

Making Ethnography Accessible: Bringing Real-World Experience to HCI Designers and Software Engineers

Ian Sommerville

Computing Department, Lancaster University, UK

is@comp.lancs.ac.uk

Abstract

This short paper proposes that bringing real-world experience through field studies of work and other environments where systems are used is a means of bridging the gap between software engineering and HCI. We argue that all disciplines involved in systems development can benefit from field studies and observations of users at work and at home and we briefly discuss a range of techniques that we have developed to facilitate these field studies.

1. Introduction

Simplistically, the primary preoccupation of software engineering has been the translation of a system specification into an executable system. Hence, the software engineering research community has focused on tools and techniques to facilitate this process – means of expressing specifications, approaches to system design, tools for programming support, testing methods, etc. Equally simplistically, the focus of user interface designers is to design an interface that provides a way for system users to interact with the functionality defined in the specification. Hence, the HCI research community has been concerned with methods of analyzing and learning from existing interfaces, methods and processes for interface design, representation and visualization techniques, etc.

Both of these communities focus on the ‘how’ rather than the ‘what’. The HCI community on how to interact and the software engineering community on how to deliver functionality. For some types of system such as large aerospace systems, the ‘what’ is the concern of systems engineers; for others, the responsibility for what the system is intended to do depends on business analysts, project managers or end-users who are already busy enough with their everyday work.

This is not to suggest that either software engineers or UI designers are narrow specialists who do not have an interest in what the system should do. They are often keenly interested but are frustrated by organizational or technical barriers that limit their access to users and requirements and hence do not allow them to acquire the

broad knowledge of the purpose of the system, its role in an organization, the constraints on its development, etc.

In this short paper, I argue that the way of bridging the gap between software engineering and HCI and to improve the design process for both software and user interfaces, is to bring descriptions and models of the real-world of system use to both disciplines. To do so, we propose that we need to adapt and extend methods of social analysis such as ethnography so that all participants in the design of a system can quickly understand and relate to the practical contingencies of how computer-based systems are used.

2. The importance of field studies

An increasingly widely used approach to software system specification is based on use-cases where the specification is expressed as a collection of interaction descriptions with a system. Therefore, in an electronic patient record system (say), there will be a use-case describing the creation of a patient record, a use-case describing how x-rays are integrated with other information, a use-case describing how primary care practitioners are informed of hospital treatment, etc. At one level, this is an encouraging development as it clearly indicates to the software engineers involved in the development of the system that interaction is taking place and is important. Equally, the set of use cases defines the types of interaction that take place, are often derived from discussions with system users and so can be used as a starting point for the user interface design.

However, the practical realities of work are far from the simple and often elegant models presented in a use case description. In real working and domestic environments, people have many concerns apart from interacting with some system and are inevitably multi-tasking – doing other things as well as using the system. To continue the EPR example, the use-case for a nurse updating a patient will record the steps involved in making the changes to the record but will (unless remarkably well-designed) not take into account the fact that the nurse may be interrupted in this process for various reasons ranging from patients’ relatives seeking

information to crises where the nurse has to respond to a medical emergency.

One such system that we observed required that, before updating a record, the nurse had to logon to the system – an apparently sensible security precaution. It also had a timeout feature where after a period of inactivity, the system logged out the user – again, a security feature that reduced the probability of unauthorized access. However, if the user was logged out before completing inputs then all inputs were lost and had to be restarted. In the environment where this system was used, interruptions were normal and in many cases nurses found that they were continually having to re-input data because their partially completed data had been lost. After a very short time, all nurses resorted to maintaining records on paper and, when they had time, inputting these to the system.

The system designers here had clearly spent time in discussing with system users how such a system would normally be used and the user interface design made it fairly easy to input patient information. The problem here was that the designers had no knowledge of where the system would be used and what else the system users were doing. Had they spent even a short time observing nurses at work, they would have realized that their proposed designs were impractical.

