Connecting rigorous system analysis to experience centred design

M. D. Harrison[1], J. Creissac Campos[2], G. Doherty[3], and K. Loer[4]

[1] Informatics Research Institute, University of Newcastle upon Tyne, NE1 7RU, UK
Michael.Harrison@ncl.ac.uk
[2] Department of Informatics, Universidade do Minho, Campus de Gualtar 4710-057 Braga, Portugal
Jose.Campos@di.uminho.pt
[3] Dept of Computer Science, Trinity College Dublin, Ireland
Gavin.Doherty@cs.tcd.ie
[4] Deapartment Strategic Development, Germanischer Lloyd AG - Head Office, Vorsetzen 35,20459 Hamburg, Germany
Karsten.Loer@gl-group.com

Abstract The chapter explores the role that formal modelling may play in aiding the visualisation and implementation of usability and particularly experience requirements in an ambient and mobile system. Mechanisms for requirements elicitation and evaluation are discussed, as well as the role of scenarios and their limitations in capturing experience requirements. The chapter then discusses the role of formal modelling by revisiting an analysis based on an exploration of traditional usability requirements before moving on to consider requirements more appropriate to a built environment. The role of modelling within the development process is re-examined by looking at how models may incorporate knowledge relating to user experience and how the results of the analysis of such models may be exploited by human factors and domain experts in their consideration of user experience issues.

1 Introduction

Ambient and mobile systems are often used to bring information and services to the users of complex built environments such as leisure complexes, hospitals, airports and museums. The success of these systems is dependent on how users *experience* the space in which they are situated in addition to the traditional requirements of usability. They may serve to provide users with an experience of the built environment as a place rather than a forbidding sterile space. They may serve to alleviate the anxiety of travelling in an unfamiliar world. The problem of concern in this chapter is how to reason about such systems so that they satisfy experience requirements and how this framework for reasoning may be integrated with more traditional analyses of interaction with the devices and displays involved. While the chapter is speculative about a set of tools and techniques required to achieve experience centred design it aims to demonstrate the maturity of similar techniques for analysing usability in interactive systems, and extend the debate to the assessment of further qualities in the design of the interactive system.

A focus on experience and ambient and mobile systems provides an important trigger for a fresh look at evaluation in interactive systems. Traditional notions of usability need reconsideration. Ambient and mobile systems have distinctive characteristics that lead to a requirement for special treatment:

- the impact of the environment as the major contributor in understanding how the system should work — its texture and complexity
- the possible role of location and other features of context in inferring user action implicit or incidental in the activities of the user — how natural and transparent this inference is.

A distinction is being made here between the *physical* environment (the walls, windows, notices and the position of the public displays and passengers within this environment) and the software model of context (what the system knows about the physical environment and the user). Context is used to make inferences about action and to supply parameters to services implicitly. Context is updated through interaction with the environment and through noting user characteristics in relation to action and other kinds of preferences. These two features of the real system make it difficult to assess ambient and mobile systems early in the design process. The chapter explores how experience requirements can be related to more rigorous methods of software development.

Although the chapter is mainly concerned with how experience requirements can be gathered and applied in ambient and mobile systems, the techniques described are also relevant to more traditional usability requirements. For this reason the first example focuses on how a method of analysis can be applied to more traditional usability properties. Section 2 discusses methods for eliciting experience requirements and explores the limitations of exclusive use of scenarios and personae. Section 3 explores methods of assessing and evaluating a proposed design against such requirements. As a result the ingredients and requirements for a tool for combining the analysis of usability requirements with experience requirements is developed. Two examples are discussed. The first focuses on usability requirements in relation to a mobile device to support process control (Section 4). The second focusses on experience requirements by considering information flows (Section 5) in an airport system. Finally the chapter sketches a future agenda for completing a tool to support the objectives established.

2 Eliciting and making sense of user experience requirements

Whether particular usability requirements are implemented in ambient and mobile systems is difficult to assess outside the target context that is envisaged for it. This creates a problem because 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. For example, a prototype running in a busy airport will certainly have consequences for passenger satisfaction and therefore commercial consequences if it fails and may have safety consequences if crucial safety information is not provided in a timely way. Methods are needed to enable the implementation of usability and experience requirements and to explore whether they are satisfied in the implemented system. Here it is envisaged that implementation of requirements means that a typical user's experience is consistent with the requirements. Ideally this kind of assessment should be possible realistically before expensive decisions are made.

Two examples are examined, both relating to ambient and mobile systems. The first example is based on mobile hand held technology but will explore the basic technique for evaluation in the context of conventional usability requirements. The second example has more focus on experience and relates to a system designed for passengers in an airport. Two brief scenarios illustrate the second example that will form the basis for the main discussion of the chapter on experience requirements.

– On entry to the departures hall, a sensor recognises the electronic ticket and subscribes the passenger to the appropriate flight while updating the passenger's context to include current position in the departures hall. The flight service publishes information about the status and identity of queues for check in. A message directing the passenger to the optimal queue is received by the passenger's hand-held device because the passenger's context filter contained in the device permits its arrival. This information is displayed on a public display in the departures hall. When the passenger enters the queue a sensor detects entry and adds the queue identifier to the passenger information. As a result different messages about the flight are received by the passenger — this might include information about seating so that the passenger can choose a seat while waiting to check in baggage. This process continues as the passenger progresses through the various stages of embarkation.
– The passenger enters the main hall. The passenger is now additionally subscribed to a retail service. Information about available facilities are received by the passenger according to preferences and flight status.

