Computer Supported Cooperative Work

For other titles published in this series, go to
http://www.springer.com/

Alex Voss • Mark Hartswood • Rob Procter •
Mark Rouncefield • Roger Slack • Monika Büscher
Editors

Configuring User-Designer Relations

Interdisciplinary Perspectives

Springer

Editors
Alex Voss
University of Manchester, UK
alex.voss@ncess.ac.uk

Mark Hartswood
University of Edinburgh, UK
mjh@inf.ed.ac.uk

Rob Procter
University of Manchester, UK
rob.procter@ncess.ac.uk

Mark Rouncefield
University of Lancaster, UK
m.rouncefield@lancaster.ac.uk

Roger Slack
University of Bangor, UK
r.slack@bangor.ac.uk

Monika Büscher
University of Lancaster, UK
m.buscher@lancaster.ac.uk

Computer Supported Cooperative Work ISSN 1431-1496
ISBN: 978-1-84628-924-8 e-ISBN: 978-1-84628-925-5
DOI: 10.1007/978-1-84628-925-5

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Control Number: 2008936631

Printed on acid-free paper

Springer Science+Business Media
springer.com

Contents

Contributors

John Bonner has worked for over 20 years in the fields of ergonomics, product, and interaction design in both academic and commercial environments. Much of his research work has involved the use and implementation of user-centred design methods for emerging interactive domestic technologies. He is currently investigating the development and evaluation of foresight-based tools to assist digital media organisations to design new, innovative products and services.

Monika Büscher is a senior lecturer at Lancaster University, working within Sociology, ImaginationLancaster and the Centre for Mobilities Research. She studies people's everyday material and epistemic practices – on the move or *in situ* – including practices of place-making and (distributed) collaboration. Her approach is ethnographic, rooted in ethnomethodology, and science and technology studies. Her work critically informs collaborative, interdisciplinary socio-technical innovation in different settings.

Michael Christensen is a senior software developer and researcher at the Department of Computer Science at the University of Aarhus, and CTO, partner and co-founder of 43D ApS. He has been participating in a range of international and interdisciplinary projects, in many of them having the lead responsibility for software development. His main research interests are within experimental systems development, spatial IT, pervasive computing, and object-oriented programming. Recently, he participated as manager of software development in the EU Integrated Project PalCom.

Klaus Marius Hansen is Associate Professor at the computer science department, University of Aarhus, manager of the software area of the ISIS Katrinebjerg competency centre, scientific manager of the infrastructure group of the Danish national network for *pervasive communication* (KomIAlt), and manager of the *software* research area of the Alexandra Institute. Since 1997, Klaus Marius Hansen has been involved in numerous research and development projects with partners from industry. His current research interests include software architecture, pervasive computing, middleware, dependable systems, and experimental system development.

Mark Hartswood is a research associate in the School of Informatics, at the University of Edinburgh. His research interests concern the interdisciplinary study of user–designer relations in technology production and the impact of technologies on work practice, particularly in health care contexts. His work draws upon the traditions of participatory design, computer-supported cooperative work, and workplace studies in exploring trust, ethics, and collaboration in IT system production and use.

Sampsa Hyysalo is a postdoctoral researcher at the Collegium for Advanced Studies, University of Helsinki. The focus of his research has been on ethnographic and historical studies of health ICT that follow their development, use, and early evolution. His main interests are in the interrelations between action, imagination, materiality, and practice. Traditions of science and technology studies, activity theory, as well as user-centred and participatory design are close to his heart.

K. Neil Jenkings is a researcher at the Institute of Health and Society, at the University of Newcastle, UK. His research focuses primarily on the situated use of textual, visual, and technical artefacts in interaction. Other interests include problem-based research techniques and ICT design for user–designer–implementation activities in health care settings.

John Mariani is a senior lecturer in the computing department at Lancaster University. His research interests are in the use of ethnography to inform the design of systems, databases to support collaborative work, and data visualisation. Recently, he has been involved in study of the use of electronic patient records within the UK National Health Service.

David Martin is a research scientist working for Xerox Research Centre Europe (XRCE) in Grenoble, France. He works for the Work Practice Technology group. His work involves employing ethnographic techniques to study work and technology use from an ethnomethodological perspective. These studies are used to understand the impacts of technology in real-world settings and to feed into the design of innovative new applications. His work has covered a diverse set of topics ranging from ambulance control, to banking, health care, eXtreme Programming, and now is focused on graphic design, high-end colour printing and legal work.

Preben Holst Mogensen is associate professor at the computer science department, University of Aarhus, and CEO, partner and co-founder of 43D ApS. He has been coordinating, managing, and participating in several EU projects since 1990, most recently as co-coordinator in the EU Integrated Project PalCom. His research interests concern tools and techniques for active user involvement in system development (participatory/cooperative design); pervasive computing; cooperative analysis; prototyping; CSCW; object-oriented approaches to system developments.

Neil Pollock is a senior lecturer at the University of Edinburgh where he teaches and researches on the sociology of information technologies. He has published his work in leading journals and has written with James Cornford *Putting the University*

Online: Technology, Information and Organisation (Open University Press) and more recently with Robin Williams *Software and Organisations: The Biography of the Enterprise-Wide System Or How SAP Conquered the World* (Routledge).

Rob Procter is professor and research director of the National Centre for e-Social Science at the University of Manchester. His research interests lie within the field of socio-technical issues in the design, implementation, and use of computer systems, with an emphasis on ethnomethodologically informed ethnographic studies of work practices, CSCW, and participatory design. He is a co-editor of the special issue of the CSCW Journal on Collaboration and e-Research published in 2006.

Mark Rouncefield is currently a Microsoft European research fellow and senior research fellow in the Department of Computing, Lancaster University. His research involves the study of various aspects of the empirical study of work, organisation, human factors, and interactive systems design. This work is strongly interdisciplinary and has led to extensive and continuing collaborations with colleagues in sociology, computing, informatics, and management both in the UK and abroad.

Dan Shapiro has a long-standing track record of research in social informatics, CSCW, ethnographic studies, participatory design, and evaluation with end-users, and interdisciplinary theory for system design and spatial computing. He was coordinator and participant in several EU projects since 1992. He is currently working on projects in ambient and palpable computing and collaborative environments, and in support for the aesthetic design professions and the emergency services.

Roger S. Slack main interests are in sociology and social research at the University of Bangor. His main interests lay in ethnomethodology, conversation analysis, and science and technology studies. He has published in CSCW, social shaping of technology, interaction and advice giving in medical settings, and reflexivity. He is currently engaged in research on meeting memory using access grid tools, toxicology advising, and is about to begin a study of finding images and sounds as instructed action.

Bettina Törpel is an assistant professor at the Department of Management Engineering and teaches in the study program ‘Design and Innovation’ (http://www.design-ing.dk/) at the Technical University of Denmark. Her research interests include computer-supported cooperative work (CSCW), participatory design (PD), heterogeneous work infrastructures, and new forms of work/organisations, lay technology development and lay research, for example, for enhancing personal and collaborative work infrastructures. Her research and teaching explores how design gaming, story telling, and other creative and participative methods can be used to engage stakeholders in ICT design.

Alex Voss works for the ESRC National Centre for e-Social Science at Manchester University. His main research interests are the organisational use of ICTs and, in particular, how people meaningfully relate practices, technologies, and organisational arrangements to each other as they grapple with the practicalities of their day-to-day work. Recently, Alex has been studying the uptake and use of advanced information technologies in research practice.

Robin Williams has convened the interdisciplinary programme of social research on technology at the University of Edinburgh for over 20 years, where he established and leads the Institute for the Study of Science, Technology, and Innovation. His many books and articles on the social shaping of information and communications technology includes: *Social Learning in Technological Innovation: Experimenting with Information and Communication Technologies* (Edward Elgar, 2005 with James Stewart and Roger Slack).

Chapter 1

Introduction: Configuring User–Designer Relations: Interdisciplinary Perspectives

Alex Voss[1], Mark Hartswood[2], Rob Procter[1], Roger Slack[3], Monika Büscher[4], and Mark Rouncefield[5]

[1] National Centre for e-Social Science, University of Manchester, alex.voss@ncess.ac.uk, rob.procter@manchester.ac.uk
[2] School of Informatics, University of Edinburgh, mjh@inf.ed.ac.uk
[3] School of Social Sciences, University of Wales at Bangor, r.slack@bangor.ac.uk
[4] Department of Sociology, University of Lancaster, m.buscher@lancaster.ac.uk
[5] Computing Department, University of Lancaster, m.rouncefield@lancaster.ac.uk

1.1 Introduction

It is a commonplace but one that is probably worth repeating: user–designer relations are both multifaceted and also highly situated. In this book we want to examine some of the ways that the design of information and communications technology based systems (ICTs) can be conceptualised and what the attendant issues and rewards might be. We do not seek to set out a policy or advance a platform for the (re)configuration of user–designer relations, but to show how such relationships might be arranged and managed. In all cases the reader is invited to consider how a particular configuration of users and designers might be achieved, how it could apply to their own situation and how the practical exigencies of their own situation might impact on the production of particular configurations.

By focusing on the processes of negotiating and shaping the relations that connect use with design, we follow Suchman's call to

> replace the designer/user opposition – an opposition that closes off our possibilities for recognising the subtle and profound boundaries that actually do divide us – with a rich, densely structured landscape of identities and working relations ... (Suchman, 1994, p. 22).

In their effort to realise productively the transformative potential of new technologies in use, users and designers are inescapably thrown together – whether they actively seek collaboration or separations – neither can escape the influence of the other. The contributions to this book map out the multifaceted and situated nature of some important user–designer configurations, describing often difficult but effective (and also not so effective) ways of configuring them. They show, for example, how reconfiguring user–designer relations does not take place in isolation and provide

A. Voss et al. (eds.), *Configuring User-Designer Relations*,
DOI: 10.1007/978-1-84628-925-5_1,

some pointers to how we might begin to take the social, economic, cultural, material context into account. They illustrate and call for methods that might enable practitioners to justify the need for and hence find an organisational space for collaborative user–designer relations, itself often one of the more difficult challenges. We do not aim to produce an exhaustive typology or to specify some ideal–typical configuration; rather we want to look at the 'grammars' of user–designer relations within working divisions of labour. That is to say we want to consider how the terms are deployed in practice and in context and the impacts that these have on what is developed. Moreover, user–designer relations change or need to be reconfigured over time to fit different goals and evolving circumstances. We will start by looking at the concerns raised by the contributors to the present volume and then attempt to draw out some common themes which address the 'grammars' of user–designer relations.

1.2 The Chapters

In Chapter 2, Törpel, Voss, Hartswood, and Procter set out to provide the reader with an overview of participatory design (PD) practices and how the PD community has responded to new challenges. The PD community has been at the heart of debates about user–designer relations for more than 20 years. Törpel et al. trace the evolution of PD from its roots within the socio-technical systems school at London's Tavistock Institute, through its politicisation by the Scandinavian school of IT systems development and, finally, the varieties of PD that have subsequently emerged as it has been taken up and adapted by the mainstream of ICT systems development practice. Törpel et al.'s point, then, is that PD no longer stands for a unitary set of ideals but a rich, heterogeneous, and fluid constellation of practices, whose commitments to user–designer relations vary widely.

Chapter 3, by Voss, Procter, Slack, Hartswood, and Rouncefield, explores relationships in development and use with a focus on the role of ethnographic studies. The separation between design and use makes it difficult to design technologies that genuinely support work practices, not only because designers lack a thorough understanding of practice but also because practices evolve dynamically, in interaction with new technologies. Separated from use in time and space, design often fails to address changing opportunities and needs – often with substantial cost implications. Ethnography promises a route out of this conundrum. However, its potential is hard to realise. Voss et al. insightfully examine the difficulties that arise and seek to sketch out productive modes of engaging ethnography, concentrating on practical methods of doing ethnographically informed design. Their approach is symmetrical, that is, they strive to explicate design practices and practices of investigating use.

A brief historical review of changing systems design practices reveals how ICT system design, rooted in engineering rather than aesthetic design methods, struggled to move beyond a linear conception of its processes. Approaches based on iterative and evolutionary models – building repeated encounters between design and use into the process – have gained dominance today, especially in the 'agile' methods of extreme programming. But it remains to be seen whether such methods can facilitate new user–designer relations and socio-technical innovation processes. Moreover, changing user–designer relations are not just a matter of changing the way we design. As research within the field of computer-supported cooperative work (CSCW) has shown, it is crucial that designers also change their conception of the use practices they are designing for. Ethnographic research in CSCW potently unveils the thoroughgoing sociality of work. Whether actively collaborating or engaged with others through peripheral awareness, whether constructively working together, or involved in conflict, work is a social process and technologies must fit into, and support, the sociality of work. Some basic principles for user–designer relations can be drawn from these considerations. They point towards using ethnography to inform design and PD processes. However, as Voss et al. point out, such attempts to bridge between design and use may not address the more fundamental problems that obstruct successful design and appropriation.

Building on their own comprehensive experience, the authors then appraise just how ethnography has been used to inform design; the challenges designers, users, and ethnographers face; and the strategies they have developed to fold knowledge of dynamically evolving work practices into design. They explore analytical motivations and the uneasy relationship between analysis of existing practices and the task of envisaging the future. A summary of particularly productive methods of, and orientations towards, conducting ethnographic studies then leads into a discussion of the potential to constructively transform user–designer relations by folding ethnographic sensitivities into iterative and participatory approaches to design.

One important move towards such a goal can be made through ethnographic studies of design work itself. Drawing on a number of studies, including the investigation undertaken by Martin, Mariani, and Rouncefield (Chapter 7), Voss et al. argue that ICT system design – like all forms of work – is deeply collaborative and social, distributed across time and space, involving both users and designers. This exploration leads into a conceptualisation of design as one approach amongst several that need to be practically aligned to enable innovation in use. It benefits from and depends on analytical (ethnographic) and practical–political engagement with users. Voss et al. conclude with an exploration of how such a move towards design as part of innovation in use constitutes a fundamental, yet doable and profitable reorientation and reappreciation of user–designer relations.

The chapter by Jenkings (Chapter 4) is the first of six empirically based chapters which present some of the diversity of approaches to user–designer relations within a range of very different contemporary ICT systems development projects. The question posed by Jenkings (and it is one which has been asked many times) is whether it is possible to practice user engagement meaningfully in the context of a large organisational project involving potentially thousands of users located at multiple work sites. His point is that most – if not all – PD techniques have been developed for application within small user communities and the mechanisms for user–designer interaction on which they typically rely do not scale well. Jenkings attempts to find solutions to this recurring problem in a project within a setting which, in terms of scale and complexity, is arguably the most challenging one could possibly wish for. The UK National Health Service (NHS) is the largest employer in Europe with multiple and overlapping organisational decision-making structures and has launched Connecting for Health (CfH), said to be the world's largest civil ICT project, with the aim of providing the UK with a national electronic health record service. Jenkings' contribution is particularly timely as the NHS in England grapples with the problems of delivering the CfH project. The project has been under fire for many reasons, not least for its approach to (or lack of) user engagement. Martin, Mariani, and Rouncefield (Chapter 7) examine user–designer relations specifically in the context of the CfH project in their own chapter.

