 4) is critical to the timing characteristics of the design, and alternatives be explored by the designers to create a system that satisfies the required properties. This would involve adjusting the rate and the order of distribution of messages. Alternative dispatchers taking account of passenger arrival volumes should also be considered. The example in figure 4 distributes messages in strict order. Messages relevant to flight and location are sent in sequence. The next message is sent every time interval. The rate of distribution (the variable `workload`) can be adjusted to assess the properties of different rates of distribution. In this process the variable i (describing the flight number) is updated when j (the location value) returns to zero.

Two types of process receive information from the dispatcher. The sensor process (Figure 5) combines the behavior of the public display with the room sensor. The passenger process (Figure 6) describes the passenger and the relevant behavior of the passenger's mobile phone and the mobile device respectively. In the model that was analyzed a sensor was instantiated for each room of the fictional airport (entry hall, queue1, queue 2, check in, main hall, gate). The aim was to ensure that these processes model the key interaction characteristics that are required of the proposed system design insofar as they relate to the properties P1-P5.

The sensor process (Figure 5) describes the key interaction features of:

- the public display located in the room
- the sensor that recognizes the entry and exit of passengers – this assumes an interaction between the sensor and the passenger device

The sensor communicates by means of three channels.

- It receives messages that have been distributed to it from the dispatcher by means of the channel `mchan`.
- It receives requests from the passengers' hand held devices (via `arrive`) where they arrive in the room that relates to the sensor
- It receives requests from the passengers' handheld devices (via `depart`) when they leave the sensor's room.

When the sensor receives a message from the dispatcher, the function `read()` checks the tags on the message and if the location tag coincides with the location of the sensor then the display is updated. Of course a realistic implementation of this system would update a flight information array for display each time a relevant message is received. The array updating mechanism is not of interest to interaction analysis. When the sensor receives a message from the `arrive` channel this signals the entry of a passenger. The array `present`[] keeps a count of the number of passengers present for a particular flight and is incremented with the arriving passenger's flight number. When the sensor receives a message from the `depart` channel then the array is decremented using the departing passenger's flight number. If the result of this is that there are no passengers for a particular flight left in the room then the flight information is removed from the display. In the event that the last passenger moves out of the space the display is cleared. When the passenger is newly arrived in the space then the array `present` is incremented and so next time a message arrives about this flight the information will be displayed for the first time.

The passenger process (Figure 6) describes the activity of the passenger and the key features of their mobile phone. This activity has a number of characteristics:

- The passenger is given a specific path to follow. This is defined in the array `path`.
- The process notifies the room sensor that it has `arrived`. The passenger ticket is updated to point to the current location.
- The passenger moves to a state where it receives messages from the dispatcher via `mchan`. If the received message is tagged with the passenger's current location and the passenger's flight number then the mobile phone display is updated.