Eliciting experience requirements for an envisaged ambient system such as this one can be carried out using a combination of techniques, some of which have been developed to deal with the broader class of usability requirements of a system. Firstly stories can be gathered about the current system, capturing a variety of issues relating to usability and to experience, both normal and extreme. Different types of user or persona can be used to explore the particular requirements of user types. The results of this story gathering process are a collection of scenarios that can be used to explore how the new design would behave. They can be used to evaluate the design (see for example [Rosson and Carroll, 2002]), perhaps using a specification of the design or using a rapidly developed prototype.

In addition to scenario orientated techniques for elicitation other techniques are valuable. Techniques such as cultural probes [Gaver et al., 1999] can be used to elicit "snapshot experiences". The elicitation process here involves subjects collecting material: photographs, notes, sound recordings to capture important features of their environment. While these snippets may make sense as part of a story they may equally well be aspects of the current system that are common across a range of experiences or stories.

A further process of probing is described by [Buchenau and Suri, 2000]. Their approach ("experience centred design") involves the construction of prototypes, sometimes very inexpensive and approximate prototypes, which can be used to imagine the experience that users would have with the design. The quality and detail of the prototype 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. In the earliest stages this technique can be used for requirements elicitation, while in the later stages more refined prototypes may be evaluated. To explore and to visualise the proposed design effectively it is important that systems can be developed with agility, trying out ideas and disposing of prototyps that are not effective, using a context that is close to the proposed target environment. They help envision the role of the "to-be-developed" artefact within that work. Prototypes can be used to "probe", that is explore, the validity and representativeness of the scenarios and may lead to alternative or additional scenarios. Testing the prototypes appropriately can help develop an understanding of the experience of the system in its proposed setting.

A formal approach may be combined with the rapid development of prototypes. The modelling approach can be used to generate interesting sequences that form the basis for scenarios or stories that are the focus of further study. The stories can be explored through the animation of the specification but can also be explored through rapid development of prototypes for parts of the desired systems that can then be tested with users. Agile techniques [Agile, 2004,Niu and Easterbrook, 2005] are therefore combined with formal techniques. This approach to combined analysis is particularly valuable if the software framework with which the prototype is developed has a semantic underpinning that will enable the designer to be assured that the prototype is consistent with the properties of the model.

In the settings illustrated by the two examples, there are a combination of ambient displays, kiosks and mobile services for hand-held devices. Facilities and information provided by services is distributed through the built environment by means of hand held devices and public displays making use of context information to infer parameters to supply to the services. They combine together to provide an environment in which passengers can obtain the information they need, in a form that they can use it, to experience the place. Information about the environment relevant to an understanding of this experience might be captured using a combination of cultural probes and scenario analysis. For example, in the case of cultural probes, passengers might be asked to identify those elements in the space that relate strongly to their experience of the airport, perhaps by taking photographs or making audio-video recordings and then by annotating these snapshots. In addition they might be asked to tell stories about situations where they did or did not experience place. The following examples might derive from such elicitation:

- 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";
- photographs of signposts pointing to where my gate is annotated with "I wish I had better information about how far it was and whether there were likely to be any delays on the way";
- 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";

– stories about where the airport helped me to feel aware of what was happening, for example "There has been an incident at Paris airport which means that one of the runways has been closed".
– stories of long and complicated situations that caused me problems. This might involve a long description of how the airport reticketed the passenger to fly to Los Angeles via London, managed the retrieval of baggage and organised checking in again.

Thus an idea can be obtained about how the system works. Further elicitation gathers information about non-central features of the system, capturing stories dealing with other subsets of the facilities and functionality, for example relating to food services, dealing with extreme situations where there are reasons for delay, and where rerouting and reticketing may be necessary. Another story might relate to whether there is enough time to get a meal and whether the meal is vegetarian. Through this process, in the spirit of an agile approach to development, more than one prototype would be developed to explore the different stories, producing segmented functionality — a prototype dealing with flights and flight schedules; a prototype dealing with retail services. Prototypes might be explored, running in-situ using the user stories as the means of testing, exploring the prototype in a simulation of the situation, assessing whether an experience of place is being contributed to. This means that the whole system might be built up using partial prototypes thereby reducing the need to wait until a complete system is available. These prototypes can be explored both from the perspective of user experience and from the perspective of usability.

A limited set of scenarios cannot capture all aspects of the experience of place in the airport. The value of cultural probes is that they provide an orthogonal viewpoint. To achieve an experience of place, the familiar things – for example the constant presence of the notice board – must be captured across scenarios. It is not sufficient simply to focus on scenarios to establish a proper sense of the overall experience as well as the other features of usability of the envisaged system under design. Further exploration may be required to assess and probe how well these static elements of the environment (such as the continually present notice board) are represented across a wider set of behaviours of the design. It is also necessary to investigate the unforeseen consequences of the proposed design. The complexity and interaction between the different components of the system may result in unexpected, emergent properties of behaviours. As a system design evolves, so will the experience associated with using the system. This can contribute to producing a more consistent overall experience, even though the design of the system has emerged in piecemeal fashion.

