Connecting rigorous system analysis to experience centred design in ambient and mobile systems

M. D. Harrison and C. Kray
Informatics Research Institute, University of Newcastle upon Tyne, NE1 7RU, UK
michael.harrison@ncl.ac.uk,c.kray@ncl.ac.uk

1 Introduction

Ambient and mobile systems are often used to bring information and services to the users of complex built environments. The success of these systems is dependent on how users *experience* the space in which they are situated. Such systems are designed to enable newcomers to appropriate the environment for the task at hand and to be provided with relevant information. The extent to which a system improves the user's experience of such environments is hence important to assess. Such a focus on experience provides an important trigger for a fresh look at the evaluation for such systems but there are other reasons too why traditional notions of usability need reconsideration:

- 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 action and as a result action may be implicit or incidental in the activities of the user — how natural and transparent this inference is.

Due to these and other problems it is difficult to assess ambient and mobile systems early in the design process. For this reason, we are investigating how to relate experience requirements to more rigourous methods of software development.

2 Eliciting user experience requirements

A conclusion that may be drawn from these differences is that the evaluation of such ambient and mobile systems must be carried out in-situ within the target environment, with typical users pursuing typical activities. The problem with this conclusion is that it is usually infeasible to explore the role of a prototype system in this way, particularly due to cost considerations or when failure of the system might have safety or commercial consequences. Eliciting experience requirements for an envisaged ambient system can be carried out using a combination of techniques. For example, it can be valuable to gather stories about the current system, capturing a variety of experiences, both normal and extreme, and visualising the experiences that different types of user or persona might have in the design. The results of this story gathering process will be a collection of scenarios that can be valuable in exploring how the new design would behave. These scenarios can be used to evaluate the design (see for example [11]), perhaps using a specification of the design or using a rapidly developed prototype. Techniques such as cognitive walkthrough [10] or cooperative evaluation, [9], can provide valuable complementary approaches to evaluation based on these scenarios. In addition to scenario orientated techniques for elicitation other techniques are also valuable. Techniques such as cultural probes [7] are used to elicit "snapshot experiences" and complement these more scenario orientated approaches to the establishing and discussing of user requirements. 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.

3 Formal analysis and experience requirements

A question then is how to make sense of these snapshots. Consider the example of a frequent flyer who is nevertheless anxious about missing his flight. One could imagine that he might take a snapshot of the public display and comment that he always looks for a seat where this information is visible. He might also comment: that the flight information relevant to him is not always clearly discernible on the display; that delay information is often displayed late and is not updated so there is no sense of there being any progress.

This information is not captured well by a specific scenario because although one such situation can be captured well, the scenario does not cover all situations — that this information needs to be available whatever "path" the user takes. The paper argues that more formal approaches may play a role here. The challenge is whether it is possible to produce models of systems in such a way that it becomes feasible to explore experience issues in the design of these systems. In particular we claim that an approach used in earlier work that combines scenarios with property checking would be of value.

Techniques such as model-checking may play a role here [3]. This kind of analysis can also be carried out at an experience level. Suppose that a passenger reports that she wants to be able to access "up to date flight information" wherever she is. Different properties of the model could then generate a number of situations that would be at a level to capture the notion of a passenger with hand-held moving through a space past sensors and in the vicinity of public displays. Just as in [8] we might explore, in an appropriate model, possible paths that passengers might take to reach the flight gate.

4 Technical realisation

It is feasible to explore a simplified class of systems with a software architecture to support ease of programming. Such a system will not involve positioning systems or other more sophisticated forms of sensor. In the software framework we envisage a publish-subscribe architecture [5] being used coupled with a sensor interface, public displays and handheld devices as clients, and a model of context to filter published messages. Context would be based on location (closeness to sensors), histories (also called trails [4]) and user preferences. Generic publish subscribe models are a relatively well established area of research [6, 2]. In general these approaches focus on features of the publish subscribe mechanism including:

- reusable model components that capture run-time event management and dispatch
- components that are specific to the publish subscribe application being modelled.

With such models it is possible to explore the integration of sensors and appropriate ways to introduce filters associated with context enabling the exploration of properties such as:

- 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 handhedl 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, then public displays in the location are updated to include this flight information
- when the last passenger in the location leaves it then the public display is updated to remove this information
- as soon as a queue sensor receives information about a passeneger entering a queue then queue information on the public display will be updated

These properties can all be connected to the experience that a user has of the system. The system's failure to adhere to all of these properties will not necessarily mean that the system cannot perform effectively but in some sense or other they may relate to the potential for anxiety or a sense of where the passenger is. Given the style of approach discussed in the previous section checking properties of the model will generate sequences that do not satisfy them. The domain expert will use this information to generate scenarios that are potentially interesting from a user point of view. These scenarios may then be used perhaps to visualise how different personae would experience them. A potential user might be ask to adopt the persona and then to visualise the system. Paper or electronic prototypes would be used to indicate what the system would appear to be like at the different stages of the scenario.

5 Conclusion

In this brief position paper, we argued that experience centred design is of particular importance in the context of ambient and mobile systems, where a user is situated in a dynamic environment. We presented some initial ideas on how to gather experience requirements early on in the development process, and on how to use formal methods such as model checking to identify potential problems related to the user experience. We also sketched how such an approach could be realised technically using publish-subsribe models.

References

[1] E. Aarts and S. Marzano, editors. *The New Everyday*. 010 Publishers, Rotterdam NL, 2003.

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

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

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

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

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

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

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

[9] A. Monk, P.C. Wright, J. Haber, and L. Davenport. *Improving your human-computer interface: a practical technique*. Prentice-Hall, 1993.

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

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