Jenkings concedes that the techniques of user–designer engagement that have become synonymous with PD cannot survive intact in an encounter with a user constituency of such scale and diversity. Jenkings describes the origins, development, and evaluation of a prototype tool, the 'Animator' intended to facilitate user–designer communication and, in particular, to help raise user awareness. While the tool itself was crafted to address the needs of a specific project, Jenkings suggests that we might look to technology-demonstrator tools more generally to facilitate user engagement on this scale. What Jenkings sets out to show is that it is possible to produce useful tools to facilitate engagement with large-scale and diverse user communities. Jenkings readily acknowledges that the 'proof' of this assertion rests, in part, on accepting more limited user-engagement goals for the approach that he describes. One lesson is clear from Jenkings's account and it is that this approach is not a short cut but must be underpinned by the very same kinds of painstaking user-engagement work which we have come to associate with PD 'in the small'.

In Chapter 5, Bonner examines the uses of PD tools and techniques by product designers developing interfaces for domestic appliances. His designers – charged with designing a new cooker – undertook four tasks: cooking a meal; considering how the technology being designed could support that activity; thinking about a week in the life of a cooker; and finally developing scenarios about potential users of the cooker.

Cooking a lunch is something that most of us will have done. Yet, when we attend to the seen but unnoticed aspects of how we go about doing this and the technologies that support our activities, it becomes interesting to consider just how design impacts on this mundane activity. Bonner shows how designers of domestic technologies employed Muller et al.'s CARD methodology to topicalise the activity of cooking a meal, and how, when looking at the process of cooking, the designers saw how their design of domestic technologies could afford that activity (Muller et al., 1995). Just how do we go about cooking lunch and how does technology afford or get in the way of this activity? The CARD methodology was sufficiently disjunctive from the normal activity to produce what we might call 'aids to imagination' vis-à-vis what it was to cook lunch.

One of the designers highlights an interesting issue regarding the 'ownership' of methods and their attendant findings – 'if all this goes well – it will be our idea – if it all goes wrong – it will be yours'. While this may well have been said jokingly, the issue of what the payoff of adopting of novel methods for reconfiguring user–designer relations might be is something to consider. We might also consider the role of the IT professional here as a methodological intermediary, providing new ways for persons to research user requirements, which can be taken up with little control yet potentially bearing the *imprimatur* of the IT professional. Further, the comment highlights the ways that such methodologies are (or become) embedded within organisational frameworks and can be used to obtain resources, make decisions about products, and so forth. Reconfiguring user–designer relations does not take place in isolation.

A second set of exercises was convened and led by designers with employees of the company not engaged in design. The designers set up a 'function filter' where cards were prioritised by frequency and importance of use within scenarios around using a microwave oven. Bonner points out the ways in which the designer-led exercises differed from those led by the IT professional – this might be glossed as a prioritisation of prototypes as a means of blocking off more innovative suggestions from users. The emerging prototype was both a means of solidifying the findings and a concrete embodiment of solutions to issues raised.

Bonner's chapter raises a number of foundational issues for reconfiguring user–designer relations – notably the importance of users of PD coming to trust the methods and what they elicit, and the organisational dimensions of design that is the need for a methodology to be recognised as acceptable and viable, and thereby to find an organisational space. The issue of ownership is also important – when the method becomes part of the organisation's repertoire of research tools, it is potentially subject to what, in another context, Knorr-Cetina (1981) has referred to as 'conversion–perversion'. To our minds, there is a need for all reconfigurations to bear a health warning – once enmeshed within organisational exigencies what a method is and what it becomes may be very different.

Picking up the challenge of achieving organisational acceptance, Hyysalo's chapter (Chapter 6) sets out a compelling rationale for the adoption of PD within the commercial sector. Many companies have employed the rhetoric of being 'customer driven', but, as Hyysalo demonstrates, this has often been merely a rhetorical device and attendant user involvement has been merely at a 'Guinea pig' level. Hyysalo notes that if we examine the 'long wave' (Freeman and Louçã, 2001) of innovation, we find that competitive advantage is not inevitably to be realised through being first to market or by being the cheapest: as artefacts and users move closer together (consider the rise of open source software as an example), it makes economic sense to involve users in innovation and development processes. This is not simply about creating niches for products but about achieving a tighter coupling between what users want and what manufacturers produce (and, in some cases, the blurring of the distinction between users and designers).

Hyysalo examines the biographies of two innovative products – a record system for diabetics and 'Wristcare', a physiological monitoring and alert system – and the configuration and reconfiguration of user–designer relations over time within each.

In the first case, patients with chronic diseases and health care professionals collaborated with designers to develop the product. While no formal 'methodology' was used, the sharing of experiences of managing illness informed the design of the system. Through this, the company developing the software found that their task was too difficult and re-focused their development efforts on a more generic patient–health care professional system for management of chronic illnesses, which was subsequently rolled out in 1998. The central issue within this collaboration is that users have knowledge and experience that would take developers considerable time and effort to acquire (if it is possible at all) – the 'learning curve' would have been too steep for the system to be developed under a reasonable business logic. Users were a key resource in the development at the start, but, as Hyysalo points out, as the system became more widely used, the company distanced itself from users and took on less of their suggestions for features and amendments. This brings out an important issue – the management of user-led calls for change. When a product is in development or has been rolled out in a small number of settings, it is comparatively easy to change things – but when there are large numbers of users with potentially conflicting needs, there are problems. In short, it is easy to manage a core set (Collins, 1988) of users as opposed to a larger and potentially more diverse population. This does not mean, as some have suggested, that PD does not 'scale', but that the ways that it does so have to be managed. Pollock and Williams (Chapter 9) show, for example, that once a system becomes a package, some of the competitive gain from tailoring based on specific uses is attenuated and – at least from the company's perspective – user participation needs to be turned towards 'generification' to maintain its value.

In Hyysalo's second case study, the 'wristcare' product developed out of company experience in the health care sector and had been developed with a vision of what the product could do for seniors. User involvement impacted only marginally on development at the early stages since the vision had solidified what the product would look like and what it would do. As the device was rolled out it became apparent that users were unable to cope with its functionality and that a substantial number could work the basic system only with great difficulty, leading to a number of potentially costly false alarms. Issues with the system meant that there was a need to focus on how it was used in context, which led to some redesign and also to a reconfiguration of functionality so as to act as a monitor for seniors rather than being solely an alert system. As with the first system, the designers found that the number of variants became unsupportable and sought to add some of the functionality into later versions of the product. The aim was, as before, to produce a packaged solution capable of being sold internationally.

Both of Hyysalo's case studies illustrate the centrality and the changing nature of user involvement over time and the ways that it can add value to products. While users were involved in both cases, there was no formal method in use – users were involved as a part of the 'natural history' of the developments. It should also be noted that the companies involved moved away from engagement with users after a brief time and that this was in part driven by the need to produce a packaged solution. Returning to the comments made above on the rhetorics of collaboration, the case studies suggest that the relevance and sustainability of collaborative relations lessens once the need to commodify is felt.

Hyysalo concludes by making some proposals for the configuration of user–designer relations in product development. He suggests that informal collaboration and social learning (Williams et al., 2005) are important, and that the extent of user involvement be considered temporally since what informs development at one stage may prove to be problematic later. Finally, Hyysalo notes that as technologies become increasingly configurable, the appropriateness of particular layers of configuration and the opportunities for and benefits of involvement of a range of players should also be considered in more commodity-oriented phases.

Martin, Mariani, and Rouncefield (Chapter 7) consider the practical issues around stakeholder participation in the deployment of a hospital information system (HIS) incorporating an electronic health record (EHR) within a UK hospital trust: their ethnographic study investigates 'participation "in the wild"' and explicates the reasons for the particular configuration of user–designer relations. The development of integrated electronic medical records is the goal of a number of national health care providers, and stakeholder participation in design, development, and implementation is seen as important. As the authors note, their study was not simply another critical engagement with the development of EHRs but an exploration of the pragmatics of participation. A hospital is a diverse and complex organisation and to understand how EHR systems might fit in requires substantial effort. This diversity and complexity is mirrored in the pool of potential user-participants and their areas of expertise. It is not surprising that a significant number of users involved in projects are 'expert' or 'super' users with substantial domain knowledge. Of course, they are not the only users involved – others will be drawn from a variety of domains, but Martin et al. observe that the choice of participants is likely to be influenced by pragmatic and political considerations. There are many differences of perspective and opinion within such groups and also project stakeholders in addition to users of the system. Their participation has to be managed in and as a part of developing the system. Martin et al.'s findings help to explain why user participation in projects of this kind may fall somewhat short of expectations reflected in the PD literature.

The project involved the configuration of a commercial off-the-shelf (COTS) software package. Martin et al. report on the ways that the project is made manageable by division into phases (Button and Sharrock, 1996) and how, in turn, these phases then influence the character and emphasis of user engagement at any particular time. So, for example, during the tendering process, user engagement is limited to Trust board members and it is only during later phases, when potentially crucial choices have already been made that users 'at the sharp end' are given an opportunity to influence the way the project unfolds. The main focus of Martin et al.'s study, however, is the realities of achieving user engagement at the clinical-user level. Here, the authors observe a variety of problems, beginning with the difficulties of recruiting end-users with the requisite expertise to inform the configuration process and the subsequent impact this had on the goals of the configuration work. As the project unfolds, Martin et al. document how the seemingly endless contingencies faced by the project manager in the struggle to keep the project on schedule – ongoing negotiations about tasks and responsibilities, evolving requirements and changing priorities, inter- and intraorganisational tensions, etc. – shape what is practical in terms of user engagement.

To address some of the observed failings in user engagement in this and similar projects, Martin et al. argue that, while research should further develop methods to enhance understanding of users' practices and contexts and to facilitate user involvement throughout the design and implementation process, it should also consider how such methods can be made to work in commercial and organisational settings and real-world design projects. This is not just a matter of revealing the need for more resources and more time. Difficulties arise because user–designer relations can be stretched beyond constructive tension by the different perspectives, interests, and pressures brought to development and deployment. Martin et al. highlight the need to shape user–designer relations in ways that are mindful of their larger contexts – including organisational demands and regulatory frameworks. In turn, user–designer collaborations can be powerful catalysts in making organisations aware of the fact that transforming the organisation is inseparable from realising innovative technical ‘solutions’.

Büscher, Christensen, Hansen, Mogensen, and Shapiro (Chapter 8) take an interesting direction in redefining the boundaries between use and design by designing for assembly – supporting users in the assembly of a repertoire of technologies for a specific task – and by focusing design on producing what we might call work-affording artefacts. Their research, undertaken as part of the palpable computing initiative, indicates the need for designers to support people in making what systems or assemblies afford for their users perceivable or ‘palpable’. The ‘disappearing computer’ is seen as eliding some of its affordances because it is embedded within objects, as opposed to being a discrete entity that can be noticed, explored, and combined with others. Put simply, what we cannot fully see we cannot fully appreciate, nor can we exploit to the full the affordances of such artefacts. To exploit the potentialities of technologies we must be able to engage with them. The disappearing computer is thus shorn of some of its affordances – the aim of palpable computing is to maintain the promise of computers embedded in artefacts, while enabling a focus on affordances. It is no use embedding computers in artefacts if these artefacts continue to get in the way by, for example, asking users about configuration options (Anderson et al., 2003). Part of the vision of palpable computing is to design artefacts that support various configurations without being overly intrusive as to the choices made – although it should be appreciated that this ‘quiet optimisation’ necessitates compromise.

The ‘visibility arrangements’ of such systems are central – the focus is not on design for an array of uses but on inspectable configurability for these uses. This suggests that designers no longer occupy an intermediating position but that the visibility arrangements of artefacts make an array of possibilities available to users – in short, users become situated designers, assemblers of arrays of work affording artefacts. In order to realise this vision, Büscher et al. propose a reconfiguration of PD, opening up longitudinal collaborations to encompass software architectures and involving software architects in the design process and making users familiar with some of the affordances of software architectures. Now, this does not mean that users have to become computer scientists, it does mean however that software architects – ‘travelling architects’ (Corry et al., 2006) – become involved in the design and development process and prototypes include some idea of software architectures that will afford the kinds of work envisaged. The long-term engagement proposed by the authors turns on a reflexive relationship between work practice and the development of technologies – ethnographic observation of work practice informs that technology development and artefacts are designed to better afford work practice. The kinds of ‘assemblies’ that users put together – in the case at hand, landscape architects using *inter alia* cameras, GPS, and maps – are central to this process: how to make things work together to enable people to do their work is, obviously, the motivation behind this technique. This does not mean designing one configuration or assembly, but enabling a number of potential configurations to be made, something which requires a robust architecture.

We have already noted the importance of long-term engagements with users and work practice in interdisciplinary teams: the design technique proposed by Büscher et al. continues this and also suggests that 'futures laboratories' are useful in developing such systems. Futures laboratories are fora for emerging work practices and the development of technologies to support them – they enable the exploration of just what types of assemblies of artefacts might support emergent work. For example, emergency medical professionals may be able to save valuable time and make more effective treatment interventions by using video devices to feed back images of a patients' condition and to receive advice as to effective management. What kinds of technologies will support this and how far these are usable and dependable in practice is something that can be explored in futures laboratories. The kinds of long-term engagement afforded by participatory design and futures laboratories are central to the development of novel assemblies of the type discussed – Büscher et al. provide an excellent illustration of the ways that such developments can work in their chapter.

Pollock and Williams (Chapter 9) take a very different focus for their examination of user–designer relations. They observe that we commonly associate the issue of user engagement with the question of how a software package or ICT system can be made to work for a particular group of users within a specific setting. As ICT systems projects seldom involve building solutions from scratch, the study of user–designer relations has tended to focus on the processes by which commodified and generic packages are adapted to meet the needs of particular users (see, e.g. Chapter 7 by Martin et al.,). Pollock and Williams note how this overlooks the issue as to how generic software packages that are capable of bridging different organisational settings (albeit with varying degrees of ease) come to be generic in the first place. They use case studies in which they track the 'biographies' of two COTS software packages to explore the nature of the 'generification work' and how it leads to artefacts, which successfully embody those characteristics that are common across different organisational settings and yet are seemingly capable of being 'localised' for any particular one.

Based on their findings, Pollock and Williams argue that, far from exemplifying 'design from nowhere' (Suchman, 1994), the suppliers of generic packages practice their own strategies for achieving an adequate degree of engagement with users. A key question often raised in academic debates about user engagement is: how is it possible to satisfy the diverse needs of multiple users? Pollock and Williams show through their findings how package suppliers employ particular user-engagement strategies so as to be able to rein in demands to meet diverse and potentially conflicting requirements and so achieve what is, for them, a practical and appropriate balance between being seen to be responsive to their users while pursuing a generic solution. To put it simply, Pollock and Williams illustrate how software package suppliers employ a collective user-engagement strategy to discipline and shape user requirements, leading user community members to compromises rather than insisting that their individual needs be met. By pursuing engagement at the user community level, suppliers are able to manufacture a situation where users recognise that it is in their best interests to align their requirements, which – though they are less optimal –

have more chance of being implemented, rather than hold out for multiple, distinct solutions, which – though they better reflect individual requirements – are also in more danger of being ignored. At the same time, software suppliers are willing to give some clients preferential treatment, especially those who are perceived to be able to exercise leadership of their communities.