The physical characteristics of alternative platforms may be important in contributing to the experience of sense of place — frequent flyers may use smart phones, large plasma screens may be placed in the space in a number of different ways. The advantage of using walkthrough techniques is that early exploration may be carried out before the platform is decided and may assist an understanding of whether a particular combination of system components is appropriate.

Given some means of eliciting the significant requirements, the next question is to assess whether a design or implementation will satisfy them.

3 Analysis and evaluation

[McCarthy and Wright, 2004] have argued that while the emphasis within the GUI paradigm has been on technology as tools, the new paradigms require thought about technology we live with (see also [Bannon, 2005]). Elsewhere, this has been characterised as a shift from understanding the use of artefacts to understanding their presence in people's lives [Halnass and Redstrom, 2002]. While user-centred design helps understand the practices and routines into which technologies are expected to fit, they are not as helpful with feelings of resistance, engagement, identification, disorientation, and dislocation. Prototypes can be explored from a variety of perspectives, from a spectrum of usability-engineering evaluation techniques to "experience" explorations through active engagement with prototypes (see [Buchenau and Suri, 2000] and [IST, 2004]). Techniques that are used should be formative and prototypes developed within the simulated scene may be used to stimulate communication and exploration of design ideas as a dialogical process between user, designer and software engineer. A number of techniques may be used to identify experience characteristics of a design.

3.1 Scenario analysis

Scenario analysis [Rosson and Carroll, 2002] can be used at a number of levels to explore the role that the system might play and to evaluate usability and experience issues. Scenarios can be used to capture important characteristics of the environment, either typical uses of the system or "critical incidents" where current arrangements have failed users. They can be analysed by usability engineers to explore how the system would work — what information would be displayed at specific times within the scenario, what actions the user would have to take to obtain further information and so on. Techniques such as cognitive walkthrough [Lewis et al., 1990] and THEA [Pocock et al., 2001] are designed to be used at the action level by usability engineers who have enough knowledge of the environment. Both approaches involve consideration of a sufficiently detailed scenario to be able to consider and visualise the design. While reservations are appropriate in terms of their objectivity [Gray and Salzman, 1998] they are nevertheless of value as a formative mechanism in the hands of designers because they provide feedback that can be used constructively to improve the design.

Scenarios can also be "visualised" by users as they re-experience in their imaginations the scenario in the context of the new design. This might involve the user adopting a persona – a frequent flyer who is nevertheless an anxious flyer. This would not create a detailed account of how the technology works rather it would provide an impression of aspects that require further analysis. Assessing how an artefact contributes to experience requires observation or assessment of the artefact embedded in the proposed situation. Although experience prototypes can be constructed, simulated conditions are required that can deal with realistic scenarios in order that a "passenger-to-be" might visualise the effect that the proposed technology would have and how it would feel to use it. Consider, for example, a system developed to help passengers experience a sense of place at check-in, security screening, passport control and while waiting in the main body of the airport and making use of the many facilities made available to them.

3.2 Alternatives to scenario analysis

Scenario analysis inevitably restricts consideration of the system to the particular situations that are captured by the limited number of narratives that form the basis of the analysis. Issues of coverage are therefore important. In practice, requirements that relate to experience lead to properties that hold true whatever the circumstances and cannot therefore be captured in a limited set of scenarios. Experience level requirements that can be captured specifically for the application in question can be used as probes of a design representation in the same way as usability heuristics [Nielsen, 1992] perhaps using the expertise of a multi-disciplinary team.

[Campos and Harrison, 2001] and [Loer and Harrison, 2006] explore the synergistic role that modelling and scenario based evaluation can play. Properties, formal expressions of usability heuristics, are used to generate traces, that is sequences of actions in the model that serve to demonstrate a situation where the property does not hold. These traces can provide the basis for scenarios. Domain experts can use the bare sequence of actions to create a plausible narrative that can form the basis of a scenario. This scenario can then be subjected to an analysis such as a cognitive walkthrough in order to explore potential problems with the interface to the design. Alternative perspectives can be explored using representative personae. Consider an example of mode confusion. A system is checked for some formal representation of mode confusion and a trace is generated that indicates a circumstance where confusion might occur. This forms the basis for a scenario that is investigated. It is quite possible that although formally there is mode confusion, the interface signals the mode clearly. While a persona representing a newly trained operator will perceive the mode change, an experienced operator is more likely to fail to notice it. This kind of analysis can also be carried out for properties that result from an exploration of the experience requirements of the design. Suppose that a passenger reports that she wants to be able to access up to date flight information wherever she is. An appropriate model might be used to explore possible paths that passengers might take to reach the flight gate and whether up-to-date flight information is always available. This approach is analogous to that taken in [Loer and Harrison, 2005] where a system is explored that controls a process either using a central control room or a hand-held PDA. This will be explored in more detail in the next section.

