

## Is there a role for rigorous system analysis in experience centred design?

Michael Harrison, Informatics Research Institute, University of Newcastle, Newcastle NE1 7RU, England. michael.harrison@ncl.ac.uk

Gavin Doherty, Dept of Computer Science, Trinity College Dublin, Ireland

Jose Creissac Campos, Department of Informatics, Universidade do Minho, Campus de Gualtar 4710-057 Braga, Portugal

This position statement discusses the possibility that formal modelling techniques and agile software development, normally viewed as antithetical, may both have contributions to make to experience centred design in general and in particular in building systems in which users can experience place. Although the juxtaposition of experience centred design and formal methods may seem bizarre (perhaps the fevered hope of a hapless formalist) we argue that there are characteristics of experience that are not captured well by user stories used in agile design and these characteristics require a more rigorous analysis of unforeseen consequences. We begin by taking the position that agile methods of development are well suited to designing and implementing systems that satisfy non-functional requirements such as experience of place. We note that there are characteristics of experience however that are not well captured by a limited set of user stories. This observation has led to the use of techniques such as cultural probes (Gaver et al, 1999) that are designed to elicit snapshot experiences. Continuous and detailed testing is an important aspect of agile development (Agile, 2004), and the mechanisms for achieving this are effective for building software rapidly. However, such tests are problematic when applied to interface design, and completely ineffective for detecting unforeseen consequences of a design, where these snapshot experiences are crucial. We propose a combination of agile design of software components evolving to the required system with component modelling, simulation and analysis. The aim is to support reasoning about design decisions which impact on the capabilities and resources provided to the user, and their availability and distribution across space and time.

Agile approaches to software development, such as Extreme Programming, are characterised by rapid “throw-away” development of software. Prototypes are developed to reflect requirements as captured by scenarios or user stories. These user stories reflect the important characteristics of the work which the technology is to be designed to support. They help envision the role of the “to-be-developed” artefact within that work. Software is developed rapidly, different scenarios may be implemented concurrently and tested using the user stories as a means of assessing the quality of the prototypes. Test suites based on the user stories may act as measures of quality in a role that is similar to the one played by models / specifications in more formal approaches. While the main emphasis of these approaches is rapid development and equally rapid discarding if necessary, agile development also helps develop appropriate user stories. Although this does not appear to be a particular focus for the technique, prototypes can also be used to “probe”, that is explore the validity and representativeness of the user stories. Testing the prototypes develops understanding of the nature of the context and work or experience that the prototypes are being developed for. They can be used to reflect on the appropriateness of the user stories and produce further characteristics or trigger further stories.

Agile development is about producing (and testing) code. The prototype eventually becomes the actual software. Refinement as understood in the context of formal techniques, described as a rigorous stepwise process, is replaced by an evolutionary process where software that is fittest for purpose emerges. With the experience of developing successive prototypes comes robustness and readiness. This model of development can be seen as matching well with environments which become increasingly technology-augmented over time, rather than designed and deployed as one monolithic system.

Whereas it is often assumed that the user story can be used to provide the basis for a test battery for the prototype, hence simplifying and automating the testing of the prototype, it can also be used to explore software based systems that satisfy non-functional

requirements. If a criterion such as usability (performance) or some characterisation of experience (place) is to be satisfied in the software then quickly developed prototypes can be explored using standard techniques for exploring an artefact in its context in a close-to-everyday setting. These prototypes act as probes and once used and thrown away pave the way for the development of prototypes that capture more richly the requirements of the system. Assessing how an artefact contributes to experience necessarily requires observation or assessment of the artefact embedded in the proposed situation. In order to do this, as “natural” a situation as possible is required. Prototypes embedded within these contexts can be constructed that can deal with realistic scenarios that have the effect of providing this context and can be assessed within these contexts. It is in this sense that agile techniques may be considered as an effective means to develop systems that satisfy less tangible requirements such as that system should satisfy an experience of place.

Consider, for example, a system developed to help passengers experience a sense of place in the unfamiliar setting of an airport. One might imagine a combination of ambient displays, kiosks and mobile services for hand-held devices such as telephones 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 techniques such as “cultural probes” (Gaver et al., 1999) or scenario analysis (Rossen and Carroll, 2004). Cultural probes capture snapshot information, while scenario analysis derives stories relating to the environment. Passengers might be asked to identify those elements in the space that capture ideas of place in 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:

1. 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”;
2. 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”;
3. 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”.
4. stories about where the airport helped me to feel aware of what was happening;
5. stories of long and complicated situations that caused me problems.

The information elicited by these techniques combines significant static information, it might be homing information or security information, with dynamic scenario information that has common characteristics with the user stories. The scenario information fits well with the notion of user story. It becomes possible to capture stories that deal exclusively with subsets of the required functionality, for example one might deal with details of flights, whether there is food, what the reason for the delay is, whether there are any other flights that I can catch that would get me to Los Angeles today. Another story might relate to whether there is enough time to get a meal and whether the meal is vegetarian. Hence in parallel, prototypes might be developed that deal with segmented functionality - a prototype dealing with flights and flight schedules; a prototype dealing with retail services. The prototypes will 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.

The problem is that the scenarios elicited cannot capture all aspects of the experience of place in the airport. The value of cultural probes is that they provide an orthogonal viewpoint. In order 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 exercise the user stories in order to develop a place centred design. Further exploration may be required to assess and probe how well these static elements of the environment are represented across all possible behaviours of the design. It is also

necessary to investigate the unforeseen consequences of the proposed design. As a system design evolves, so will the experience associated with using the system. What we are proposing can contribute to producing a more consistent overall experience, even though the design of the system has emerged in piecemeal fashion.

However, the unforeseen consequences of a developed piece of software are not magically dealt with by lightweight throwaway developments. Problems that arise because of inferences about availability of information at a location for example may occur at the extremes and exhaustive analysis may be needed to find these extremes. The problem with doing this is the time it takes to exhaust all the possible uses of the prototype in order to explore how well the information is provided over all these paths and where the extremes are. Formal approaches to analysis provide modelling 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 it becomes possible to capture, for example, the knowledge that users in the environment might have (Fagin et al, 2004) or the resources for action that are required by users (Campos & Doherty, 2005). 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, models can be developed of the navigation features and the actions separately. Analysis by simulation or model checking can lead to the discovery and exploration of paths that were not envisaged in the user stories. 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.

Finally, a further opportunity is afforded by such modelling activity. In reality the development of ambient mobile intelligence applications might be carried out without a complete understanding of the platforms and software that will be running on these platforms. The physical characteristics of alternative platforms may be important in contributing to the experience 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 models is that early exploration can be carried out before the platform is decided and may assist an understanding of whether a particular configuration is appropriate.

Agile (2004) <http://agilemanifesto.org/>

Campos, J.C. & Doherty, G. (in press) Supporting resource-based analysis of task information needs. Proceedings Designing, Specifying and Verifying Interactive Systems (DSVIS 2005). Springer Lecture Notes in Computer Science.

Fagin, R., Halpern, J.Y., Moses, Y. & Vardi, Y. (2003) Reasoning about knowledge. MIT Press.

Gaver, W., Dunne, T. & Pacenti, E. (1999) Design: cultural probes. ACM Interactions Volume 6 Issue 1.

Rosson, M.B. and Carroll, J.M. (2001) Usability Engineering: Scenario-based Development of Human-Computer Interaction. Morgan Kaufmann Publishers