Pollock and Williams' final point is that large-scale software packages are not the monolithic artefacts they have sometimes been characterised as being, but an intricate and pragmatic balancing of the generic and the particular, such that distinctive organisations and standardised solutions are able to co-exist effectively.

1.3 The grammars of User–Designer Relations

The case studies in this volume elaborate a variety of 'grammars' of user–designer relations found in different organisational settings, each uniquely and individually rooted in their historical contexts, but continuously reformed and remade, both in the light of shifting circumstances, and, crucially, in peoples' (both users' and designers') attempts to comprehend and realise technology's transformative potential.

By 'grammars' of user–designer relations, we point to how the various working divisions of labour between users and designers actually play out in practice – the everyday shared understandings of with whom various responsibilities, expertise, and competencies reside, and how they might be properly discharged or applied. Understanding the logics underpinning grammars of user–designer relations is not simply an academic turn, but something that both users and designers undertake to do. We see through the case studies presented here how people variously judge what might be reasonably asked for, what they might expect to get and when, who has influence, who might be best approached with particular problems, what promises it is safe to make to whom, who might make good allies and who is in competition, who has need for information, and how best they might be approached, and so on (seen most strikingly in Voss et al.'s chapter). We see people not only exploring and orienting to local logics or grammars in this way, but also evaluating them (again, an undertaking not solely in the purview of academics); they make judgements about whether current modes of engagement are appropriate to shifting business models and objectives (e.g. Hyysalo, Pollock and Williams) or whether novel approaches are likely to survive and become part of the design team's repertoire (e.g. Bonner), as well as seeking to reshape them to meet new needs or respond to changing circumstances and aspirations (e.g. Büscher et al.).

All of the studies in this volume play an important role in adding to our understanding of a variety of user–designer relations in different contexts and settings – their affordances, problematics, and adaptation to shifting circumstances. This includes general lessons and recurring patterns that we might see played out across a number of contexts (which we attempt to draw out in the conclusions), exemplars of particular forms of practice that we can draw upon as resources and appropriate for own needs, as well as a host of consequential nuances that may or may not be of immediate relevance – but which can perhaps give us the sensitivity to anticipate the implications of different configurations of user–designer relations in our own circumstances.

The aim of this book, then, is to deepen our understanding of user–designer relations so that we, as users, designers, or academics offering advice, can grapple with the problems user–designer relations pose with a sophistication born of an engagement with their workaday exigencies. It is in this spirit that we invite you to read on.

References

Anderson, S., Hartswood, M., Procter, R., Rouncefield, M., Slack, R., Soutter, J. and Voss, A. (2003). Making Autonomic Computing Systems Accountable: The Problem of Human-Computer Interaction. In *Proceedings of the 1st International Workshop on Autonomic Computing Systems, 14th International Conference on Database and Expert Systems.* Applications, Prague, September.

Button, G. and Sharrock, W. (1996). Project work: the organisation of collaborative design and development in software engineering. *Computer Supported Cooperative Work*, v.5, n.4, pp. 369–386.

Collins, H.M. (1988). Public experiments and displays of virtuosity: The core-set revisited. *Social Studies of Science*, v. 18, pp. 725–748.

Corry, A.V., Hansen, K.M. and Svensson, D. (2006). Travelling architects – A new way of herding cats. *Quality of Software Architectures (Lecture Notes in Computer Science 4214).* Berlin: Springer, pp. 111–126.

Freeman, C. and Louçã, F. (2001). *As time goes by: From the industrial revolutions to the information revolution*. Oxford: Oxford University Press.

Knorr-Cetina, K. (1981). *The Manufacture of Knowledge: An Essay on the Constructivist and Contextual Nature of Science*. Pergamon. Oxford.

Muller, M.J., Tudor, L.G., Wildman, D.M., White, E.A., Root, R.W., Dayton, T., Carr, R., Diekmann, B. and Dykstra-Erickson, E. (1995). Bifocal tools for Scenarios and Representations in participatory activities with users. In Carroll JM (ed.) *Scenario-based design: envisioning work and technology in system development.* New York: John Wiley and Sons.

Suchman, L. (1994). Working relations of technology production and use. *Journal of Computer-Supported Cooperative Work*, v. 2, pp. 21–39.

Williams, R., Slack, R.S. and Stewart, J. (2005). *Social Learning in Technological Innovation: Experimenting with Information and Communication Technologies*. Aldershot: Edward Elgar.

Chapter 2

Participatory Design: Issues and Approaches in Dynamic Constellations of Use, Design, and Research

Bettina Törpel[1], Alex Voss[2], Mark Hartswood[3], and Rob Procter[2]

[1] Technical University of Denmark, bt@ipl.dtu.dk
[2] National Centre for e-Social Science, University of Manchester, alex.voss@ncess.ac.uk, rob.procter@manchester.ac.uk
[3] School of Informatics, University of Edinburgh, mjh@inf.ed.ac.uk

2.1 Introduction

This chapter has two related aims: (1) to give an overview of the various approaches developed under the heading of participatory design (PD) and (2) to discuss their assumptions and commitments. Introducing the field of PD involves exploring its scope and definitions and delineating its internal structure. This will necessarily involve highlighting certain features at the expense of others, so we do not wish to claim that we are producing a definitive and comprehensive account. Rather, our aim is to delineate different key traditions and approaches in PD. The field of PD is diverse and it is therefore necessary to match its different approaches to the situation at hand, reflecting on who the relevant actors are, what their interests and commitments are, and how they relate to each other. This will then allow us to ask what concerns, assumptions, and commitments guided work in PD in terms of the questions asked about practice, design, and research, the direction the answers take as well as the concepts, methods, and literary genres used.

We will argue that it is of vital importance that this background is taken into consideration when doing PD and that a key ability of PD practitioners is to skilfully *match PD approaches* with their concepts, methods, and assumptions on the one hand and *the phenomena* encountered in the setting in which (re) design takes place on the other. We do not believe that there can be a comprehensive set of rules that allow decisions about an adequate match to be made in a schematic way, but that, instead, a process of reflection is required that takes into consideration the specific features of the design situation. Only by doing this, can one make an adequate choice of the approach to take in a particular PD intervention. Consequently, this chapter seeks to sensitise the reader to the fact that approaches in PD are diverse in terms of the questions they foreground, their approaches to answering them, as well as the concepts and methods used. The specific objectives and settings for PD are just as

A. Voss et al. (eds.), *Configuring User-Designer Relations*,
DOI: 10.1007/978-1-84628-925-5_2,

diverse and differentiated, and the picture is further complicated by the fact that approaches are often modified or combined. In sum, then, there is a need to unpick what it might mean for PD to be practiced in a way that is *adequate* with respect to the situation at hand.

In the next section we introduce PD, explore its relevance in the context of user–designer relations, and define how we delimit the field for the purposes of this chapter. In the main part of this chapter, approaches in PD and those relevant for PD from neighbouring fields are described and put into context. Each approach will be characterised in terms of assumptions made, concepts introduced or applied, and/or methods developed. Some approaches are treated more extensively than others, according to the role they have played in PD. In the discussion we comment on the differences between the approaches and the need for practitioners to be aware of them when selecting a conceptual framework and practical approach to PD.

2.2 Participatory Design

Participatory design is about the direct participation of those whose (working) lives will change as a consequence of the introduction of a computer application. Participation potentially relates to all aspects, phases, and activities of development, for example, decision making, designing, developing, deployment, and further development in use. An examination of the PD literature shows that different authors have stressed different aspects of the participatory process when attempting to define the field.[1] Reasons provided for engaging in PD can be categorised as pragmatic, theoretical, and political (Greenbaum 1993; Bødker et al. 2004, p. 58).

Pragmatic reasons for PD include, for example, that workers, as experts of their work, work practices, work organisation, and means of labour, are able to contribute their expertise to discussions about activities shaping their own future work so that it is most beneficial and efficient, and the resulting products are good and the employed technology is appropriate.

Theoretical arguments for PD can be made on the basis of a range of different analytical approaches: phenomenology (Ehn, 1988; Winograd and Flores, 1986); ethnomethodology (Suchman and Trigg, 1991); or activity theory (Bødker, 1991). Science and technology studies (see also below) have provided theoretical arguments for choosing a PD perspective, often derived from arguments about the inseparability of the social and the material, practice, and technology (e.g. Suchman, 2000, 2002).

Political arguments in support of PD typically refer to industrial democracy and the right of workers to determine their own working conditions, including their means of labour (e.g. Ehn and Kyng, 1987).

---

[1] Classical references with tentative answers to the question what PD is about are, for example, Bjerknes et al. (1987), Greenbaum and Kyng (1991), Schuler and Namioka (1993), Kuhn and Muller (1993), Trigg and Anderson (1996), Blomberg and Kensing (1998). The evolution of the term can also be seen in the proceedings of the biannual Participatory Design Conference.

Major issues concerning PD are (1) *expertise* (e.g. the expertise regarding workers' own work as a useful resource for designing computer applications); (2) *innovation* that is beneficial and sustainable; (3) *multiple viewpoints* and taking differences seriously as facts and resources; (4) the interplay between work practices, technology, organisational, and other aspects of the considered work environment; and/or (5) *context*; (6)the meaning of authentic *experience*, of 'being there' instead of 'talking about' and 'developing for'; (7) real-world problems with real-world solutions that get achieved by *hands-on methods* and activity; (8) empowering weak and/or marginalised societal groups as part of ICT design; and (9) *reflective practice* in all those areas of practice where relations of design and use of computer applications are of importance.

One important root of PD is in technology development projects with worker involvement, which took place in Scandinavia and other countries from the 1970s onwards (for an overview see Floyd et al. 1989). The term 'participatory design' was introduced after extensive practical experiences had been gained, especially in the Scandinavian projects. It was formulated through an international dialogue that has resulted in and has been expressed through the biannual Participatory Design Conference (PDC), principally amongst scholars interested in worker participation in technology development. When Scandinavian ICT development projects with worker participation are mentioned in publications, the reader is also often referred to the so-called 'Scandinavian School' of systems development.[2]

In the years following the pioneering efforts of the Scandinavian School and others (see next section), PD has continued to evolve, such that now it is not only *about* multiple voices and their inclusion in design, but also *has* multiple voices distinguished by approaches and efforts, proponents, assumptions, design foci, etc. The field has become internally differentiated so that diverse traditions have been established in PD and related fields, and has advanced in pursuit of productive working relations. Not only newcomers to PD, or users interested in or embarking on PD projects, but even experienced scholars and other practitioners can find it difficult to understand the different motivations, approaches, and theoretical considerations that make up and inform PD.

In an attempt to make this diversity intelligible to potential practitioners, Muller et al. (1993) devised a taxonomy of PD practices along the following dimensions: time during the development lifecycle (from requirements gathering through to summative evaluation); modes of participation (software professionals participating in the users' world and vice versa); and scale (small groups through to large groups), and then mapped within them many of the common variants of PD.

Muller et al.'s taxonomy remains a useful tool to help navigate the space of PD techniques. Overviews from other perspectives can be found in Muller (2002), Kyng

---

[2] Floyd et al. (1989) contains a historical reconstruction of Scandinavian ICT development projects with worker/user participation, concepts and ideas that provided the background, and important references for further reading. Bansler (1987, 1989) argues for a particular classification of Scandinavian approaches to ICT development, reconstructs relations between the categories and locates predecessors of PD.

(1998), Clement and van den Besselaar (1993), and Floyd et al. (1989). In the following sections of this chapter, we attempt to provide a thematic overview of PD. We will pay particular attention to the specific assumptions, analytical commitments, and the methods that characterise each aspect. In the subsequent discussion we will relate these to the possibilities and constraints of making an appropriate match between the respective approach and a setting to be (re)designed. Most approaches within and around PD are related in one way or another, often in ways that are not made explicit. However, the specific historic interrelations will not systematically be discussed here; this chapter rather serves the function of introducing approaches in their thematic variety.

2.3 Participatory Design Approaches

Some PD approaches might be called 'comprehensive' because they are meant to cover a very broad range of issues in the whole ICT development process and in many different settings. Examples are the socio-technical approach and the collective resource approach described below. Other relevant efforts have less far-reaching ambitions, such as efforts in corporate research and development. Many other efforts whose main focus is not PD have been influential on the development of the PD community in terms of concepts or methods. These include computer-supported cooperative work and workplace studies, as well as science and technology studies.

2.3.1 The Socio-Technical Approach

The aim of the socio-technical approach to the design of work is for social and technical systems to be locally optimised within a specific organisation, especially in their interplay, and in ways that are beneficial for the workers/employees, for example, in terms of work satisfaction and working conditions. Local conflicts and obstructions are mainly addressed by local structural change. The socio-technical approach is based on the assumption of *societal harmony* in the sense of generally equal possibilities for everyone, for example, in terms of participation, articulation, and negotiation.

Although proponents of the socio-technical approach acknowledge that conflict, struggle, and power on levels beyond the local organisation can be relevant, research and practice activities within the approach have continued to be geared towards local organisational phenomena (e.g. Mumford 1987, pp. 70f). Workers/employees are to be supported locally, without assuming and taking into account possible further-reaching conflicts, struggle, and power differences on a larger scale and their potential impact on, and consequences for, local conditions.

Soon after World War II, the socio-technical approach emerged at the Tavistock Institute of Human Relations in London (see e.g. Mumford, 1987; Trist, 1981). Some central constituents of this approach were:

- a theoretical grounding in the general systems theory of von Bertalanffy;
- action research as the preferred strategy for research and intervention;

- a preference for autonomous workgroups with a democratic, participative, and *laissez-faire* style as an organisational structure (following Kurt Lewin's early work with such groups); and
- research in the tradition of the so-called 'human relations school' in work psychology (whose proponents include Argyris, Bennis, Schön, and others).

Originally, work within the socio-technical approach was conceptualised as analogous to therapeutic work: scholars specialising in work improvement would intervene in order to help, or heal, the workplace by means of their particular expertise. This help was not immediately directed towards individuals, but rather towards the local work organisation and the latter was referred to as a *socio-technical system*.

In the early 1960s, Einar Thorsrud from Norway was instrumental in inviting members of the Tavistock Institute to collaborate on some of the early Scandinavian industrial democracy projects. Drawing on the experience gained, the research objectives of the socio-technical approach expanded to address the challenges of how workers can gain more rights and responsibilities, participate effectively, and be increasingly involved in decision making and determining the circumstances of their work.