3.3 Modelling and Prototyping

Formal modelling techniques and agile software development may together have contributions to make to experience centred design as well as in the assessment of more conventional usability. The modelling approach provides the basis for exploring paths to be used by domain experts or usability experts to create narratives that can then be used to explore the experience through an appropriate evaluation technique. The modelling issue is to create a model at an appropriate level of detail to provide a basis for expressing properties and generating sequences. The prototyping issue is to be able to produce systems quickly that can be used to explore the role that the system will play. The modelling and analysis mentioned here is similar to the more rigorous analysis of

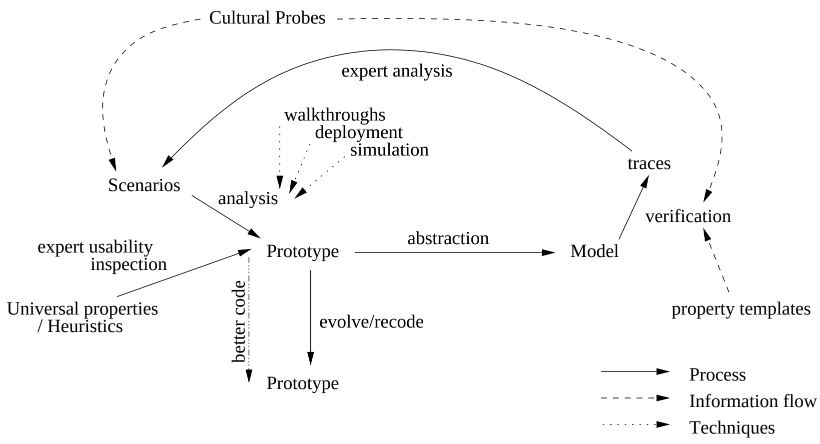

Figure 1. The formal process of experience requirements exploration.

unforeseen consequences that can be carried out using techniques such as model checking. The sequences as scenarios might be visualised through some kind of team based analysis approach based on an animation of the model or by using the model to construct a "throw away" prototype to explore the scenario in some more realistic situation with potential users. A design process is envisaged that is depicted in Figure 1.

The two examples that now follow illustrate a range of properties, capturing features of two types of ambient and mobile system. In the first example the focus of the exploration is the interface with a mobile device where the behaviour of the device is affected by its location within a processing plant. The analysis compares a proposed new interface to the handheld device with the existing control room interface. The focus of the specification in this example is the device design in the context of a model of the process being controlled and its location. In the second example the focus is the model of the broader system in the built environment, concerned with location sensors, passenger devices and public displays, and the means by which information is distributed through the environment.

There are a number of ways in which a software framework could be implemented to support the ideas described here. It could be similar to the Hermes framework of Clarke and Driver [Clarke and Driver, 2004] for example which incorporates a model of context ("trail") that comprises a history of the user's activity and can be used as a basis for inference. The trail itself could be used to control the activities of the user. The software framework could be based on a communications protocol such as publish-subscribe [Eugster et al., 2003].

4 Properties of interactive devices

The system concerns the operator interface to a process control system from a centralised control room (see Figure 2) as well as an alternative hand-held device (see Figure 3) [Nilsson et al., 2000]. A limited subset of information and controls for these components is “stored” in the hand-held device to ease access to them in the future – analogous to putting them on the desktop. These desktop spaces are called *buckets* in [Nilsson et al., 2000]. The operator can view and control the current state of the components when in their immediate vicinity. Context is used in identifying position of an operator, checking validity of a given action, inferring an operator’s intention, checking action against an operator’s schedule, while assessing and indicating the urgency of these actions.

In this type of system, context confusions can be avoided through design by changing the action structure (for example, using interlocks) so that these ambiguities are avoided, or by clearly marking the differences to users. There are a variety of other properties that could be considered here including experience requirements, for example a requirement of the system is that it should provide an experience that enhances the safe operation of the system. Requirements associated with such experience criteria might include the requirement for information that provides overall situation awareness about the plant, or ready access to current data trends to give confidence that system is running smoothly. The analysis that proceeded takes the exploratory approach described in the previous section to scrutinise “interesting” traces. An analysis is now described in which questions are articulated in LTL (Linear Temporal Logic) via a number of templates designed to make the formulation of properties in LTL easier for human factors engineers. The properties are used to check models using the SMV model checker [McMillan, 1993]. Details of the specifications and of the mechanism for formulating properties using templates and the link with SMV is described in [Loer and Harrison, 2006]. They are omitted here because the emphasis is on the process of analysis rather than the detailed specification.

The hand-held device, the control room and the plant were modelled using Statecharts [Harel, 1987]. A requirements process might plausibly have generated top level requirements for the interactive system controlling the plant: (1) to inform the operator about progress; (2) to allow the operator to intervene appropriately to control the process; (3) to alert the operator to alarming conditions in the plant and (4) to enable recovery from these conditions.

The plant involves tanks, pipes, valves and pumps that feed material between tanks. The tanks are designed to be used for more than one process and, in order to change processes, a tank must be evacuated before material can be pumped into it. In order to achieve this some of the pumps are bi-directional. The functioning of the plant, the flows and evacuations, can be expressed as a simple discrete model so that the significant features of the environment can be explored. This is discussed in more detail in [Loer and Harrison, 2006]. The model of the plant captures the characteristics of the plant in the simplest terms consistent with its relevance to the actions and displays provided in the control room and the handheld device. Hence the state of the tank is simply described as one element of the set $\{full, empty, holding\}$ — there is no notion of quantity or volume in the model. This is a minimal model that will allow analysis to take account of the physical consequences of the system.