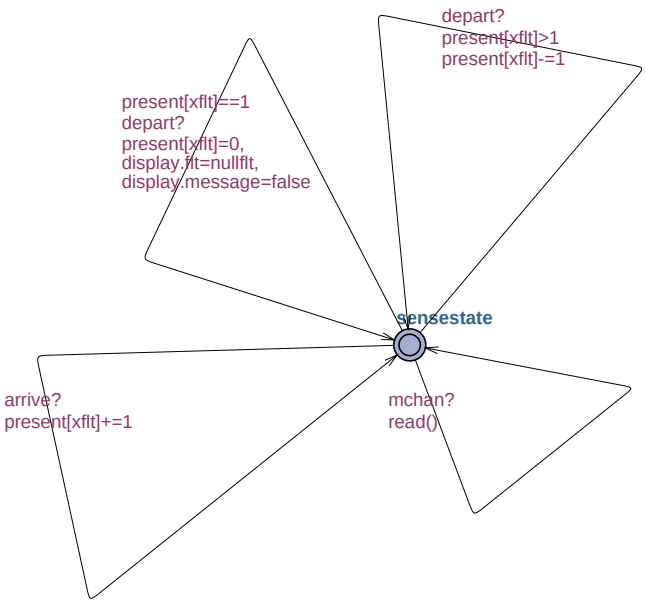

Fig. 5 The sensor process

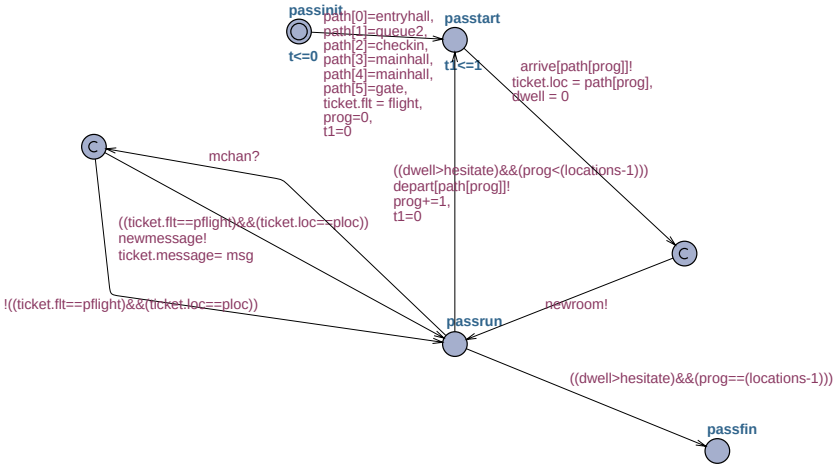

Fig 6: The passenger process

This completes the description of the model. The next stage is to prove properties of the model.

6.2 Checking the properties

The model captures those features of the airport system that relate to properties P1-P5. Space only permits a limited description of the analysis of the system.

P1 requires that "when the passenger moves into a new location in the airport then flight status information is presented to the passenger's hand-held device within 30 seconds." In fact P2 is a property that can be checked in the same way but relates to the sensor rather than the passenger. It must be updated within a period of delay after a passenger arrives. P1 can be characterized as proving that from the point that the passenger enters the location (regardless of flight number) the relevant message will be received by the passenger. Two transitions are of interest in the passenger process (Figure 6). The first occurs as the passenger moves into the new location and the second occurs when the passenger receives a message from the dispatcher that matches the flight number and location of the passenger. This property is checked by introducing an observer process (Figure 7) and adding a communication (`newroom`) in the passenger process (Figure 6) to signal arrival in the new location and similarly a communication (`newmessage`) to signal receiving a relevant message. If the message does not arrive while the passenger is in a location (this time is determined by the variable `dwell`) then the observer will deadlock. Given that `dwell` is the required time interval and the processes accurately reflect temporal aspects of message distributions, deadlock checking can be used to check P1 and P2. When appropriate diagnostics are switched on deadlock generates a trace that can then be further analyzed to work out why the system does not satisfy the properties.

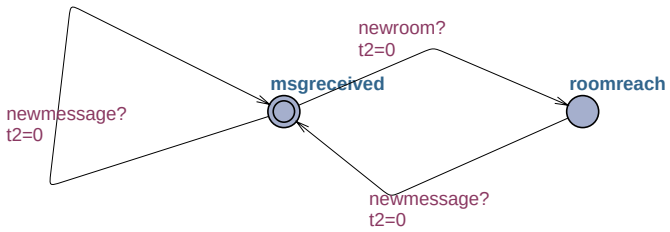

Fig 7: The observer

In practice the generalized deadlock property is very compute intensive and on a no-frills specification PC the uppaal system (UPPAAL 4.0.0 (rev. 1900), May 2006 see http://www.uppaal.com/) ran out of memory after three hours execution. Alternative, more specific properties relevant to P1 and P2 were checked within a minute. For example `A[](o1.roomreach imply (o1.t2<maxdelay))` was checked for different values of `maxdelay`. This property holds true as long as passenger 1 (this passenger's observer is o1) receives a message within `maxdelay` after entering any new room. The airport system that was used for analysis contained two instantiations of the passenger process. As a further elaboration, to check that the passenger received regular updates while occupying a particular room, `A[](o1.msgreceived imply (o1.t2<maxdelay))` was