Of course, the importance of field studies has been recognized by both the requirements engineering and the HCI research communities for many years. Suchman [1] was the first to discuss the value of ethnographies of work in the 1980s and, since then, there have been a many studies reported of lessons learned from ethnography [2-6]. By and large, however, these field studies have involved ethnographers with a background in the social sciences who have spent time observing and documenting work.

However, in spite of the recognition in the research communities of the value of ethnographic studies, the practical impact of this approach has been minimal. A recent survey of the state of the art in requirements engineering (**ref**) showed that ethnography was not used in any of the project surveyed and this is in accord with our own experience. Fundamentally, the reasons for this are lack of expertise and lack of time and resources – it takes considerable time for ethnographers to carry out field studies and report their conclusions to systems designers.

We are now convinced that to improve the design of the software and the user interfaces of systems, it is essential for designers to spend time in the field observing users at work and developing an understanding of how they interact with a range of systems, how they cooperate with other people who may also be system users and how they organize and structure their work. If both user interface designers and software engineers are involved in this process, they develop a shared understanding of the

goals of the system and hence communication between these disciplines is facilitated.

However, it is not simply a question of designers visiting a workplace for a short time and watching what is going on. Developing an understanding of a system takes a considerable time and professional ethnographers have developed a range of techniques for helping them develop this understanding. We believe that it is important to draw on these (sometimes implicit) techniques and to make them accessible to other disciplines. Only then will it be possible for designers to effectively analyse a setting where a system is to be used and to develop a deeper understanding of what the system has to do and how the users may interact with this functionality.

3. Making ethnography accessible

Our experience of ethnographic studies of work started in the early 1990s with studies of air traffic control [7, 8] and later studies in financial institutions [9]. Our focus then was on the problems of communication between designers and ethnographers and we developed an informal design tool [10] to help present these results. Later work focused on the development of a presentation framework that was designed to structure and organize an ethnographic record for discussions with systems designers [11]. This framework was based on presenting the record from three perspectives:

1. The setting of the work. This is a description of the physical layout of the setting and how physical artifacts are used by people in that setting.
2. Social and organizational perspectives. This is a set of illustrative vignettes that describe how work gets done, how people cope with exceptions, etc.
3. Work flow. This is a description of the division of labour in a setting, the ways that artifacts are used at different stages in the work, the flow of work from one team member to another, etc.

Even at this stage it was clear that there were real practical difficulties of involving ethnographers in the systems design process. The work could be presented to designers but the differences in language and culture between them and sociologists involved in the ethnography led to communication difficulties. We therefore moved on from the work on presentation to incorporate it into a more structured approach called COHERENCE [12, 13] where we explored how this presentation framework could be used by non-ethnographers to help organize observations and to develop use-cases defining the work.

In the COHERENCE approach, we started out with the presentation framework and investigated how this could be adapted and used by designers themselves who were involved in field studies. We were also concerned with integrating the results of field studies with other design methods and looked at using structured notions

such as use-cases to document the field studies rather than the narrative text that is used by ethnographers.

The fundamental problem that we encountered in the COHERENCE approach was that designers found it very difficult to get started. Although they could appreciate the descriptions of the different viewpoints, they found it difficult to relate the practical observations that they could make with the structured framework (which was originally designed for use by professional ethnographers) and to separate the essential from the accidental. Ethnographers are very good at understanding what are essential characteristics of a task and what are surface characteristics; software designers found this to be very difficult. It was therefore hard to relate the observations made to the systems design decisions.

Our recent work therefore has focused on how to reuse knowledge from field studies and we have developed a number of Patterns of Interaction that encapsulate and illustrate various situations and the design solutions that have been developed in such situations [14-16].

A pattern of interaction is a pattern that we have observed in at least 2 field studies. A pattern description includes a narrative description of the pattern and its general form and a number of vignettes illustrating actual instantiations of the pattern. Critically, we also include a discussion of design implications and discuss how designs have been realized in different settings. Currently, we have about 10 patterns that can be accessed through the patterns of interaction web site¹.

The key advantage of the patterns approach is that it gives a designer something to relate to when they are involved in field observations. They can see the existing work situation and, in many cases, link these through to pattern descriptions. Currently, we are working on how the patterns collection can be the basis for a process to support fieldwork by designers who then go on to develop software and use interface designs.