The control room, with its central panel, aims to provide the plant operator with a comprehensive overview of the status of all devices in the plant. Situation awareness is considered to be critical to the operator’s work in the system — in experience terms

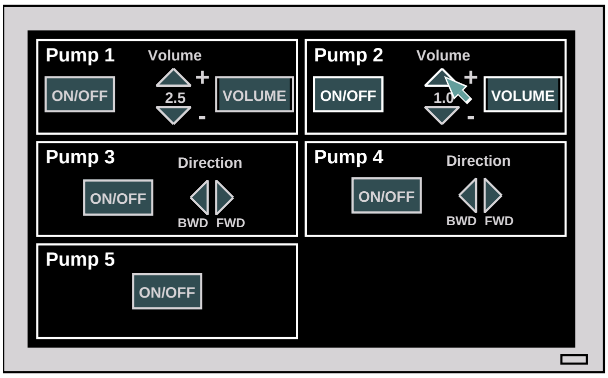

Figure 2. Control Screen layout.

the operator needs to know that they can see everything that is going on. Availability and visibility of action are therefore seen to be primary concerns. For this reason a model of the interface is chosen that focuses on these aspects of the design. Other models could also have been considered to focus on other facets, for example alarms or recoverability. The control panel is implemented by a mouse-controlled screen (see Figure 2). Screen icons are both displays and controls at the same time — clicking on an icon will have an effect. These features of the design are all modelled, showing when icons are illuminated and when actions trigger corresponding actions in the underlying process. The Statechart here builds a bridge between actions that relate to the behaviour of the process underneath and actions performed by the user, such as using the mouse to point and click at the relevant icons.

The hand-held device uses individual controls that are identical to the central control panel. However there is only a limited amount of space available for them. As a controller walks past a pump she may "save" controls onto the display. While the controls continue to be visible on the display, the pumps relating to the controls can be manipulated from anywhere in the system. The hand-held control device (Figure 3) knows its

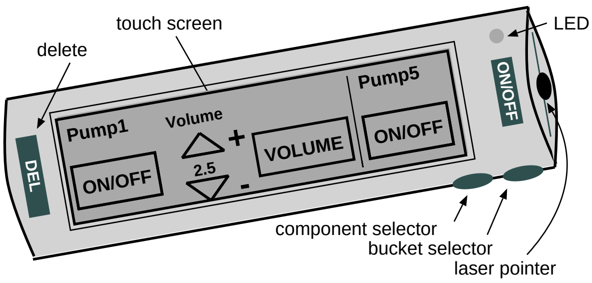

Figure 3. A hand-held control device (modified version of the "Pucketizer" device in [Nilsson et al., 2000]).

position within the spatial organisation of the plant. An area that merits further consideration is the visibility of these displays and the status of saved controls.

By pointing a "laser pointer" at a plant component and pressing the component selector button, the status information for that component and its controls are transferred into the currently selected bucket. Components can be removed from a bucket by pressing the delete button. With the bucket selector button the user can cycle through buckets. The specification of the hand-held device describes both the physical buttons that are accessible continuously and other control elements, like pump control icons, that are available temporarily and depend on the position of the device. When the operator approaches a pump, its controls are automatically displayed on the screen (it does not require the laser pointer). The component may be "transferred" into a bucket for future remote access by using the component selector button. Controls for plant devices in locations other than the current one can be accessed remotely if they have been previously stored in a bucket. When a plant component is available in a bucket and the bucket is selected, the hand-held device can transmit commands to the processing plant, using the pump control icons.

In the case of the hand-held control device the interface to be explored is the device in the context of its environment. The environment in this case is a composition of the tank content model and the device position model. The model presumes that the appliance should always know its location. An alternative approach would allow the designer to explore interaction issues when there is a dissonance between the states of the device and its location. The effect of the type of software architecture used to implement these types of system is to mask the possibility of discrepancy from the implementer.

In order to explore the effect of the difference between the control room and the hand-held device and to generate traces that may be of interest, a reachability property is formulated for a user level "goal" of the system. The goal chosen here for illustration is "*Produce substance C*". This is a primary purpose of the system. The analysis proceeds by making a comparison between traces generated by the alternative models, using domain knowledge and user experience to generate appropriate scenarios. If a property does not hold then the checker finds one counter-example. Alternatively, the negated property may be used to find a trace that satisfies the property. Usually the model checker only produces a single trace giving no guarantee that it is an interesting one from the point of view of understanding design implications. Additional traces can be created by adding assumptions about the behaviour. The technique that is described does not include a model of what it is envisaged that the user should do, constraints are used to narrow down the user behaviours. This contrasts with an approach using explicit tasks (see for example, [Fields, 2001,Loer, 2003]) where the model checker is used to explore a particular way in which the goal can be achieved (the task). So far as this chapter is concerned any behaviours required to achieve a goal are of interest.

The sequences in Figure 4 represent the traces obtained by checking for different models including representing different devices and adding constraints to capture some characteristics of users. In each case the sequence gives one example of how the plant can deliver substance C to the outside world. The property asserts that, eventually, pump 5 will be turned on with tank 1 holding substance C. These sequences provide the basis

Figure 4. Traces generated by runs of the model checker

for the scenarios that the domain expert or human factors expert will use to assess the interaction. A narrative based around the sequence could be used by a potential operator to visualise the experience that they would have using the designed system. This process of visualisation, as mentioned above, may be aided by a process of stepping through the specification using animation tools. Alternatively the scenario generated may be used as the basis for exploring a prototype of all or part of the system. The first sequence in Figure 4 satisfies the control room interface. The second sequence was generated by checking the property against the hand-held device model. While the first two traces assume a serial use of pumps, the third and fourth sequences show the same task for a concurrent use of pumps. Simple comparison of these sequences yields information about the additional steps that have to be performed to achieve the same goal.