2.3.2 The Collective Resource Approach

The collective resource approach is built on the assumption of inherent and pervasive conflict, struggle, and unequal power relations, rather than an assumption of societal harmony as in the socio-technical approach. Society, in this view, consists of classes with antagonistic interests such that conflicts between them cannot be avoided, neither in society at large nor in smaller units such as organisations. Proponents of the collective resource approach take an explicitly partisan approach to developing technologies, favouring and supporting the weakest parties in the conflict, usually assumed to be 'the workers'. In this view, the workers' position should be strengthened to enable them to push their agendas forward against exploitative capital endowed with far greater power and resources.

The actors according to the collective resource approach are: the workers (or users), the designers, the unions, and PD researcher-designers. The workers are conceptualised as jobholders who make their labour available in return for money and as part of formalised long-term employment. The unions are assumed to know and represent the interests of the workers, act on their behalf and according to their interests. The PD researcher-designer carries out research and design, on the worker, for the worker, and in the interest of the worker. He/se does this in collaboration with the worker and in accordance with the unions' strategies. This kind of research-design includes creating situations, for example, workshops, prototyping, etc., where the workers have the opportunity to contribute directly to the research and design, responding within the frame predefined by the researcher-designer.

In summary, the principles, goals, and claims of the collective-resource approach in PD include:

- a societal conflict perspective;
- the overarching goal of creating industrial democracy: participation in design, design for participation, design in support of work, and design for skill;
- joint development of work, organisation, and technology;
- work and workers' expertise as point of departure for technology development;
- design by doing;
- using languages that are familiar to the participants;
- design as mutual learning;
- participation in design as enjoyable; and
- design as situated.

For more on these principles, goals, and claims see, for example, Ehn and Kyng (1987), Ehn (1988, 1993), Greenbaum and Kyng (1991), Bødker et al. (1993). A range of concrete procedures are an essential part of the collective-resource approach. They are also extensively described and explained in the collection of papers edited by Greenbaum and Kyng (1991).

The collective-resource approach was formulated in the course of design-oriented research projects such as the NJMF (Nygaard, 1975), DUE (Kyng and Mathiassen, 1982), DEMOS (Sandberg, 1979), UTOPIA (Bødker et al., 1985, 1987) and the AT project (Bødker et al., 1993). Initially, it was largely a response to the socio-technical approach and its assumption of societal harmony, but eventually became a foundational programmatic framework in its own right (e.g. Ehn and Kyng, 1987). The concept of labour processes as proposed by Braverman (1974) has served as reference concept of the collective-resource approach (e.g. Ehn and Kyng, 1987, 33ff). Early projects were underpinned by political ambitions to further workplace democracy through collective bargaining, while later ones, such as UTOPIA, began incorporating methodological innovations such as prototyping (Bjerknes and Bratteteig, 1995).

The contrast to the socio-technical approach seems to almost have been part of the identity of the collective-resource approach and is usually mentioned in its programmatic articles. The collective resource and socio-technical approaches have developed in parallel and, through debate within and between their respective communities, many of the differences between them have been partially overcome (for a discussion, see Bjerknes and Bratteteig 1995).

Some projects attempted to avoid aligning themselves with either the collective-resource or socio-technical approach. One such project was the Florence project (Bjerknes and Bratteteig 1987). One of its key aims was empowerment of a group of workers that was marginalised in more than one way, namely nurses as low-wage category women workers (for a retrospective assessment see also Bjerknes and Bratteteig, 1995; Bratteteig, 2003). Relations of local and global power structures and conflicts were assumed to exist; they were explored and attempts were made to transcend the 'local confines' of the socio-technical approach. At the same time, alternatives were developed to the collective-resource approach to analysing and transforming organisations and wider society. While being neither a socio-technical approach nor a collective-resource approach project, the Florence project was both influenced by and influential for their development.

Publications relating to the Collective-resource approach are amongst the most widely cited such that it has become a common frame of reference in discussions on PD. For example, after the collection of research papers edited by Greenbaum and Kyng (1991) containing its programme, concepts, methods, and creeds was published, the procedures and concepts of the collective-resource approach became internationally acknowledged and increasingly acquired the status of 'standard' appropriate measures for PD.[3]

2.3.3 MUST

There have been further PD programmes that are comprehensive and integrative in the sense that they provide very concrete guidance for organising, planning, and conducting PD processes within organisations. One such approach is MUST (a Danish acronym for theories and methods for design activities; Bødker et al., 2004). This approach is mainly concerned with making PD relevant and practically feasible within the divisions of labour found in modern companies. In particular, it emphasises the importance of the wider organisational context in which PD tends to take place, as well as the fact that most systems development today does not simply take place in-house but involves external players and market relationships. MUST stresses the fact that any design project takes place in a division of labour involving different stakeholders with different (potentially conflicting) interests. However, in contrast to other PD approaches, it does not take an explicitly partisan stance, but, instead, seeks to include the interests of management and of ICT departments in the PD process. Another important distinguishing element is that MUST is explicitly designed to be taken up within ordinary organisational contexts in the absence of a research intervention. For example, while ethnographic-style observation of work is recommended as an element of its methods, MUST calls for such studies to be undertaken by ICT professionals themselves rather than professional researchers.

MUST makes a fundamental distinction between an ICT *design* project and an *implementation* project (ibid. p. 23). The former provides the input for key decisions to be taken in the selection of a standard system, the development of an invitation to tender, or a possible decision to develop in-house. It emphasises the need to understand stakeholder needs as well as the vision the system is meant to implement. As this vision will inevitably involve a change process within the organisation (rather than being merely technical change), it is crucial to establish organisational support for this vision amongst all stakeholders – end-users and management alike.

A key element in ensuring this is seen in insisting "on openness of expected consequences and [offering] the various stakeholders in the design project the best possible decision-making foundation" and providing "an evaluation of the expected pros and cons of implementing a vision" (ibid. p. 55). Bødker et al. (ibid. p. 58) cite both

[3] As an indicator, 12 out of the 23 full papers presented at the Participatory Design Conference 2004 (Clement and van den Besselaar, 2004) contain at least one reference to "Design at work" edited by Greenbaum and Kyng (1991); at the Participatory Design Conference, 2006 (Jacucci et al., 2006), 9 of the 15 full papers refer to it at least once.

pragmatic and political motivations for the inclusion of strong user participation in their approach (but they do not draw explicitly on any underlying theoretical framework), and they point out that even when the emphasis is on pragmatic aspects, it is important to consider just who participates in the process. For example, there is a danger that middle management will present an overly simplistic view of how work gets done if they are not intimately familiar with actual working practices and how the procedures they are primarily concerned with get translated into practical action on the ground (ibid. p. 59).

The MUST approach defines a number of mechanisms for participation ranging from ways to inform staff about development through interviews and participation in workshops to membership on steering committees. Which of these mechanisms is adequate is to be decided on the basis of the specific circumstances of the project as are questions about the involvement of union representatives or shop stewards (ibid. p. 60).

MUST does not make any assumptions about or recommendations for the methods used in the implementation phase of the project. While it is not uncommon for PD approaches to focus on design-oriented activities rather than implementation, few make as clear a distinction between them as MUST does. Inevitably, the implementation project will involve further design activities, but these are carried out on the basis of the understanding and agreement established in the original design project (ibid. pp. 24–25).

2.3.4 Participatory Design in Corporate Research and Development

PD in corporate research and development (corporate PD) is largely practiced in (or close to) research and development departments of corporations, for example, in product development within the ICT sector. Here, the participation is part of the collaboration of members of the research and development departments and members of potential target groups who would either buy or work with the products under development. This differs from the approaches discussed so far that refer to specific target settings, where participants are obliged to accept the introduction of new technologies rather than open markets of largely anonymous, but discretionary, consumers. Examples of this latter variety of PD include Muller (1992), Buur et al. (2000), Buur and Bødker (2000), and Poltrock et al. (2003).

Usually, the motivation behind these efforts is a practical one stemming from the experience that usability issues are often addressed too late in the development lifecycle to have real impact and that laboratory-based approaches are found to be inadequate to reflect the complexities of real-world use situations (Buur and Bødker, 2000). Consequently, the emphasis in these studies is often on ways of establishing contexts (events, physical spaces, and configurations of resources) that bring together the “context of use” and “design knowledge”, while opening up spaces to enable and “inspire innovation” (ibid. p. 302). The aim is to provide ways in which different participants (designers, usability experts, end-users, etc.) can bring their knowledge to bear and to facilitate the development of a common ground and shared understanding that can enable the collaborative exploration of design options.

These studies tend to take the existence of a design project for granted, assume that the actors are willing to play a role in them, and that the main problem is to open up spaces for interaction for the development of shared understandings and collaborative design activities. Conflict is often not a central concern, but this does not mean that the authors deny its existence. Potentially or actually conflicting perspectives between different groups of people are frequently mentioned as empirical findings and they are treated as phenomena to be dealt with in the PD process. However, they are not accorded the status of a primary concern underlying the PD approach as a whole as, for example, in the collective-resource approach. In addition, corporate PD often deals with market situations where the end-user is a potential customer for a product. This implies a different set of relationships between the sponsors of a design exercise, the designers, and the end-users as the latter are not concrete individuals known in advance and can make choices to buy or not to buy the product.

2.4 Related Approaches

The PD community has enjoyed fruitful relations with a number of other (sub)communities involved in ICT systems research and design. We characterise some important examples in this section.

2.4.1 Computer-Supported Cooperative Work and Workplace Studies

An especially strong relationship exists between the communities of PD and *computer-supported cooperative work* (CSCW), key aspects of which are elaborated in the special issue of the CSCW Journal on ‘Participatory Design in CSCW’ edited by Blomberg and Kensing (1998).

One formative direction in CSCW of relevance to PD is the tradition of longitudinal studies of the introduction and adoption of technology. The aim of these studies is to understand the role ICTs play in the context of collaborative work and the way they are appropriated and used by workers as part of their everyday activities. Technologies do not enter the workplace readymade and ready-to-use, but need to be embedded in a set of local circumstances and existing practices. This has led to an emphasis in CSCW studies on the co-development of work, organisation, and technology in use (e.g. Karsten and Jones, 1998; Orlikowski, 1996). CSCW researchers have pointed to the fact that participation by workers in processes of appropriation is often inevitable even where it is not actively encouraged or acknowledged (Procter and Williams, 1996). The question is not so much *whether* workers participate but *when*, *how, and to what extent* they can influence decisions about system implementation and use. Very often, their influence is restricted to later phases of technology adoption and use. Studies have highlighted the forms such appropriation can take in the absence of formal procedures for participation (e.g. the special issue of the CSCW Journal, edited by Andriessen et al., 2003; also *cf.* Törpel et al., 2003).

Another formative direction is geared towards designing new basic technological options for supporting specific work practices in cooperative work. These practices are meticulously explored, observed, and analysed in extensive ethnographic fieldwork, focusing on the cooperation between individuals with respect to location, time, content, and resources. These *workplace studies* (e.g. Luff et al., 2000) are often guided by the study principles of ethnomethodology (Garfinkel, 1967) and are used in CSCW and PD to uncover features of situated action (Suchman, 1987) by providing a detailed, systematic understanding of peoples' practices, the working division of labour and the resources used in conducting activities.

Workplace studies rely on the presence of a researcher to observe and analyse, and therefore the approach is not intrinsically a participatory one. However, the fact that workplace studies pay close attention to the detailed features of peoples' activities, are conducted through direct observation of actual practice and attempt to represent their findings in a way that members of a setting would recognise means that they can be an important addition to participatory design practices. They can help, for example, to uncover *seen-but-unnoticed* features of activities that participants in a PD process might otherwise find difficult to uncover and draw attention to. Examples of workplace studies of particular relevance to PD are Suchman et al. (2002), and the contributions in Luff et al. (2000).

Recently, researchers have developed approaches that combine workplace studies with PD to develop new forms of longitudinal engagement between designers and users, for example, *co-realisation* (Hartswood et al. 2000; Hartswood et al. 2002, 2007, see also Chapter 3). The aim is to find ways of drawing on observations made of working practices and making them directly relevant to design, while facilitating the participation of end-users in the process of developing and appropriating technologies.

2.4.2 Science and Technology Studies

Science and technology studies (STS, e.g. Bijker and Law 1992; Mackenzie and Wajcman 1999) investigate the interplay between social, economic, and political factors and the development of science and technology. They are often interdisciplinary endeavours drawing on disciplines such as sociology, social anthropology, economics, or philosophy of science. With their critical stance and critique of simplistic linear accounts of technological development, they have, in many cases, been highly relevant for PD (e.g. Suchman 2000, 2002). They provide rationale, motivation, and impulses for building concepts and devising methods for PD. They are also useful for guiding reflective practice, for example, when exploring and understanding the possibilities and limits of participation in technology development (e.g. Hyysalo and Lehenkari, 2002).

Science and technology studies host diverse theoretical traditions, for example, Social Construction of Technology (SCOT; e.g. Bijker and Law, 1992), Actor Network Theory (ANT, e.g. Latour, 1999), and an interest in historical reconstructions of technological systems (e.g. Hughes, 1983). Because of the diversity of theories within science and technology studies and the various ways in which they have interacted with PD, it is difficult to analyse the role they have played. However, it seems fair to say that their usefulness mostly lies in the extensive and thorough reconstruction of constituents, meanings, and relations. In STS, serious efforts are undertaken to appreciate and take into account the languages, statements, and agendas of all actors, and also in their dynamic interplay. While STS researchers tend to restrict their work to such reconstructions, there are some who either seek to engage with other disciplines such as HCI, CSCW, and PD or who span the boundaries between these communities themselves. As with CSCW, the question of making science and technology studies relevant to design is a problematic one as the general lessons to be learned are difficult to factor into concrete design efforts or even methods for design.

2.4.3 Participatory Action Research, Developmental Work Research, and Related Approaches

Action research (Lewin 1946) refers to research where one important aspect of studying social – including socio-technical – phenomena is to attempt to change them in a beneficial direction through an iterative cycle of analysis and intervention. For several PD approaches, the notion of action research is central and integral to the self-image of the PD community.

Design-relevant research provides better chances to be appropriate for its target settings if members of the settings are provided a chance to give feedback on whether the research and changes were appropriate and beneficial. Yet, doing research, intervening, and obtaining information on possible effects does not necessarily involve opportunities for far-reaching participation. Some directions within action research involve the danger of establishing or consolidating the ideology of a good and benevolent researcher who knows the interests of the participants under scrutiny and who neither has her/his own interests nor is personally affected by the researched phenomena.

Participatory action research (PAR; e.g. Whyte, 1991) builds on the assumption that the people whose lives, practices, circumstances, etc. are scrutinised in research and who potentially are affected by the research are, or can be, qualified to become co-researchers along with people who traditionally have the background, role, and responsibility to act as researchers. A number of the proponents of PAR are themselves part of the PD community (e.g. Greenwood, 1992). Such approaches are especially interesting for PD to the extent that they provide all actors with possibilities to contribute to the design process.

Another approach to research and development with relevance for PD is that of *developmental work research* (DWR, e.g. Engeström, 1987), which is based on activity theory (Leont'ev 1978, 1981). A worker-inclusive repertoire of action research and collaborative development methods geared towards developing the means for improving work practices is laid out under the label 'change laboratory'. Many proponents of the developmental work approach and other PD authors/practitioners draw on PAR, especially as far as it is related to technology development and use (e.g. Karasti, 2001).