checked. This property checks whether subsequent messages that are received, while the passenger 1 is in a particular space, arrive at intervals of more than `maxdelay`. This property failed for an appropriate value of `maxdelay` though successful because the passenger had completed its path through the airport and terminated in `passfin`. While the observer o1 continued to wait expecting a further signal from the passenger to say it had received another message the observer's local clock t2 exceeded the `maxdelay` limit.

7. Conclusions

The models illustrated in this paper, taken together with the prototypes that were developed to explore some of the concepts in a virtual environment, have enabled the exploration of experience properties. These techniques can provide early warning of ways in which the system will not create the experience that is the purpose of the design. The model can be used to demonstrate that experience properties (derived from snapshots) are satisfied or fail to be satisfied in specific situations. These situations can be used as a basis for scenarios, used creatively to give early valuable feedback. The paper aims to set these formal processes in the context of other engineering processes that provide early visualization of the design either relying on the user's (or analyst's) imagination or using prototypes in simulated contexts that capture some of the texture of the proposed target environment.

While the proposed approach focuses on the individual's relationship with their environment, it is clear that social aspects of experience are crucial to the success of a design. In the context of the method proposed here these social aspects are captured through the probing of individuals, however it would be envisaged that social modeling would provide additional clarity about how human human interactions contribute to experience and can be supported by the ambient systems. Further techniques would be appropriate for identifying such requirements as discussed in [4]. These considerations are left for future work.

If ubiquitous computing is to become a robust feature of everyday life then engineering techniques such as those described here are required. These techniques will be particularly valuable if it becomes possible to develop generic models for classes of ambient and mobile systems in the style discussed in the context of analysis of publish subscribe systems [3, 11]. In the same way template properties should be developed that frequently occur in experience evaluations and can be instantiated to the specific circumstances of the system being developed. Finally it is to be envisaged that models and prototypes can be developed in synchrony using the style hinted at in [25] and thereby provide coordination between formal models and agile prototypes [1].

Acknowledgement

We thank Jose Creissac Campos for valuable input during the preparation of this paper.