As a result of this process, and in particular the comparison, it can be seen that the repetitive process of saving controls may cause slips or mistakes, a direct effect of location on the actions of the hand-held device. While these slips or mistakes may not be dangerous, it may be concluded that the frustration of continually delaying because of omitting actions may be significant. To explore the effect of this a further assumption may be introduced to the property to be analysed, namely that an operator might forget certain steps.

For example, if it is assumed that controls for the pumps are not saved and the original property is checked, the sixth sequence in Figure 4 is obtained. This sequence highlights the likelihood of context confusions as well as user frustrations and therefore the need for the redesign of the device. As can be seen, an identical subsequence of actions at positions 2 and 6 have different effects. An interlock mechanism could be introduced to reduce the frustration caused by forgetfulness. The proposed redesign warns the user and asks for acknowledgement that the currently displayed control elements are about to disappear. The warning is issued whenever a device position is left and the device's control elements are neither on screen nor stored in a bucket. It is straightforward to adjust the model of the interface to the hand-held device to capture this idea, and this specification is given in [Loer and Harrison, 2004]. The design however does not prevent the user from acknowledging and then doing nothing about the problem. Checking the same properties, including the assumptions about the forgetful user, produces Sequences 7 and 8 in Figure 4. In this example the central control panel can be used to identify the key actions to achieving the goal since the additional actions introduced by the hand-held device are concerned exclusively with the limitations that the new platform introduces, dealing with physical location, uploading and storing controls of the visited devices as appropriate. The analysis highlights these additional steps to allow the analyst to judge if such additional steps are likely to be problematic from a human factors perspective. The reasons why a given sequence of actions might be problematic may not be evident from the trace but it provides an important representation that allows a human factors or domain analyst to consider these issues. For example some actions might involve a lengthy walk through the plant, while some actions may be performed instantaneously and some might depend on additional contextual factors like network quality. The current approach leaves the judgement of the severity of such scenarios to the designer, the human factors expert or the domain expert. It makes it possible for these experts to draw important considerations to the designer's attention.

So in summary from the perspective of the diagram (Figure 1), property templates were used to generate traces from a model of the system (the right side of the diagram). Through expert analysis scenarios were identified and as a result alternative options were identified both in terms of user behaviour and by developing alternative models. The piece of the figure that is not dealt with in this example is the rapid development of prototypes based on the consistent software framework.

5 Information arrival

The second case study is concerned with information flows throughout a built environment, hence location and message arrival are important issues. The focus of this case study is concerned more directly with requirements that relate to experience. At the stage of writing this chapter a requirements elicitation had not been carried out in the proposed environment, however the properties that will be considered are plausible in the context of the proposed environment.

A number of different modelling frameworks would be appropriate for capturing different properties of the model and for carrying out verification. Indeed it is envisaged that a toolset for this process would include patterns and guidance about different features of the system and how they should be modelled, and templates representing classes of property that relate to experience requirements. It is envisaged that generic models of ambient and mobile systems will be developed, similar to [Garlan et al., 2003] and [Baresi et al., 2005] who are concerned with generic models of publish-subscribe systems.

Many characteristics of systems, associated with timeliness or likelihood of occurrence, contribute to the experience that we have of them. Such properties require models that incorporate notions of time (the message relating to the flight will be received within a fixed time span) and stochastic models (with a given probability). [Loer et al., 2004] have used `uppaal` models to analyse human scheduling behaviour in relation to process control systems, while [Harrison and Loer, 2006] describes a model in `uppaal` of features of an airport system. [Doherty et al., 2001] have explored stochastic properties of interactive systems and [ten Beek et al., 2006] have used both timed model checking and stochastic model checking to analyse a "groupware system". Properties that are relevant here relate to the dispatching of messages, for example:

1. the message is the next message
2. the message is most likely to be the next message [De Nicola et al., 2005]
3. the message will arrive within 30 seconds [Loer et al., 2004]

Rather than focussing on the modelling and analysis aspects of this example the paper focuses on the properties that would be checked of such a system that are relevant from an experience perspective. The following concrete properties all have characteristics that would improve the experience of the passenger while they are within the built environment. While these properties are not usability properties as conventionally listed they nevertheless capture important features of the user acceptability of a system.

– 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.

- when the passenger moves into a new location then if the passenger is the first from that flight to enter that location, public displays in the location are updated to include this flight information
- when the last passenger on a particular flight in the location leaves it then the public display is updated to remove this flight information
- as soon as a queue sensor receives information about a passenger entering a queue then queue information on the public display will be updated.

