Available online at www.sciencedirect.com
ScienceDirect
Int. J. Human-Computer Studies 67 (2009) 437–454
International Journal of Human-Computer Studies
www.elsevier.com/locate/ijhcs

Engineering the social: The role of shared artifacts

Jeni Paay[a], Leon Sterling[a,*], Frank Vetere[b], Steve Howard[b], Anne Boettcher[a]

[a]Department of Computer Science and Software Engineering, University of Melbourne, 111 Barry St. Carlton, Victoria 3058, Australia
[b]Department of Information Systems, University of Melbourne, 111 Barry St. Carlton, Victoria 3058, Australia

Received 30 June 2008; accepted 11 December 2008
Communicated by C. Sierra
Available online 24 December 2008

Abstract

This paper presents a multidisciplinary approach to engineering socio-technical design. The paper addresses technological design for social interactions that are non-instrumental, and thereby sometimes contradictory or surprising and difficult to model. Through cooperative analysis of cultural probe data and development of agent-oriented software engineering (AOSE) models, ethnographers and software engineers participate in conversations around shared artifacts, which facilitate the transition from data collected in a social environment to a socially oriented requirements analysis for informing socio-technical design.

To demonstrate how this transition was made, we present a case study of the process of designing technology to support familial relationships, such as playing, gifting, showing, telling and creating memories. The case study is based on data collected in a cultural probes study that explores the diverse, complex and unpredictable design environment of the home. A multidisciplinary team worked together through a process of conversations around shared artifacts to cooperatively analyze collected data and develop models. These conversations provided the opportunity to view the data from the perspective of alternative disciplines that resulted in the emergence of novel understandings and innovative practice.

The artifacts in the process included returned probe items, scrapbooks, videos of interviews, photographs, family biographies and the AOSE requirements models. When shared between the two communities of practice, some of these artifacts played important roles in mediating discussions of mutual influence between ethnographers and software engineers. The shared artifacts acted as both triggers for conversations and information vessels—providing a variety of interpretable objects enabling both sides to articulate their understandings in different ways and to collaboratively negotiate understandings of the collected data. Analyzing the interdisciplinary exchange provided insight into the identification of bridging elements that allowed 'the social' to permeate the processes of analysis, requirements elicitation and design.

Keywords: Cultural probes; Agent-oriented software engineering; Shared artifacts; Ethnography; Requirements engineering; Social technical environment

1. Introduction

Despite best efforts, contemporary technologies often fail to meet basic human needs and desires (Bell et al., 2005; Christensen, 1997; Haines et al., 2007; Norman, 1999). Recent developments in design processes have ensured technologies are generally accurate, reliable and usable (Sharp et al., 2007). However, meeting these measurable requirements and qualities constitutes only part of what it means to design technology for people. As social beings we often have loftier needs, such as to experience social connection and empathy, to care for others and be cared for, and to share pleasure. These particular types of social requirements cannot be easily reduced to functional specifications for information provision. In existing software development processes, these social requirements are often neglected or trivialized (Sommerville, 2007). We believe it is valuable to match socially oriented user studies with requirements elicitation methods that are able to identify and document social requirements in a form compatible with existing software engineering methods. Technology in social settings will be of increased value if it demonstrably addresses and fulfills the often ephemeral,

*Corresponding author. Tel.: +61 3 8344 1404; fax: +61 3 9348 1184.
E-mail address: leonss@unimelb.edu.au (L. Sterling).

1071-5819/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ijhcs.2008.12.002

and hard to measure, felt needs of people in these environments.

The disciplines of Software Engineering (SE) and Human Computer Interaction (HCI) share the goal of effective technology, but differ in their theories, methods, and terminology in interpreting 'effectiveness'. This difference creates a communicative divide, which is accentuated for technological innovation that focuses on socially complex situations. In our research, we used a multidisciplinary approach to engineering socially oriented software systems. This approach allows us to combine social understanding of technological use in a human context, extracted using ethnographically informed HCI methods, with SE knowledge and experience of modeling user requirements for software design. In bringing together these two Communities of Practice (Wenger, 1998) we are confronted with similar issues of communication and translation faced by HCI research for well over a decade (Constantine et al., 2003; Cunningham and Jones, 2005; Hughes et al., 1994; Kjeldskov et al., 2006; Viller and Sommerville, 2000). These issues concern how to 'bridge the gap' between ethnography and software engineering for the purposes of designing technology.