Finally, approaches such as social learning theory, with its concepts of 'communities of practice' (Wenger, 1999), the notion of communities of interest (Fischer, 2001), and notions of knowledge management and social capital (Ackerman et al., 2003) have been relevant for PD as far as they are concerned with practices of technology development, including the actors involved.

2.5 Moving Beyond Use, Design, Research, and Participation as 'a Matter of Course'

As ICTs become ever more ubiquitous, the circumstances in which design, research, and other work/use practices take place become increasingly (or more visibly) multifaceted, heterogeneous, and dynamic. We can no longer assume, for example, that ICT design and use take place in the context of waged work and its organised relationships. To focus exclusively on waged work as the context of design and use might mean to miss relevant practices, meanings, and relations. Different kinds of work, unwaged work, leisure, and recreation are now potentially relevant design spaces involving both individuals as well as formal and informal groups of people.

Even if we focus on one area of life, for example, work, we need to carefully inquire into the specific features of the setting. We need to pay attention to the kind of work undertaken, organisational factors, and other circumstances we are dealing with if we do not want to miss relevant and specific local realities in our research or design activities. Concrete work practices and organisational arrangements can differ along dimensions such as the following:

- Size of the organisation: from small to large;
- Employment status: from voluntary work to short-term freelance work to long-term contracts;
- The extent to which a horizontal division of labour is present: from clear and stable organisational units to diversity and change in communities;
- The extent to which a vertical division of labour is present: from clear and stable hierarchies to diversity and change in roles;
- The degree to which the organisation serves as a buffer between individual workers and the market: from strong to weak or even enforcing market forces within the organisation;
- The degree to which stakeholder groups can be represented and their issues negotiated: from structures of representation in formalised bodies to transient negotiation constellations;
- The degree to which work takes place in particular locations (including hot-desking and work at home) or is mobile.

This (non-exhaustive) list shows how multifaceted the context of design work is and points to some of the phenomena that are of relevance to the practitioner and of interest to the researcher. We have shown how different traditions within the PD community build on different assumptions about features such as these and wish to alert PD practitioners to the need to select conceptual frameworks and practical approaches according to the circumstances at hand. It is important to stress that this appropriation of previous work on PD involves more than selecting tools from a set of possible choices but requires a reflection on what constitutes the situation at hand, who the relevant actors are, what commitments they have, what assumptions are made, what resource are available, and so forth.

What unites the various voices in PD is that they tend to view the processes of design and use as being related to each other and mutually shaping, taking place in differentiated environments, involving people, artefacts, and various historical relations between these. The constituents and relations display specific features and current states; they have evolved in specific ways so that they have generated specific sediments. Appropriate ways of analysing specific settings need to be found and practiced; these involve meticulous scrutiny, openness to experiences beyond what we might expect as 'a matter of course' and sensitivity for historical processes and future possibilities. It is these socio-material relations and everyday practices of life that design and use to be seen as being part of.

This raises the question how we, as PD practitioners or researchers, relate to the setting, how we are connected to it, to the actors involved, how we participate ourselves as we enter the "networks of working relations" (Suchman 2002). We cannot remain detached analysts and elicit a picture of factual relationships ("A talks to B") as input for design, but we need to understand and engage with their significance and the practical politics involved, both the large-scale societal 'capital-P' – Politics – and the 'small-p' – politics of the setting[4] (Büscher et al. 2002). Approaches based on the assumption of a collective-bargaining situation (like the collective-resource approach) will be less useful in situations involving primarily local conflicts embedded in the wider context of labour relations.

Participatory design's conceptual vocabulary has largely evolved in the world of organised waged labour and consequently the activities constituting PD are oriented to this context: relevant actor groups are defined or to be defined; hierarchy levels, departments, units, responsibilities are taken as phenomena relevant for developing ICT systems; constituents and activities of the PD process are proposed as interventions in this space that are often organised as projects, a format that fits well in the context of organisations and their established social structures. These assumptions and the format of the intervention that arises from it may not fit well with other settings of work outside traditional organisational forms or non-work activities (Törpel et al., 2003). In a similar fashion, approaches that assume a bespoke design that takes place in isolated settings without the influence of wider market relations will fail to capture the divisions of labour and potential conflicts involved in product procurement (Bødker et al., 2004).

2.6 Conclusions

In this chapter, we introduced PD and outlined important developments within this field as well as in related areas such as CSCW, science and technology studies, and action research. We have stressed the differences between approaches, especially regarding their motivation and underlying assumptions. Some approaches such as the collective-resource approach make explicit and specific assumptions about the exis-

[4] In practice, larger societal issues and local relations are inevitably tied to each other, so this distinction should be seen as merely an analytical one.

tence of conflict in the design context and take a partisan stance, aiming to support workers in their struggle against opposing forces. On the other hand, approaches like MUST and corporate PD approaches specifically aim to include all actors involved, thus making PD easier to adopt as a strategy for ICT systems development within organisations.

Another important dimension in PD is the conceptualisation of skills, the role of situated working practices, and the role of 'experts' in the design process. While the socio-technical approach, for example, relies on bringing about change through interventions administered by external change agents, others focus on facilitation and the development of a process that draws on the skills and actions of end-users and designers to shape the design process and its outcomes. The question how PD practitioners and researchers relate to the networks of social relations they find in the setting is an important aspect of doing PD.

This chapter provides some background information that we hope will sensitise the reader to the important aspects of the practical decisions they will inevitably have to make when practicing PD. It cannot provide a set of simple rules that can be applied without problems to choose a suitable approach, but it can act as a starting point for deliberations conducted in the light of information about the situation at hand, so that an *adequate* way forward can be chosen.

References

Ackerman, M. S., Pipek, V. and Wulf, V. (eds.). (2003). *Sharing Expertise: Beyond Knowledge Management*. Cambridge, MA: The MIT Press.

Andriessen, E., Heeren, E., Hettinga, M. and Wulf, V. (eds.). (2003). Special issue on "Evolving Use of Groupware", *Computer Supported Cooperative Work*, 12, 4.

Bansler, J. (1987). Systemudvikling: teori og historie i skandinavisk perspektiv. Lund, Sweden: Studentlitteratur.

Bansler, J. (1989). Systems development research in Scandinavia: Three theoretical schools. *Office, Technology and People*, 4(2). Also in: *Scandinavian Journal of Information Systems*, 1, 3–20.

Bijker, W. E. and Law, J. (1992). *Shaping Technology/Building Society: Studies in Technological Change*. Cambridge, MA: MIT Press.

Bjerknes, G. and Bratteteig, T. (1987). Florence in wonderland: System development with nurses. In: Bjerknes, G., Ehn, P. and Kyng, M. (eds.), *Computers and Democracy – a Scandinavian Challenge*. Aldershot, UK: Avebury, 279–296.

Bjerknes, G. and Bratteteig, T. (1995). User participation and democracy: A discussion of scandinavian research on system development. *Scandinavian Journal of Information Systems*, 7, 73–98.

Bjerknes, G., Ehn, P. and Kyng, M. (eds.). (1987). *Computers and Democracy – a Scandinavian Challenge*. Aldershot, UK: Avebury.

Blomberg, J. and Kensing, F. (eds.). (1998). Special issue on participatory design in CSCW. *Computer Supported Cooperative Work*, 7, 3–4.

Bratteteig, T. (2003). *Making Change. Dealing with Relations Between Design and Use*. Dr. Philos dissertation. Oslo, Norway: Department of Informatics, University of Oslo.

Braverman, H. (1974). *Labor and Monopoly Capital – The Degradation of Work in the Twentieth Century*. New York: Monthly Review Press.

Büscher, M., Shapiro, D., Hartswood, M., Procter, R., Slack, R., Voß, A. and Mogensen, P. (2002). Promises, premises and risks: Sharing responsibilities, working up trust and sustaining commitment in participatory design projects. *Proceedings of the Participatory Design Conference*, Malmö, Sweden.

Buur, J., Binder, T. and Brandt, E. (2000). Taking video beyond 'hard data' in user centred design. In: Cherkasky, T., Greenbaum, J., Mambrey, P. and Pors, J. K. (eds.), *Proceedings of the Participatory Design Conference*, Nov. 28–Dec. 1, 2000 in New York, USA. Palo Alto, CA: CPSR Press, 21–29.

Buur, J. and Bødker, S. (2000). From usability lab to "design collaboratorium": Reframing usability practice. *Proceedings on Designing Interactive Systems*, pp. 297–307.

Bødker, S. (1991). *Through the Interface? A Human Activity Approach to User Interface Design*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Bødker, S., Christiansen, E., Ehn, P., Markussen, R., Mogensen, P. and Trigg, R. (1993). *The AT Project: Practical Research in Cooperative Design* (DAIMI report No. PB-454). Aarhus, Denmark: Aarhus University, Department of Computer Science.

Bødker, S., Ehn, P., Romberger, S. and Sjögren, D. (eds.). (1985). *The UTOPIA Project: An Alternative in Text and Images (Graffiti 7)*. Swedish Center for Working Life, the Royal Institute of Technology, Stockholm, Sweden and the University of Aarhus, Denmark.

Bødker, S., Ehn, P., Kammersgaard, J., Kyng, M. and Sundblad, Y. (1987). A Utopian experience. In: Bjerknes, G., Ehn, P. and Kyng, M. (eds.), *Computers and Democracy – a Scandinavian Challenge*. Aldershot, UK: Avebury, 251–278.

Bødker, S., Grønbæk, K. and Kyng, M. (1993). Cooperative design: techniques and experiences from the Scandinavian scene. In: Schuler, D. and Namioka, A. (eds.), *Participatory Design: Principles and Practices*. Hillsdale, NJ: Lawrence Erlbaum Associates, 157–175.

Bødker, K., Kensing, F. and Simonsen, J. (2004). *Participatory IT Design: Designing for Business and Workplace Realities*. Cambridge, MA: MIT Press.

Clement, A. and van den Besselaar, P. (1993). A retrospective look at PD projects. *Communications of the ACM*, 36(4), 29–37.

Clement, A. and van den Besselaar, P. (2004). *Proceedings of the Participatory Design Conference*. Toronto, July 2004.

Ehn, P. (1988). *Work-Oriented Design of Computer Artifacts*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Ehn, P. (1993). Scandinavian design: On participation and skill. In: Schuler, D. and Namioka, A (eds.), *Participatory Design: Principles and Practices*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Ehn, P. and Kyng, M. (1987). The collective resource approach to systems design. In: Bjerknes, G., Ehn, P. and Kyng, M. (eds.), *Computers and Democracy – a Scandinavian Challenge*. Aldershot, UK: Avebury, 17–57.

Engeström, Y. (1987). *Learning by Expanding*. Helsinki, Finland: Orienta-Konsultit.

Fischer, G. (2001). Communities of interest: Learning through the interaction of multiple knowledge systems. In: Bjørnstad, S., Moe, R. E., Mørch, A. I. and Opdahl, A. L. (eds.), *Proceedings of the 24th Information Systems Research Seminar in Scandinavia* (Vol. 1), Ulvik, Norway, 11–14 August 2001. Bergen, Norway: Department of Information Science, University of Bergen, 1–14.

Floyd, C., Mehl, W.-M., Reisin, F.-M., Schmidt, G. and Wolf, G. (1989). Out of Scandinavia: Alternative approaches to software design and system development. *Human-Computer Interaction*, 4, 253–350.

Garfinkel, H. (1967). *Studies in Ethnomethodology*. Englewood-Cliffs, NJ: Prentice-Hall.

Greenbaum, J. (1993). PD: A personal statement. In: Kuhn, S. and Muller, M. (eds.), *Special Issue on Participatory Design. Communications of the ACM*, 36(4), p. 47.

Greenbaum, J. and Kyng, M. (eds.). (1991). *Design at Work: Cooperative Design of Computer Systems*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Greenwood, D. (1992). Labor-managed systems and industrial redevelopment: Lessons from the Fagor Cooperative Group of Mondragón. In: Rothstein, F. A. and Blim, M. L. (eds.), *Anthropology and the Global Factory: Studies of the New Industrialization in the Late Twentieth Century*. New York, NY: Bergin and Garvey, 177–190.

Hartswood, M., Procter, R., Rouncefield, M. and Sharpe, M. (2000). Being there and doing IT in the workplace: A case study of a co-development approach in healthcare. In: Cherkasky, T., Greenbaum, J., Mambrey, P. and Pors, J. K. (eds.), *Proceedings of the Participatory Design Conference*, Nov. 28–Dec. 1, 2000, New York, USA. Palo Alto, CA: CPSR Press, 96–105.

Hartswood, M., Procter, R., Slack, R., Voß, A., Büscher, M., Rouncefield, M. and Rouchy, P. (2002). Co-realisation: Towards a principled synthesis of ethnomethodology and participatory design. *Scandinavian Journal of Information Systems*, 14(2), 9–30. Republished in Ackerman, M. S., Halverson, C. A., Erickson, T. and Kellogg, W. A. (eds.). (2007). *Resources, Co-Evolution and Artifacts*. Springer Verlag London.

Hughes, T. P. (1983). *Networks of Power: Electrification in Western Society, 1880–1930*. Baltimore, ML: Johns Hopkins University Press.

Hyysalo, S. and Lehenkari, J. (2002). Contextualizing power in a collaborative design. In: Binder, T., Gregory, J. and Wagner, I. (eds.), *Proceedings of the Participatory Design Conference*, Malmö, Sweden, 23–25 June 2002. Palo Alto: Computer Professionals for Social Responsibility, 93–103.

Jacucci, G., Kensing, F., Wagner, I. and Blomberg, J. (eds.). (2006). *Proceedings of the ninth biannual Participatory Design Conference*, August 1–5, Trento, Italy. Palo Alto, CA: Computer Professionals for Social Responsibility(CPSR).

Karasti, H. (2001). Bridging work practice and system design – integrating systemic analysis, appreciative intervention and practitioner participation. *Computer Supported Cooperative Work*, 10(2), 211–246.

Karsten, H. and Jones, M. (1998). The long and winding road: Collaborative IT and organisational change. *Proceedings of the Conference on Computer Supported Cooperative Work*, Oct. 1998, Seattle, WA. New York: ACM SIGCHI, 29–38.

Kuhn, S. and Muller, M. (eds.). (1993). Special issue on participatory design. *Communications of the ACM*, 36(4) 24–103.

Kyng, M. (1998). Users and computers: A contextual approach to design of computer artifacts. *Scandinavian Journal of Information Systems*, 10(1 and 2), 7–44.

Kyng, M. and Mathiassen (1982). Systems development and trade union activities. In: Bjørn-Andersen, N., Earl, M., Holst, O. and Mumford, E. (eds.), *Information Society, for Richer, for Poorer*. Amsterdam: North Holland.

Latour, B. (1999). *Pandora's Hope: Essays on the Reality of Science Studies*. Cambridge, MA: Harvard University Press.

Leont'ev, A. N. (1978). *Activity, Consciousness, and Personality*. Englewood Cliffs, NJ: Prentice-Hall.