The system's failure to adhere to all of these properties does not mean that the system does not perform correctly. The correct information might be passed around the environment but the system would fail to generate the information in the right place at the right time that is needed to maintain the user's experience of the environment. Returning to the process described in Figure 1, model checking these properties of the model will detect traces that require expert analysis and will thereby generate scenarios. These scenarios may then be used perhaps to visualise how different personae would experience them. A potential user might be asked to adopt the persona and then to visualise the system. These scenarios might then be prototyped using the software framework and tested to see whether the undesirable characteristic is likely to modify the experience of a passenger by running the prototype and using a process of visualisation to assess the system.

Further properties of the airport system with an impact on user experience would be more difficult to evaluate by through prototypes using sample passengers. For example:

- 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
- 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 offered
- when the passenger moves into the location then flight status information is presented to the passenger's hand-held device within 30 seconds
- information on public displays should reflect the current state of the system within a time granularity of 30 seconds
- if the passenger enters a location then the passenger's trail will be updated with the action that should occur at that stage (for example screening hand baggage) within an appropriate time (two minutes). If not a reminder of the current activity will be delivered to the user's hand-held
- queue information relating to the best queue to join for a specific flight will be designed to avoid jitter, that is it will be updated sufficiently frequently to improve the experience of passengers but not so frequently that which queue to join and information about how long the delay in the queue changes in a way that is annoying to passengers.

It can be seen that properties such as these will be particularly appropriate to meet passenger uncertainties about flight status, avoid the frustration of jittering information about queues and of being offered services that cannot be received through lack of time.

6 Conclusions

While the primary focus of this paper has been experience requirements, the two examples have illustrated how a range of properties might be explored through a combination of modelling and prototyping techniques. Ambient and mobile systems provide a rich context for the process of requirements elicitation. They challenge our presumptions about how to analyse interactive systems. Two classes of such systems have been considered here. The first was concerned with interaction between a controller and a hand-held controller of the process plant. However, a particularly interesting class of such systems provides the occupants of built environments with a sense of the space — to support a feeling of place and provide access to the services that are offered within the environment. The evaluation of the effectiveness of these systems requires the full richness of the target environment and yet in reality it is not possible for a variety of reasons to explore these systems in a live environment. The possibility that these systems can be explored through a process that involves the use of formal methods has been discussed. Part of this has been demonstrated by reconsidering an analysis that was performed with a more traditional usability perspective. Further examples derived from the specific concerns of an ambient and mobile system in an airport environment.

Formal techniques that can be used to capture abstractly the key features of the prototype currently being developed and can be used as a means of simulation or exhaustive path checking. The model can be developed at the same time as the prototype. Using the model further properties may be appropriate and may be within the scope of the kinds of system we describe. For example, it would be feasible to capture the knowledge that users in the environment might have [Fagin et al., 2004] or the resources for action that are required by users [Campos and Doherty, 2006]. The development of prototypes that support a subset of functions may be accompanied by simple models and simulations in which these prototypes can be explored. So for example, separate models can be developed to reason about the features pertaining to movement through space, and the actions that the user may perform explicitly using the system. Analysis by simulation or model checking can lead to the discovery and exploration of paths that were not envisaged in the original set of scenarios. With the help of domain experts, situations can be envisaged in which the design fails to provide the passenger with the information they need to experience place.

Two important issues underpin our agenda for future research. The first concerns the mapping between models and prototypes and how to maintain an agile approach to the development of prototypes, while at the same time providing the means to explore early versions of the system using formal models. Our concern is to produce generic models that reflect the software architecture used for rapid development and to maintain synchrony between prototype and model. The second concerns the class of models required to analyse the range of requirements that would be relevant to ambient and mobile systems — how to ensure practical consistency between them, and to avoid bias and inappropriate focus as a result of modelling simplifications.

References

[Agile, 2004] Agile, w. g. (2004). The agile manifesto. `http://agilemanifesto.org`.

[Bannon, 2005] Bannon, L. (2005). A human-centred perspective on interaction design. In Pirhonen, A., Isomäki, H., Roast, C., and Saariluoma, P., editors, *Future Interaction Design*, pages 31–52. Springer-Verlag.

[Baresi et al., 2005] Baresi, L., Ghezzi, C., and Zanolin, L. (2005). Modeling and validation of publish / subscribe architectures. In Beydeda, S. and Gruhn, V., editors, *Testing Commercial-off-the-shelf Components and Systems*, pages 273–292. Springer-Verlag.

[Buchenau and Suri, 2000] Buchenau, M. and Suri, J. (2000). Experience prototyping. In *Proceedings Designing Interactive Systems (DIS00)*, pages 424–433, Brooklyn, New York. ACM Press.

[Campos and Doherty, 2006] Campos, J. and Doherty, G. (2006). Supporting resource-based analysis of task information needs. In Harrison, M. and Gilroy, S., editors, *Proceedings 12th International Workshop on the Design, Specification and Verification of Interactive Systems*. Springer Lecture Notes in Computer Science. in press.

[Campos and Harrison, 2001] Campos, J. and Harrison, M. (2001). Model checking interactor specifications. *Automated Software Engineering*, 8:275–310.

[Clarke and Driver, 2004] Clarke, S. and Driver, C. (2004). Context-aware trails. *IEEE Computer*, 37(8):97–99.

[De Nicola et al., 2005] De Nicola, R., Latella, D., and Massink, M. (2005). Formal modelling and quantitative analysis of KLAIM-based mobile systems. In Haddad, H., Liebrock, L., Omicini, A., R.Wainwright, Palakal, M., Wilds, M., and Clausen, H., editors, *Applied Computing 2005: Proceedings of the 20th Annual ACM Symposium on Applied Computing*, pages 428–435.