References

[1] Agile working group (2004). The agile manifesto. `http://agilemanifesto.org`
[2] Bannon L (2005) A human-centred perspective on interaction design. In: Pirhonen A, Isomäki H, Roast C, Saariluoma,P (eds) Future Interaction Design. Springer-Verlag, pp 31-52
[3] Baresi L, Ghezzi C, Zanolin L (2005) Modeling and validation of publish / subscribe architectures. In: Beydeda S, Gruhn, V (eds) Testing Commercial-off-the-shelf Components and Systems. Springer-Verlag, pp 273-292
[4] Battarbee K, Koskinen I (2005) Co-experience: user experience as interaction. CoDesign 1:1 pp 5-18
[5] Behrmann G, David A, Larsen K (2004) A tutorial on uppaal. In: Bernardo M, Corradini F (eds) Formal methods for the design of real-time systems. Springer Lecture Notes in Computer Science, 3185 pp 200-236
[6] Buchenau M, Suri J (2000) Experience prototyping. In: Proceedings Designing Interactive Systems (DIS'00), Brooklyn, New York. ACM Press, pp 424-433
[7] Campos JC, Harrison MD (2001) Model checking interactor specifications. Automated Software Engineering 8:275-310
[8] De Nicola R, Latella D, Massink M (2005) Formal modelling and quantitative analysis of KLAIM-based mobile systems. In: Haddad H, Liebrock L, Omicini A, Wainwright R, Palakal M, Wilds M, Clausen H (eds) Applied Computing 2005: Proceedings of the 20th Annual ACM Symposium on Applied Computing. pp 428-435
[9] Doherty G, Massink M, Faconti G (2001) Using hybrid automata to support human factors analysis in a critical system. Journal of Formal Methods in System Design 19(2):143-164
[10] Forlizzi J, Battarbee K (2004) Understanding experience in interactive systems. Designing Interactive Systems (DIS2004), Cambridge, Massachusetts, ACM Press, pp. 261-268.
[11] Garlan D, Khersonsky S, Kim J (2003) Model checking publish-subscribe systems. In: Proceedings of the 10th International SPIN Workshop on Model Checking of Software (SPIN 03), Portland, Oregon.
[12] Gaver W, Dunne T, Pacenti E (1999) Design: cultural probes. ACM Interactions 6(1):21-29
[13] Gilroy SW, Olivier PL, Cao H, Jackson DG, Kray C, Lin D. (2006) CROSSBOARD: Crossmodal Access of Dense Public Displays.. In: International Workshop on Multimodal and Pervasive Services (MAPS06), Lyon, France.
[14] Grudin J, Pruitt J (2002) Personas, participatory design and product development: an infrastructure for engagement. In: Proceedings PDC 2002. pp 144-161
[15] Halnass L, Redstrom J (2002) From use to presence: on the expressions and aesthetics of everyday computational things. ACM Transactions on Computer-Human Interaction 9(2):106-124

[16] Kray C, Kortuem G, Krueger A (2005) Adaptive navigation support with public displays. In St. Amant R, Riedl J, Jameson A (eds) Proceedings of IUI 2005 ACM Press. pp 326-328.
[17] Kwiatkowska M, Norman G, Parker D (2002) PRISM: Probabilistic Symbolic Model Checker. In Computer Performance Evaluation : Modelling Techniques and Tools 12th International Conference, TOOLS 2002. Springer Lecture Notes in Computer Science. No. 2324. p. 200.
[18] Lewis C, Polson P, Wharton C, Rieman J (1990) Testing a walkthrough methodology for theory based design of walk-up-and-use interfaces. In: Chew and Whiteside (eds) ACM-CHI 90. Addison-Wesley pp 235-242
[19] Loer K, Harrison MD (2005) Analysing user confusion in context aware mobile applications. In: Constabile M, Paternò F (eds) INTERACT 2005, Springer Lecture Notes in Computer Science 3585 pp 184-197
[20] Loer K, Harrison M (2006) An integrated framework for the analysis of dependable interactive systems (ifadis): its tool support and evaluation. Automated Software Engineering. 13(4) pp 469-496.
[21] Loer K, Hildebrandt M, Harrison MD (2004) Analysing dynamic function scheduling decisions. In: Johnson C, Palanque P (eds) Human Error, Safety and Systems Development. Kluwer Academic pp 45-60
[22] McCarthy J, Wright PC (2004) Technology as Experience. MIT Press
[23] McMillan K (1993) Symbolic model checking. Kluwer, 1993.
[24] Nielsen J (1992) Finding usability problems through heuristic evaluation. In: Proc. of ACM CHI'92 Conference on Human Factors in Computing Systems. New York. ACM pp 249-256
[25] Niu N, Easterbrook S (2005) On the use of model checking in verification of evolving agile software frameworks: an exploratory case study. In: MSVEIS 2005. pp 115-117
[26] Singh P, Ha HN, Kwang Z, Olivier P, Kray C, Blythe P, James P (2006) Immersive Video as a Rapid Prototyping and Evaluation Tool for Mobile and Ambient Applications. In: Proceedings of MobileHCI'06, Espoo, Finland, 12th-15th September 2006.
[27] Weiser M, Brown J (1995) Designing Calm Technology. http://www.ubiq.com /hypertext/weiser/-calmtech/calmtech.htm, December.