Leont'ev, A. N. (1981). *Problems of the Development of Mind*. Moscow: Progress.

Lewin, K. (1946). Action research and minority problems. *Journal of Social Issues*, 2, 34–46.

Luff, P., Hindmarsh, J. and Heath, C. (eds.). (2000). *Workplace Studies – Recovering work practice and Information System design*. Cambridge, UK: Cambridge University Press.

Mackenzie, D. and Wajcman, J. (eds.). (1999). *The Social Shaping of Technology* (2nd ed.). Buckingham, UK: Open University Press.

Muller, M. J. (1992). Retrospective on a year of participatory design using the PICTIVE technique. *Proceedings of the Conference on Human Factors in Computing Systems*, Monterey, CA, May 3–7, 1992. New York, NY: ACM Press, 455–462.

Muller, M. J. (2002). Participatory design: The third space in HCI. In: Jacko, J. A. and A. Sears (eds.), *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*. Mahwah, NJ: Lawrence Erlbaum Associates, 1051–1068.

Muller, M. J., Wildman, D. M. and White, E. A. (1993). Taxonomy of PD practices: A brief practitioner's guide. In: Muller, M. and Kuhn, S. (eds.), Special issue on participatory design, *Communications of the ACM*, 36(4), June.

Mumford, E. (1987). Sociotechnical systems design – evolving theory and practice. In: Bjerknes, G., Ehn, P. and Kyng, M. (eds.), *Computers and Democracy – a Scandinavian Challenge*. Aldershot, UK: Avebury, 59–77.

Nygaard, K. (1975). Kunnskaps-strategi for fagbevegelsen (knowledge strategy for trade unions). *Nordisk Forum* 6, 10(2), 15–27.

Orlikowski, W. J. (1996). Evolving with notes: Organizational change around groupware technology. In: C. U. Ciborra (ed.), *Groupware and Teamwork – Invisible Aid or Technical Hindrance?* Chichester, UK: John Wiley and Sons, 23–59.

Poltrock, S., Grudin, J., Dumais, S., Fidel, R., Bruce, H. and Pejtersen, A. M. (2003). Information seeking and sharing in design teams. *Proceedings of GROUP*'03, London 239–247.

Procter, R. and Williams, R. (1996). Beyond design: Social learning and computer-supported cooperative work – some lessons from innovation studies. In: Shapiro, D. et al. (eds.), *The Design of Computer-Supported Cooperative Work and Groupware Systems*. Elsevier Science, 445–464.

Sandberg, Å. (ed.). (1979). Computers dividing man and work: Recent Scandinavian research on planning and computers from a trade union perspective. Demos project report #13. Stockholm: Arbetslivscentrum.

Schuler, D. and Namioka, A. (eds.). (1993). *Participatory Design: Principles and Practices*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Suchman, L. (1987). Plans and situated actions: The problem of human–machine communication. Cambridge: Cambridge University Press.

Suchman, L. (2000). *Human/Machine Reconsidered*. Department of Sociology, Lancaster University at: http://www.comp.lancs.ac.uk/sociology/soc040ls.html. Last accessed: August 21, 2002.

Suchman, L. (2002). Located accountabilities in technology production. *Scandinavian Journal of Information Systems*, 14(2), 91–105.

Suchman, L. A. and Trigg, R. A. (1991). Unterstanding practice: video as a medium for reflection and design. In: Greenbaum, J. and Kyng, M. (eds.), *Design at Work: Cooperative Design of Computer Systems*. Hillsdale, NJ: Lawrence Erlbaum Associates, 65–89.

Suchman, L. S., Trigg, R. A. and Blomberg, J. (2002). Working artefacts: Ethnomethods of the prototype. *British Journal of Sociology*, 53(2), 163–179.

Törpel, B., Pipek, V. and Rittenbruch, M. (2003). Creating heterogeneity: Evolving use of groupware in a service network. Special issue on evolving use of groupware, *Computer Supported Cooperative Work*, 12(4), 381–409.

Trigg, R. and Anderson, S. I. (eds.). (1996). Special issue on current perspectives on participatory design. *Human-Computer Interaction*, 11, 181–290.

Trist, E. (1981). *The Evolution of Socio-Technical Systems*. Toronto, Ontario Ministry of Labour, Occasional Paper No. 2.

Wenger, E. (1999). *Communities of Practice – Learning, Meaning, and Identity*. Cambridge, UK: Cambridge University Press.

Whyte, W. F. (ed.). (1991). *Participatory Action Research*. New York, NY: Sage Publications.

Winograd, T. and Flores, F. (1986). *Understanding Computers and Cognition*. Norwood, NJ: Ablex.

Chapter 3

Design as and for Collaboration: Making Sense of and Supporting Practical Action

Alex Voss[1], Rob Procter[1], Roger Slack[2], Mark Hartswood[3] and Mark Rouncefield[4]

[1] National Centre for e-Social Science, University of Manchester, alex.voss@ncess.ac.uk, rob.procter@manchester.ac.uk
[2] School of Social Sciences, University of Wales at Bangor, r.slack@bangor.ac.uk
[3] School of Informatics, University of Edinburgh, mjh@inf.ed.ac.uk
[4] Computing Department, Lancaster University, m.rouncefield@lancs.ac.uk

3.1 Introduction

In this chapter, we discuss the relationships involved in the development and practical use of information technologies. That is, we are concerned with the way we build and use IT systems in the context of complex organisational settings – how we create assemblages of hardware and software so as to support the work people do. We will look at the relationship between systems, their design, organisational settings in which they are embedded, and the work they are built to support. As information technologies become a more and more pervasive feature of modern workplaces, understanding the relationship between these technologies and the social organisation of work, of which they are a part, becomes increasingly important for those involved in their creation, deployment, and use.

From the time when the then nascent software industry resolved to adopt the methodologies of more traditional engineering products (Naur and Randell, 1969), it has tended to favour a separation of the development of IT systems from their deployment, use, and maintenance. Furthermore, there is now a well-developed division of labour within the design process itself that separates, for example, 'requirements analysts' from 'programmers'. To some extent, this division of labour is inevitable and perhaps even desirable within a mature industry with relatively stable user requirements and established, effective patterns of technology supply. After all, the aim is to maximise the reuse of candidate solutions to common problems, traded in the marketplace, and produced in an efficient way (see Chapter 9 by Pollock and Williams, this volume). However, as the many and continuing reports of the failures of IT projects attest (Standish Group International, 1995),[1] it is an approach that is

[1] Although the pessimism in the Standish Group's Chaos reports is questioned by some (e.g. Glass 2006). More recent studies report significant (though less severe) problems delivering IT projects to time, budget, and scope (Sauer et al., 2007).

A. Voss et al. (eds.), *Configuring User-Designer Relations*,
DOI: 10.1007/978-1-84628-925-5_3,

not working. As we will argue, a major problem is that it leaves open the question of *adequacy* of solutions and how the specific, detailed working practices of workers can be effectively supported. Only when technologies get translated into systems, only when these get used 'in anger' and encounter the contingencies of the workplace, can we effectively assess their 'fit' with the work that gets done. This poses an important question given that 'design' and 'use' are often separated in time and space as well as being undertaken by different people with different skills, concerns, and under different sets of constraints.

Various devices were invented under the umbrella of *software engineering* to address the problem of improving the communication between, for example, requirements analysts and programmers including representations such as flow charts and data flow diagrams, and emphasising the role of the requirements document in the systems development process. The 'rise of methodology' was brought about by the realisation that systems development did not scale very well as the coordination effort involved in managing increasingly large teams of IT professionals grew out of proportion (Brooks, 1975). Projects often ran late and over budget and adding more staff just added to the problems, leading Brooks to formulate his law that 'adding manpower to a late software project makes it later' (ibid. p. 25).

However, despite the significant effort invested and the various ways that were developed to 'inform design', to control the development process, to provide various representation of IT systems and their context, and to subject them to review and formal verification, a large percentage of IT systems projects still fail outright, while systems are not used effectively or suffer from various dependability problems (Charette, 2005).

We will argue that at the heart of the problem lies a 'missing how', a lack of appreciation by most computer scientists and IT professionals of how people go about their everyday work, be it designing IT systems or using them. While this 'how' is sometimes captured in ethnographies produced as part of academic studies, it has proven to be difficult to bring into the design of technologies (e.g. Plowman et al., 1995; Hughes et al., 1992, 1995) as it is seemingly difficult to distil requirements from these rich descriptions of practice. Various approaches to *informing design* have been developed that aim to address the problem of linking IT systems design and use by either bringing users into the design process in a stronger way (participatory design, discussed in more depth in Chapter 1) or by making representations of work practices available to designers (ethnography for design). In effect, these existing approaches 'build bridges' between the separated activities of 'design' and 'use' and between the people involved. Arguably, they attempt to repair the consequences of boundaries, rather than remove them, which we would argue is the real key to addressing the problem of informing design.

No matter how well we design a system to match a set of requirements determined using conventional methods, there will always be a need for change. First, our understanding of the situation into which a system is to be introduced will inevitably be bounded by our limited experience and subject to certain assumptions we necessarily make. Second, the introduction of the system will give rise to new requirements being formulated as people learn more about its potential uses and opportunities to change practices around the new socio-material arrangements. Finally, the situation of use changes constantly as the world keeps turning. We might say that requirements are 'moving targets' and that change is an inevitable part of IT systems development.[2]

How we make sense of and order the activities around the design and use of systems is an important element of user–designer relations, and it is these practical matters that we discuss in the following sections. Because we believe that the relationship is a symmetrical one, that is, both design and use practices need to be understood to make sense of the design/use space, we will look at both the practices that are concerned with the building of IT systems and the ways in which the working practices of those using systems can be investigated, explicated, and understood.

3.2 Design for Collaborative Work

The field of computer-supported cooperative work (CSCW) emerged in the mid-1980s and early 1990s both as a technical discipline concerned with exploiting networking technologies to enable computer-mediated communication and the development of distributed systems and as a discipline concerned with the sociality of activities involving computer systems. The latter aspect was a reaction to the way in which people and their activities were conceptualised in earlier work on 'human factors'. It is perhaps best summarised by Bannon (1991):

> Part of the problem resides in an implicit view of ordinary people which, if surfaced, would seem to treat people as, at worst, idiots who must be shielded from the machine, or as, at best, simply sets of elementary processes or "factors" that can be studied in isolation in the laboratory. […] Understanding people as "actors" in situations, with a set of skills and shared practices based upon work experience with others, requires us to seek new ways of understanding the relationship between people, technology, work requirements, and organisational constraints in work settings. (Bannon 1991, p. 25)

His call to move on from "human factors" to "human actors" places the active worker at the heart of research and practice, rather than the computer system with people as components that are somehow attached to it. Thinking about people using computers to do their work also brings their activities into focus and we find that they are normally collaborative activities involving a working division of labour rather than being solitary activities involving individual cognitive processes (Anderson et al. 1989, p. 159).

Schmidt and Bannon make the point that collaboration[3] is not restricted to 'group-work' but is a feature of all work activities and that we should not restrict the

[2] This is the case for any kind of development. In some circumstances it may be possible to control change, but this can be done only to an extent.

[3] We will use the terms 'collaborative' and 'cooperative' interchangeably to denote any socially organised work undertaken within an organisational division of labour.

meaning to a particular form of activity within particular work arrangements. Rather, it is the sociality of work in general that is of interest (Bannon and Schmidt, 1991; Schmidt and Bannon, 1992). It is therefore ironic that CSCW has developed at the fringes of computer science as it addresses a central rather than a marginal topic.

The orientation to work as inherently social allows us to treat both collaboration in the sense of cordial work relationships and conflict as aspects of the same phenomenon, that is, the production of social order. That is, seeing work as essentially collaborative does not mean that one negates the possibility of conflict (cf. Kling, 1991) but points to the socially organised ways in which most conflicts normally get managed and dealt with (if not resolved). In any real-world setting, we will find that people engage in collaborative work in that they orient to what others are doing, organise their work so that others can in turn orient to it (i.e., they make their work accountable), they orient to their responsibilities within the organisation, appeal to others to discharge their duties, voice concerns and objections, etc. It is this wider definition of what constitutes 'collaborative work' that we are interested in and we use that in the following.

People routinely make decisions as to which aspects of their work to present to their fellow workers. That is, they will make the results of their efforts available but will hide most details of their accomplishment (Suchman, 1995; Schmidt, 2000). The observation that work is often 'hidden' is therefore not a surprising one but it is interesting in terms of the implications for design and the extent to which requirements can be 'read off' from various accounts of how work gets done. For example, Blomberg et al. (1996) found that work in the litigation support department of a law firm was characterised as highly routinised and involving very little knowledge about legal matters. Subsequent observation of this work then revealed various practices that required significant understanding of how legal documents are organised.

3.3 User–Designer Relations

Communities such as CSCW and participatory design have produced a number of studies that emphasise the social organisation of design work (e.g. Button and Sharrock, 1995a, b, 1996; Bowers and Pycock, 1994; Sharrock and Anderson, 1994; Woolgar, 1991, 1994). While they focus on different aspects, what these studies have in common is that they often discuss the relationship between 'users' and 'designers', for example, as they collaborate to work up requirements for design or focus on the organisation of work within a design project.

Most approaches today share a common set of understandings and assumptions about the nature of IT systems development and the relationship between the development process, IT professionals (designers), and non-IT professionals or workers (users). Drawing on a similar list of common understandings by Greenbaum and Kyng (1991), we would contend that the following are more or less universally accepted in the disciplines named above:

1. Work is socially organised and takes place within a specific situational context and, consequently, so does IT systems use.

2. Design needs to be grounded in an understanding of what the context is.
3. Participation of users in the design process is generally beneficial.
4. Users have knowledge and skills relevant to the design process.
5. The role of various skills changes when IT systems are introduced.
6. IT systems should support working practices and they should support quality work, not merely quantity.
7. The design process is political with at least the potential for conflict.

While there is little debate about these matters in principle, different academic traditions have their respective specific takes on them. However, the big question is how to address and deal with these insights practically. It is here that differences start to appear most clearly. While some approaches focus on integrating the 'user' – as a 'human factor' or 'human actor' (Bannon, 1991) – into an existing design process, others go further in their suggestions that the design process itself has to change in more or less fundamental ways.

3.4 Understanding Practice – 'Informing Design'

In the following sections we look at the role that ethnographic studies have come to play in IT systems development, and discuss the problem of distilling from detailed observations of practice a set of requirements for systems development.

3.4.1 Ethnography for Design

Ethnography is a form of study reportage developed in the social sciences that presents data about social life, usually obtained through observations, in a way that makes it available for analysis. This is achieved through first rendering the phenomena reported on strange and interesting, and then recognisable through examining their orderliness (Anderson, 1994). There are many forms of ethnography developed in different disciplines within the social sciences which have in common their emphasis on producing a naturalistic description[4] of social life and its conduct in the natural context, often taking an 'appreciative stance' (Rouncefield 2002, p. 71) that seeks to recover the way things are seen by the members of the setting. However, approaches differ significantly in the way in which data is obtained and how it is analysed. Indeed, some forms of analysis are based on categorisations or theoretical frameworks, which are defined a priori by the researcher. As Randall et al. (1994) point out:

[4] We choose to avoid the term 'thick description' *à la* Geertz (1973) because of its various connotations, which are not of interest in the context of this chapter (cf. Ortner, 1997).