In this paper, we investigate facilitating interpersonal interactions between individuals with technology in the home, where social activity is not easily conceived in terms of tasks and goals. We are interested in non-instrumental activities, or activities that cannot be easily decomposed into tasks and sub-tasks; and where the purpose is not necessarily to achieve a goal but to participate in a process. This is illustrated by game playing. Rather than specifying the 'rules' and 'interaction style' necessary for winning, we are interested in the mechanisms that facilitate less instrumental outcomes such as 'engagement' and 'social-bonding'. Clearly 'rules' are not orthogonal to 'social-bonding', but addressing one does not engender the other. Our motivation is to support the non-instrumental characteristics, which may be achieved via any one of a myriad of concrete goals.

We acknowledge that non-instrumental activities also occur in the workplace and are often embedded within purposeful tasks. However, the domestic environment provides more acute and intangible instances of them. While it is true that purposeful work gets done at home, it is the activities that remain when work is abstracted out of family life that we find particularly interesting.

Social requirements obtained through ethnographically informed HCI methods are generally not in an appropriate form for simply feeding into traditional software analysis methods. In our case, cultural probes were used to provide access to people's daily interactions in the domestic setting. These interactions are difficult to study using traditional empirical techniques such as questionnaires, focus groups and participant observation. The data gathered using probes is fragmentary and unstructured, and in the absence of any proven method, the process of translation from probe data to the abstract generalization required in software design is not an easy one.

In our project, we created software requirements models with the agent-oriented software engineering (AOSE) methodology ROADMAP (Juan et al., 2002; Kuan et al., 2005) from the cultural probe findings. We were particularly interested in testing ROADMAP's ability to represent non-instrumental social requirements. This is because its notation extends beyond functional goals, used in traditional software engineering methodologies, to include a special type of goal called a *quality goal*. Quality goals are essentially non-functional and are designed to encapsulate social aspects of the context into the software requirements model, thus providing a mechanism to carry social aspects through to the implementation phase. Identifying quality goals became an important part of the requirements elicitation phase for capturing social requirements from the probe data.

The process of translation was enacted in team meetings, where members of a multidisciplinary team worked together to analyze probe returns and identify quality goals, while creating and maintaining their own representations of understanding. Because the two communities of practice involved came to the table with different values, practices, orientations to technology, and commitments to the process, they achieved this analysis through exchanging thoughts, interpretations and understandings in a series of conversations mediated by a collection of artifacts. These artifacts had different purposes, qualities and affiliations, and were used to cross and negotiate boundaries (Lee, 2007; Star and Griesemer, 1989) between the two groups working within an 'artefact ecology' (Vyas and Dix, 2007) consisting of different digital and physical artifacts, the members of the multidisciplinary team, and their work practices and values. The role that the shared artifacts played facilitated both communication between disciplines and the embodiment of interactions and work coordination that such artifacts play in meetings generally. Coordination was primarily achieved by conversations around artifacts, and it was these conversations that team members found the most enlightening part of the exchange. The conversations enriched their own understandings of the design situation as unanticipated viewpoints emerged; exploiting the complementarities of the different value sets and approaches to design of the two communities of practice.

Artifacts are a powerful resource for analysis, they "tell a story to the extent that they invoke stories" (Ramduny-Ellis et al., 2005, p. 77). They can represent the understandings of one individual and also be used to mediate and negotiate work in collaborative settings (Vyas and Dix, 2007). By analyzing the attributes of the artifacts that made them function as useful shared objects, or not, we can better understand the role that shared artifacts played in the process of translation of ethnographic understanding to abstracted design model.

This paper sits in the territory of the relationship between ethnography and software engineering, and we ask the question: *what is the role of shared artifacts in supporting multidisciplinary teams in engineering the social?*