4. Bringing together HCI and software engineering

User interface designers and software engineers have a shared goal of developing a high-quality system that is usable and used. Cultural differences between the disciplines have led to communication problems and, in particular, the specification-driven approach of software engineering has conflicted with the experimental, exploratory approach of UI design. To some extent, agile methods such as extreme programming have helped to bring these approaches together but, in reality, these agile methods that involve user interaction with the development team have limited applicability.

They cannot be used cost-effectively for large systems with different contractors and sub-contractors and many

different stakeholders. They also require on the involvement of a 'typical' and committed user and our limited experience of these approaches suggests that it is difficult to find users who have the time and interest to be involved directly in software development. Indeed, a similar problem applies to approaches such as participatory design.

We believe that the most effective way to support UI and software designers working together is to focus their attention round the work (or the play) that the software is intended to support. This helps establish a common vocabulary and understanding of what is required and provides a frame of references to help make the design trade-offs that inevitably arise in any complex system.

10. References

- [1] Suchman, L., *Plans and Situated Actions*. 1987, Cambridge: Cambridge University Press.
- [2] Harper, R.H.R., *Looking at Ourselves: An Examination of the Social Organisation of Two Research Laboratories*, in *Proceedings of ACM CSCW'92 Conference on Computer-Supported Cooperative Work*. 1992, ACM Press: Toronto, Canada. p. 330-337.
- [3] Hughes, J., et al., *Designing with Ethnography*. Interacting with Computers, 1993. **5**(2): p. 239-53.
- [4] Heath, C. and P. Luff. *Documents and professional practice: 'bad' organisational reasons for 'good' clinical records*. in *Proceedings of the ACM 1996 Conference on Computer Supported Cooperative Work: CSCW'96*. 1996. Boston: ACM Press.
- [5] Heath, C., et al. *Unpacking collaboration: the interactional organisation of trading in a city dealing room*. in *ECSCW'93*. 1993. Milan.
- [6] Rouncefield, M., *An ethnography of 'everyday admissions work'*. 1998, Lancaster University: Lancaster, UK.
- [7] Bentley, R., et al. *Ethnographically-informed Systems Design for Air Traffic Control*. in *CSCW'92*. 1992. Toronto, Canada: ACM Press.
- [8] Sommerville, I., T. Rodden, and P. Sawyer, *Cooperative Systems Design*. *Comp. J.*, 1994. **37**(5).
- [9] Blythin, S., M. Rouncefield, and J.A. Hughes, *Never mind the ethno stuff—what does all this mean and what do we do now?: Ethnography in the commercial world*. *Interactions*, 1997. **4**(3): p. 38-47.
- [10] Sommerville, I., et al. *Integrating ethnography into the requirements engineering process*. in *RE'93*. 1993. San Diego CA.: IEEE Computer Society Press.
- [11] Hughes, J., et al. *Presenting Ethnography in the Requirements Process*. in *Proc. RE'95*. 1995. York: IEEE Computer Society Press.
- [12] Viller, S. and I. Sommerville, *Coherence: Social Analysis for Software Engineers*. 1998, Lancaster University: Lancaster, UK.
- [13] Viller, S. and I. Sommerville, *Coherence: an approach to representing ethnographic analyses in systems design*. *Human-Computer Interaction*, 1999. **14**: p. 9-41.

¹ <http://polo.lancs.ac.uk/patterns>

- [14] Martin, D. and I. Sommerville, *Ethnomethodology, Patterns Of Cooperative Interaction and Design*. ACM Trans. on Computer-Human Interaction, April 2004. To appear.
- [15] Martin, D., et al. *Finding Patterns in the Fieldwork*. in *Proc. ECSCW'01*. 2001. Bonn: Kluwer.
- [16] Martin, D., M. Rouncefield, and I. Sommerville. *Applying patterns of interaction to work (re)design: e-government and planning*. in *Proc CHI'2002*. 2002: ACM Press.