> […] the appropriation of ethnography as a method of investigation in CSCW has not so far been accompanied by the necessary attention that needs to be given to issues such as *what kind* of ethnographic practice might be suitable for the task of gearing into the procedures of eliciting requirements, or how its analyses and descriptions can be related effectively to systems design (ibid. p. 242, emphasis in original).

We find it important to choose an approach that aims to preserve the integrity of the phenomena observed by adopting a non-ironic stance that does not substitute them with categories defined *a priori*. Such an approach avoids the problem of losing the phenomenon itself and finding only the analyst's objects. Therefore, when we speak of ethnographic studies or ethnographies in the following, we mean by this a specific form of study informed by ethnomethodology (Garfinkel, 1967; Heritage, 1984) which is based on a concern for identifying and explicating the ways in which social order is produced (the social organisation of work) and how people can act in meaningful and mutually intelligible ways as they go about their everyday business.

Ethnographic studies of work in various settings have been instrumental in uncovering the seen-but-unnoticed aspects of work that have so often escaped attention in requirements-gathering exercises and have therefore not been supported in the resulting systems designs. Even worse, systems designs are often in direct conflict with the organisation of work, as demonstrated by Bowers et al. (1995) study of the introduction of a workflow system on a print industry shop floor. These studies have been of great scientific and educational value, sensitising people to the kinds of phenomena of everyday work that are so easily missed in IT design. They have demonstrated the social character of workplace activities (even seemingly solitary ones) and have put the issue on the agenda once and for all. The success of ethnographic studies of work in this respect has instilled an interest in their use as a means for requirements capture. However, the problem of how best to incorporate findings from studies of work and technologies into IT systems design processes remains a matter of ongoing debate (e.g. Hughes et al., 1992–1995, 2000; Randall et al., 1994; Plowman et al., 1995; Schmidt, 2000). Blomberg et al. (1993) point to the different projects of ethnography and design:

> While the ethnographer is interested in *understanding* human behavior as it is reflected in the lifeways of diverse communities of people, the designer is interested in *designing* artifacts that will support the activities of these communities. The current challenge is to develop ways of linking these two undertakings. (ibid., p. 123)

In addition, there are common misconceptions amongst many wishing to employ ethnographic studies as a means to inform design. In the social sciences, ethnography is a particular 'form of reportage' (Anderson, 1994) rather than a method for data collection or an analytic approach. In contrast, the use of the term in areas such as HCI and CSCW often presents it as combination of data collection, reportage, and mode of analysis without further specifying either of those elements. Designers and many researchers are largely ignorant of the nature of ethnography in the social sciences, and, in practice, these matters do not impact on their work (cf. Anderson, 1997).

Very often, ethnographic studies, while providing rich descriptions of working practice as it exists, leave the question 'so what?' unanswered. Design implications, if any, are often quite vague in nature and few projects have managed to bring ethnographic observation and technology design together in a convincing manner. The problem of envisaging the future is notoriously difficult to solve as the real implications of any design decision can only be revealed over a period of time and through using the system in anger. While ethnographies provide an account of current working practices, they do not address this problem. As Jirotka et al. (1992) have put it:

> […] although ethnographic analyses of interactions in the workplace can highlight systematic, and often robust, features of work practices, they do not and cannot conclude either that these features *should* be preserved or that they *will* be preserved when new technology is introduced. (ibid., p. 112, emphasis in original)[5]

Furthermore, ethnographies cannot reveal what other features might or might not emerge from a particular implementation. Therefore, what ethnography can offer for IT systems development is at best a partial solution that needs to be worked into practices, which allow the implications of technological intervention to be worked out and for decisions to be made on that basis. Many studies have drawn attention to the fact that technologies rarely emerge from nowhere and that they cannot be simply adopted but have to be appropriated by their potential users (Procter and Williams, 1996a, b; Williams, et al. 2005; Stewart and Williams, 2005). This appropriation involves work to make technological offerings 'fit' their intended purpose and context. Innovation therefore happens long after the 'design' of a system has finished as people grapple with its affordances, while facing the contingencies the world confronts them with. This work of appropriating technologies is inherently a social one as various studies of computer used in real-world settings have shown (e.g. Nardi and Miller, 1990; Williams et al., 2005).

What is it, then, that ethnographies can provide for design in the light of these circumstances? Indeed, the question to what extent designers should be interested in *analytic ethnography* (Anderson, 1994) is an interesting one – what is it that designers are interested in and should they burden themselves with the different interests and commitments that social scientists have? We would argue that rather than trying to appropriate analytic ethnography as practiced in the social sciences, IT systems designers should take their own aims and commitments seriously, they should take an interest in what people know and use, how they go about their day-to-day work,

[5] We concur with Jirotka et al.'s point to an extent. We would agree that ethnography by itself does not provide a means for unambiguously settling such questions, but would argue that ethnography can *help* with the work of envisaging the future. See the conclusions of this chapter for further discussion on this point.

and how they make sense of the actions of others. But, and this is crucial, they should pursue this interest on their own terms, to pursue the aim of informing design decisions. As Anderson (1994, p. 155) puts it:

> It is simply that you do not need ethnography to do that; just minimal competency in interactive skills, a willingness to spend time, and a fair amount of patience.

Realising that the agendas of the social sciences and of design are different but may require a similar attention to how people do their work, one can then develop a programme for the study of such activities, which is *adequate* in regard to the object of study *and* the interests of design. All this is not to say that ethnographic studies of work do not have a role to play in the project of informing design but that they are useful in a different way.

> What ethnography may offer designers concerned with productivity is not just detailed description of work routines and daily life with which to fix the features of the design, but an opportunity to open up the overall problem-solution frame of reference in the context of some proposed solutions to specific identified problems. […] In other words, the contribution that ethnography may make is to enable designers to question the taken-for-granted assumptions embedded in the conventional problem-solution framework. (ibid., p. 170)

Working up requirements then is still a problem to be resolved. Ethnography has something to say about ‘is’ but cannot provide (by itself) how things ‘ought’ to be. The point, then, is not to treat ethnography for design as a requirements capture method that will provide a specification, but to treat it as a device for fostering what Anderson (1994) calls ‘design sensibilities’:

> [There is a] presumption that to be of value to designers, any description must be couched in a formalized or semi-formalized notation of some kind: as if design consisted in jigsaw-puzzle solving and only certain shaped pieces were allowed. The age-old (and tired) prescription versus description debate, with the ethnographers staunchly appearing to refuse to be prescriptive in the face of designers’ demands for requirement specification. What seems to be being missed here is the extent to which design involves sensibilities as much as models and predictions, programs and prescriptions. (ibid., p. 152–153)

In order to make these sensibilities useful for a design process, they have, of course, to be brought to bear in and on specific contexts. The opposition of ethnographer and designer in how far they are willing to enter each others’ territories is an issue we consider in greater depth later in this chapter. The production of a formal requirements document, where this is informed by detailed understandings of the workplace phenomena is not something that overly troubles us, but a strict division of labour in the production and consumption of such a document does. Of course, various representations may be created for practical purposes, for example, as an *aide memoir* (cf. Schmidt and Bannon, 1992), but it is important that the design itself is informed by the rich understanding gained rather than by an impoverished version of it.[6] One possible way of doing this is demonstrated by Blomberg, Suchman, and Trigg in their approach to case-based prototyping, which relies on an iterative process of ethnographic observation and design work. Similar work has been undertaken by Büscher et al. who draw on a wide range of participatory design methods to establish a process of long-term engagement between designers and (potential) users of novel technologies developed as part of the project (Büscher et al., 2000, 2001, 2004).

3.4.2 Doing Ethnography for Design

There are practical problems related to the study of work in the context of IT systems development, which will have to be tackled by those conducting an ethnographic study. While the use of ethnographies to inform design is advocated by many, there is a clear lack of guidance on how to actually conduct such a study (for whatever reason).[7] This is especially problematic as many who conduct these studies with a view to inform design will not have a social science background, but rather a technical one. This is not to say that 'doing' an ethnographic study would require skills that are not available to these people. After all, ethnographers rely on mundane skills that we all have – observing, recording, sorting, formulating – but since those using it to inform design will not have been exposed to ethnographic accounts, they will find it even more difficult to know what to look for and how to go about conducting their study. It is therefore worth reviewing what various people have written on this matter.

Harper (2000) makes some recommendations for people using ethnographic methods for design. His first recommendation is to attend to the flow of information as a means to investigate the scope of the study and to ensure that no important activities are missed. Note that in making this suggestion Harper intends to provide us with practical means to achieve a particular purpose but does *not* suggest that the 'information flow' is necessarily an essential feature of the fieldwork or the analysis, nor is it all that is going on. In this, Harper differs from recommendations made by, for example, Beyer and Holtzblatt (1998) who use 'information flow' as an analytical concept rather than a practical device. The observation that information is being worked up, passed from one person to another, sorted, rewritten, recipient designed, discarded, etc. is a recurrent feature of work in most settings (Harper, 2000). This is why the flow of information in its various forms can be used as a guide to investigate a setting, to gain a broad understanding of what is going on – but it should not be confused with the phenomenon itself, that is, 'just how' people go about these activities.

---

[6] In relation to formal requirements documents: we would not want designers to have to rely solely on formal requirements specifications, but be able to interrogate them as appropriate in light of a more detailed understanding of the phenomena they either represent or have emerged from.

[7] Crabtree's (2003) recent admirable book on the practical application of ethnography for design goes a considerable way to filling this gap.

A second recommendation, relating to 'ritual inductions' has as much to do with the problem of gaining access and being accepted as an 'insider' as with gathering information about the setting. While taking part in routine activities is the basis for an engagement with the setting and the basis for coming to be seen as an 'insider', there are opportunities to take a further step and to demonstrate commitment, respect, and a genuine interest in what people do. 'Doing a nightshift' is an example of this, as is accompanying members on 'missions' that take them outside their normal place of work in the organisation. Researchers have found that taking part in such activities can help the ethnographer to gain acceptance, to get access to aspects of the work previously inaccessible, and to learn more about what members would consider important. In general, sharing peoples' concerns and taking part in their activities as members of the setting (e.g. Christmas parties) is an important part of becoming an 'insider' and getting accepted and trusted. The ways in which this can happen will be specific to the setting and may be more or less obvious.

Harper's third recommendation is to pay close attention to detail. Details of how work is accomplished can often be retrieved through direct observation. When an ethnographer interviews members of a setting or asks them questions while observing their work, they will often comment that their work is not interesting, that the ethnographer would not want to know about it because it is so mundane. This leads to the ironic situation that the 'stuff' that the ethnographer is interested in, namely, the details of work's accomplishment, is deleted from accounts. Members will instead often talk about versions of 'how things should be' or will use glosses to cut short what might seem to them to lead to a complicated and tedious account. It is therefore an important task for the ethnographer to remind interviewees that the details *are* of interest and to ask them to unpick what they say and provide more details. However, again in contrast with Beyer and Holtzblatt (ibid.), this should in no way take the form of disregarding or correcting the informant's accounts. Rather, the task is to convey to members of the setting that their work is taken seriously and that they are invited to provide as much detail as they care to provide, that is, they can suspend the usual courtesy of not bothering others with the details of how they accomplish their work.

Another question that arises in relation to how ethnography for design is done concerns the choice of fieldwork method used. In some settings, participant observation will be possible (one can probably take over some office work under instruction and guidance), but in others, such as in medical work or air traffic control, this might not be possible and one might be limited to an observational role. Sometimes, it will not even be possible to use observation (participant or not), for example, for ethical reasons and other means of gathering data will need to be found that are appropriate (cf. Hemmings et al., 2002). In addition, the degree to which the ethnographer might partake in the work can vary depending on circumstances, for example, it might depend on workloads, the complexity of a case, etc. The question of what to observe, for how long, when to participate, when to ask questions or interview someone, etc. can only be answered by referring to the situation studied and the purpose of the study, be it to add to academic knowledge or to inform design or some other intervention.

Where the ethnography is produced by social scientists, the problem of collaboration between them and designers arises. As Randall et al. (1994, p. 248) suggest:

> [This involves] procedures that, while not *ad hoc*, are, nevertheless, to be viewed as practical responses to problems encountered in an evolving collaboration between sociologists and system designers. In the absence of a universally acceptable method of subsuming descriptions of cooperative work into the requirements analysis process, it could hardly be otherwise.

There is no 'silver bullet', no machinery that will translate ethnographic accounts of working practices into requirements for design. One approach is collaboration between designers and ethnographers aiming to work up requirements, which can be said to be grounded in the ethnographic account. Randall, Hughes, and Shapiro offer no universally applicable method but by offering their experiences as an example, they invite others to use this as a resource in reflecting on their own situation and developing methods that are adequate to the situation they face. We return to these points later in the chapter.

3.4.3 Typifications and Patterns

One way in which ethnographies can inform design is through the use of typifications, either of technologies or of phenomena observed. Trigg et al. (1999) provide an example of the former: they discuss a number of issues that arise in the introduction of a particular class of systems, namely, document management systems. While the specific findings provided by their study are interesting for CSCW researchers, they do not provide requirements that might inform the design of document management systems in general, or an implementation of such a system in another setting. However, the authors provide a set of questions one might ask in relation to a setting in which such a system might be introduced. These questions are motivated by the observations made within the study setting but are presented as a question one might ask of any 'typical' setting in which documents are managed. By presenting the study setting as a 'typical one', that is, other settings will be roughly similar but will differ in detail, the step is made from a specific study to something that is more generally useful. It is not that requirements are formulated in a ready-to-use form but the formulation of questions emerging from the study might allow a designer to relate the questions or perhaps even the details of the study to the setting they are interested in.

A related approach is to distil a set of *patterns* from ethnographic studies (Erickson 2000; Martin et al. 2001; Martin and Sommerville 2004) in order to make the findings of ethnographic studies available as a 'background for understanding or characterizing work in different settings' (Martin and Sommerville 2004, p. 62). By making findings of ethnographic studies available to designers in a standardised format and pointing out to what extent they are *repeated findings*, patterns aim to make the body of workplace studies more accessible and usable for the purposes of design. As with other forms of indexing or abstracting, the value ultimately lies not in the systematic presentation itself but in the fact that it makes a more substantial resource more easily accessible. Consequently, Martin et al. include in their patterns pointers to the original studies. Patterns are orienting and organising devices that provide topics for investigation and ways of relating findings to similar findings from other settings, opening up the possibility of comparison and contrasting.

The way that a setting is arranged will always be reflexively tied to the way work is accomplished and workers will always have ways or coordinating their work. As Martin and Sommerville (2004, p. 63) point out:

> […] In any given setting *just how* coordination is achieved in relation to *what*, and in *what* ways layout affects, facilitates, or constrains activities still remains to be discovered.