[Doherty et al., 2001] Doherty, G., Massink, M., and 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.

[Eugster et al., 2003] Eugster, P., Felber, P., Gerraoui, R., and Kermarrec, A. M. (2003). The many faces of publish subscribe. *ACM Computing Surveys*, 35(2):114–131.

[Fagin et al., 2004] Fagin, R., Halpern, J., Moses, Y., and Vardi, M. (2004). *Reasoning about Knowledge*. MIT Press.

[Fields, 2001] Fields, R. (2001). *Analysis of erroneous actions in the design of critical systems*. PhD thesis, Department of Computer Science, University of York, Heslington, York, YO10 5DD.

[Garlan et al., 2003] Garlan, D., Khersonsky, S., and Kim, J. (2003). Model checking publish-subscribe systems. In *Proceedings of the 10th International SPIN Workshop on Model Checking of Software (SPIN03)*, Portland, Oregon.

[Gaver et al., 1999] Gaver, W., Dunne, T., and Pacenti, E. (1999). Design: cultural probes. *ACM Interactions*, 6(1):21–29.

[Gray and Salzman, 1998] Gray, W. and Salzman, M. (1998). Damaged merchandise? a review of experiments that compare usability evaluation methods. *Human Computer Interaction*, 13(3):203–261.

[Halnass and Redstrom, 2002] Halnass, L. and 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.

[Harel, 1987] Harel, D. (1987). Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, 8:231–274.

[Harrison and Loer, 2006] Harrison, M. and Loer, K. (2006). Time as a dimension in the design and analysis of interactive systems. Technical Report CS-TR-XXX, School of Computing Science, University of Newcastle.

[IST, 2004] IST, Advisory, G. (2004). Experience and application research: Involving users in the development of ambient intelligence. Technical report, ISTAG Working Group. Final Report.

[Lewis et al., 1990] Lewis, C., Polson, P., Wharton, C., and Rieman, J. (1990). Testing a walk-through methodology for theory based design of walk-up-and-use interfaces. In Chew and Whiteside, editors, *ACM-CHI 90*, pages 235–242. Addison-Wesley.

[Loer, 2003] Loer, K. (2003). *Model-based Automated Analysis for Dependable Interactive Systems*. PhD thesis, Department of Computer Science, University of York, UK.

[Loer and Harrison, 2004] Loer, K. and Harrison, M. (2004). Analysing and modelling context in mobile systems to support design. Technical Report CS-TR-876, School of Computing Science, University of Newcastle upon Tyne.

[Loer and Harrison, 2005] Loer, K. and Harrison, M. (2005). Analysing user confusion in context aware mobile applications. In Constabile, M. and Paternò, F., editors, *INTERACT 2005*, number 3585, pages 184–197. Springer Lecture Notes in Computer Science.

[Loer and Harrison, 2006] Loer, K. and Harrison, M. (2006). An integrated framework for the analysis of dependable interactive systems (ifadis): its tool support and evaluation. *Automated Software Engineering*. in press.

[Loer et al., 2004] Loer, K., Hildebrandt, M., and Harrison, M. (2004). Analysing dynamic function scheduling decisions. In *Human Error, Safety and Systems Development*, pages 45–60. Kluwer Academic.

[McCarthy and Wright, 2004] McCarthy, J. and Wright, P. (2004). *Technology as Experience*. MIT Press.

[McMillan, 1993] McMillan, K. (1993). *Symbolic model checking*. Kluwer.

[Nielsen, 1992] Nielsen, J. (1992). Finding usability problems through heuristic evaluation. In *Proc. of ACM CHI'92 Conference on Human Factors in Computing Systems*, pages 249–256, New York. ACM.

[Nilsson et al., 2000] Nilsson, J., Sokoler, T., Binder, T., and Wetcke, N. (2000). Beyond the control room: mobile devices for spatially distributed interaction on industrial process plants. In Thomas, P. and Gellersen, H.-W., editors, *Handheld and Ubiquitous Computing, HUC'2000*, number 1927 in Springer Lecture Notes in Computer Science, pages 30–45. Springer.

[Niu and Easterbrook, 2005] Niu, N. and Easterbrook, S. (2005). On the use of model checking in verification of evolving agile software frameworks: an exploratory case study. In *MSVEIS 2005*, pages 115–117.

[Pocock et al., 2001] Pocock, S., Harrison, M., Wright, P., and Johnson, P. (2001). THEA: A technique for human error assessment early in design. In Hirose, M., editor, *Human-Computer Interaction INTERACT'01 IFIP TC.13 International Conference on human computer interaction*, pages 247–254. IOS Press.

[Rosson and Carroll, 2002] Rosson, M. and Carroll, J. (2002). *Usability Engineering: scenario-based development of human computer interaction*. Morgan Kaufman.

[ten Beek et al., 2006] ten Beek, M., Massink, M., and Latella, D. (2006). Towards model checking stochastic aspects of the thinkteam user interface. In Harrison, M. and Gilroy, S., editors, *Proceedings 12th International Workshop on the Design, Specification and Verification of Interactive Systems*. Springer Lecture Notes in Computer Science. in press.