The question to what extent these typifications have any purchase, and how it is that they are relevant (or not) to the situation at hand still needs to be answered by those who use patterns to help them inform their design activities. Being able to access workplace studies that can be seen to be of relevance to the situation at hand can serve as a sensitising device that makes certain features of work activities more readily available. The patterns that Martin et al. have collected[8] demonstrate this principle. For example, the notion of 'artefacts as audit trails' points to the

> […] way in which an artefact can serve as a stratified record of work. In this way the artifact serves as a means of coordination between workers allowing them to locate who has done what work and therefore assisting in remedying problems and so forth. It focuses on how amendments and attachments to the artefact, such as comments, date stamps, post-it notes, other documents and so forth, are accountable to the personnel within a setting. These annotations are accountable in that they readily afford information to these competent members about the process through which the artefact has progressed in the workplace. Actors are able to recover the process through viewing the artefact, seeing who has carried out work, when and why using their local knowledge of the setting and work practices.
> Pattern: Artifact as an audit trial. In: Patterns of Interaction: a Pattern Language for CSCW. Computting Department, Lancaster University http://preview.tinyurl.com/68gv63 (accessed 06.08.2008)

In addition to this description, the pattern further contains a short note on where and how the observed pattern may be of relevance, what its implications for the dependability of systems might be as well as pointers to two studies that informed the formulation of the pattern, and provide further resources. These are presented in abstracted form as 'vignettes', which summarise the findings and link to the original publications of a study of work in an entrepreneurial firm (Anderson et al., 1989) and work in air traffic control (Hughes et al., 1992, 1993).

[8] http://www.comp.lancs.ac.uk/computing/research/cseg/projects/pointer/patterns.html

3.5 IS Methodology: Prescription, Process, and Evolution

Early approaches to systems development following the waterfall model (Royce, 1987) were designed to support a strict progression from analysis and requirements specification through design to implementation. Such models relied heavily on the formalisation of design documentation and the handover process that interfaced the phases of the development process. The danger inherent in such methods is that they get followed to the letter in a highly bureaucratic manner and end up getting abandoned because of the overheads involved and the unresponsiveness to circumstances. Use of systems development methodologies can become more of a ritual than a practical means for accomplishing work and can even be detrimental to the overall project (Wastell, 1996). While systems development methodologies provide the means for management to gain (at least a sense of) control over systems development projects, they often fail to support adequately the work involved in doing the project's work, especially when they are followed rigidly. In the face of these problems, many organisations start 'tailoring' the processes to their own needs, prioritising the aspects that are seen to add most value to the task at hand, and picking the methods and tools that are readily appropriated (Wastell, 1996; Sharma and Rai, 2000). The common conception that methodologies can act as a vehicle for 'best practice' that can replace experience and skill and can provide a template for practice that can be readily adopted is highly questionable (Russo and Stolterman, 2000; Nandhakumar and Avison, 1999) as it ignores the work that goes into appropriating methods and tools and using them practically, according to the purpose at hand.

While linear systems development methods can be made to work even in the face of evolving circumstances, there is still a tension between the method's prescription and its practical use. In the 1980s, it became increasingly clear that the linear approach to systems development did not work well in many circumstances. Consequently, models incorporating feedback loops, circular, iterative, evolutionary models (e.g. Boehm, 1988; cf. Sommerville 2001) started to become popular. While retaining the notion of a phase and therefore addressing the problem of controlling the software process, they introduced the notion that systems development was a learning process and that what was learnt would have to be made available and factored into the design. The main rationale for the introduction of iteration into the software development process was the recognition that initial specifications were imperfect, that there was an inevitable drift between the static requirements specification and the changing world and that repairing these problems early saved effort and money. This realisation that software needs to evolve to meet the changing needs of a society that increasingly relies on it has led to the formulation of software evolution as a distinct field of study (Lehman, 1998).

More recently approaches to systems development have emerged under the rubric of 'agile' or 'lightweight' that are explicitly aimed to support systems development in a changing world. Some of them have become quite popular as an answer to the rigidities of traditional-phased design methodologies, the most notable being extreme programming (Beck, 1998, 2000). Extreme programming (XP) focuses on the minimisation of risk during the systems development process and employs strategies that are often opposed to traditional systems development and project management wisdom. Key elements are a focus on working code involving early-release and short-release cycles, an incremental planning approach that allows changes to be made according to changing circumstances, a reliance on automatic testing tools to ensure quality and on tests, code and communication to represent system structure and intent, as well as a commitment to ongoing design throughout the system lifecycle (Beck, 2000, p. xvii). By delivering value and spotting problems early and by emphasising simplicity, XP promises to deliver high-quality systems and flexibility in the light of changing needs.

While agile methods like XP have become quite popular, it is not clear how well they scale and integrate with other methods (Boehm and Turner, 2005; Turk et al., 2002). It remains to be seen whether they represent merely a fashion or a rethinking of traditional, top-heavy systems development methodologies that will reshape the ways we build systems. In contrast to previous promises, XP at least seems to be a key departure in that it is not overly prescriptive and actively encourages designers to fit the method to their specific needs or to consider alternative approaches if the conditions are such that they would not allow XP to work. It would seem that agile methods at least provide occasion for the software engineering community to consider a new way of socially organising development work and relationships between IT professionals and their 'customers'. This consideration of practice will hopefully lead to a less fashion-driven, more considerate approach seeking to match methods to the problem at hand as well as the larger organisational context.

3.6 Design as Collaborative Work

Developing IT systems is itself a collaborative activity carried out within a particular working division of labour. One might therefore assume that there would be a substantial amount of studies of development work to be found in the software engineering literature, providing an insight into the practices of the various professionals who involved in IT projects. Unfortunately, this is not the case. Of those studies that do exist, most report on *exceptional* circumstances such as the application of a particular methodology within a research project or a laboratory setting or are based on survey methods (e.g. Curtis et al., 1988).

To what extent such studies can be relied on to provide us with an understanding of design work 'in the wild' is questionable as the researcher intervention necessarily distorts the picture. As Bansler and Havn (1991) have noted: 'At best, comparison and evaluation of different systems development methods is based upon, what you might call "laboratory experiments". […] Although these types of experiments yield valuable insights into the virtues and shortcomings of the tested methods, they do not tell us very much about how systems development is practiced'. We would like to add to this comment that such laboratory experiments can only find what has been made findable as part of the experimental design. Given the repeated complaints that practitioners fail to adopt 'best practice' appropriately, it is surprising that studies of actual development *practice* are still so thin on the ground and are to be found within the CSCW literature rather than within software engineering.

Studies of design work have focused on different aspects such as the organisation of talk in problem solving (Button and Sharrock, 2000), the organisation of source code and the use of guidelines (Button and Sharrock 1995b), the work of 'recomposition' (or integration) and the management of source-code dependencies (Grinter, 2003; Sharrock and Anderson, 1993), organising the work as a project (Button and Sharrock, 1994, 1995a, 1996), working up and negotiating requirements (Bowers and Pycock, 1994), maintaining an overview over both the product and the process (Bjerknes and Kautz, 1991), dealing with plans that do not work out (Rönkkö et al., 2005), and instantiating the rules of a methodology within an organisational context (Button, 1993; Button and Sharrock, 1998).

It is important to recognise the detailed, situated practices that people engaged in systems development do. These practices *establish* the more or less routine ways in which systems development is normally undertaken. While development work may be officially undertaken under the regime of a particular methodology (as mandated by the organisation), it is not the rules of the methodology that by themselves provide for the orderliness of the development process but everyday ordinary activities of people. For example, Grinter (2003) describes a number of practices aimed at overcoming the problems of managing dependencies in software projects and attending to the problem of delays. In one of the settings she studied, overall progress was discussed at weekly meetings and where delays were caused by dependencies involving other departments, the department head would follow up by arranging meetings with her peers:

> When the monthly review did not immediately follow the weekly review, she would return to her office and begin scheduling individual appointments with her peers to discuss delays. She did this face-to-face over lunch when she could, but in some cases her peer was in another state or continent and a phone call had to do (ibid. p. 307).

It is the work of preparing and holding the meeting, and following up with meetings addressing particular issues that allowed the organisation to achieve one of the core aims of software development, namely, 'measured progression' (cf. Button and Sharrock 1996). The extract also shows the department head going about her business in a way that is appropriate in the light of the circumstances. Rather than employing more formal means, she seeks to resolve issues in a non-confrontational way 'over lunch' and resorts to phone calls where necessary.

The organisation studied had a policy of non-escalation, that is, there was a strong preference to seek a resolution to problems locally, amongst peers before calling on someone higher up the hierarchy to arbitrate:

> The process of engaging a more senior manager was known as "escalation" and implied that all possible negotiations among peers had not resolved the dependency, and that there were problems with that dependency that had to be made visible to management. Sometimes team leaders self-reported that they were delaying other people with particularly difficult problems. Open admissions tended to occur when a delay had only just emerged. Early admission, with a precise technical description of the complexity of the problem, often

> appeared to encourage other team leaders who depended on this code to help out with suggestions for possible design solutions (Grinter, 2003, p. 308).

Again, we see how the principles of good practice are realised and, moreover, how people orient to the organisation's 'moral order' (e.g. the rules and responsibilities around the use of technologies) and how they demonstrate competence in the work they are doing. By acknowledging the existence of a problem early on and making the details of the problem available to their colleagues, team leaders can demonstrate their competence, which lies in recognising the existence of the problem in the first place and describing its features. Offering the work of finding a practical solution to be undertaken in a collaborative way is a means to show professional respect and gives others the opportunity to raise issues themselves (e.g. how a proposed solution may impact their work).

In the work of Rönkkö et al. (2005) we can see how plans are oriented as objects of negotiation, and how changes are made in the light of circumstances. However, does this mean that the original plan is simply abandoned, or that the process is chaotic. Rather, changes are made in a systematic way in the light of a need to come up with a viable and agreed plan for the future. Various issues are raised and discussed, for example, the degree to which the prerequisites for doing a particular job will be met: 'will the prototype be in a state to make testing both feasible and meaningful?' Resources are also an issue as their planning will need to be matched to the workplan. In the action of changing a plan in the light of circumstances we can see clearly the relationship between plans and situated action (Suchman, 1987). As Sharrock and Anderson (1993, p. 159) put it:

> The carrying out of work is a matter of constant estimation: how much work is there to do, who is going to do it, how many people, for how long, doing what, needing what, with what assurance of success, and with what eventual product? It frequently turns out that the work does not go as estimated, very typically that it takes longer, is more uncertain of outcome, is more problematic, requires different personnel than have been estimated and resourced, but finding that the carrying out of the work is problematic is another of the 'normal natural troubles' of this work.

Recognising the existence of a problem and finding a practical way to deal with it is part and parcel of the work, a routine activity that does not normally occasion anything other than the normal practices of dealing with problems recognisable as those faced before and overcome for all practical purposes under the given circumstances.

Button (1993) describes the case of the development of a photocopier system involving hardware and software design that were mutually dependent but were following different trajectories. In order to allow the hardware engineers to do their work, the software team had to release software which was not developed according to the strict principles of the mandated software development method (Yourdon and Constantine, 1979, cited in Button, 1993, p. 36). Instead, they followed a strategy that allowed them to produce interim releases of the software that could be handed over to the hardware engineers. By using this strategy, the software team can be said to have preserved the integrity of the Yourdon method while attending to the particular circumstances they faced. That is, although they were aware of the ways of ‘breaking the rules’, they were also aware of what needed to be done to re-establish a state of affairs that could be seen as being in line with the requirements of the method. In this way, they could make their work organisationally accountable as work done under the Yourdon regime.

The studies cited above contain many more perspicuous examples that illustrate how an orderly systems development process is produced through various mundane activities. The point here is to note the existence and availability for study of methods of producing order in software development that are not only rules prescribed by some methodology but are also indigenous, locally relevant methods (or *ethnomethods,* cf. Garfinkel, 1967) of systems developers. Orienting to what people know and use, how they draw on various resources in their everyday work lies at the heart of understanding the collaborative achievement of a system’s development.

As Berg and Timmermans (2000) point out in relation to the use of formal tools (protocols, guidelines, etc) to organise medical work, we can see how formal project management or design methodologies depend on various sorts of informal, heterogeneous work to make them work in practice. This does not diminish their potential usefulness, but underscores that they cannot be useful if blindly and rigidly applied. To take advantage of a tool’s formality, one has to be sensitive to the local contingencies of its application. Project management and software design methods provide affordances for organising and coordinating work, for spotting dependencies, meshing the delivery of interdependent components, for bringing problems to the fore and ensuring that responsibilities are discharged and actually help ensure that large-scale organisation of projects can happen at all. But it is the skilled use of formal tools and methods that helps realise their power – otherwise anyone could deliver a software project simply by following the recipe. Following Berg and Timmermans again (ibid.), we would argue that it is important neither to be blinded by the tools, methods, and methodologies themselves, and nor by the ad hoc and heterogeneous work that erupts around them. It is not software or design methods *per se* that are at issue, but rather the *sorts* of method, their foci, and emphasis that we should question most strongly (but would argue that refinement of any prescriptive method needs to have an eye on the realities of its application). We would want to advocate methods and approaches that privilege understanding the ‘how’ of use practices; are unencumbered by ideal-typical models of how work should be conducted; to co-locate users and designers; involve users as early as possible in design; defer critical design decisions as late as possible; foster designer accountability; and reuse experience of designing for the target setting.

The studies outlined above (and others in this volume) reveal the complexities inherent in managing any software development enterprise, and the role of design and management methodologies in making the various aspects of the process predictable in ways that help meet wider organisational concerns (e.g. competitiveness). We return to this point later where we ask if ethnographic approaches need to be organised and applied in ways that make them sufficiently predictable to be acceptable as a manageable software project activity.

3.7 Meshing Ethnography and Design Practice

While there is a clear agreement that designing with an eye, sociality of work is an important (if not central) to helping deliver useful and usable software products, and that ethnography is a valuable tool for achieving this, as we have noted, there seems to be less certainty about precisely how ethnography can be effectively meshed with design. Two main points of difficulty concern what ethnography can be expected to deliver (e.g. concrete design recommendations or opening alternative design spaces), and how to effectively communicate its findings (e.g. rendering field notes relevant to design considerations). One possible explanation (and without wishing to suggest that ethnography is a universal solution) is that these difficulties or possibilities have less to do with intrinsic shortcomings or affordances of ethnographic approach itself, and more with how it has responded in the different contexts and circumstances in which it has been applied.

3.7.1 Concrete Requirements and the Need for Change

To suggest that ethnographies cannot help highlight aspects of practice that should change and those that should be preserved are perhaps to overstate the case. To take just one example, in their study of an airport control room work, Bentley et al. point to practices and use of artefacts (particularly the visibility of the manually organised 'flight strips' that provided an overview of current and future activity in airspaces) and argue that a strong case can be made for leaving these undisturbed – or providing similar functionality in an electronic version (Bentley et al., 1992). It is not hard to conceive of situations where there is widespread agreement within the workplace, both on the need for change and the character of those changes. Or indeed, agreement about which practices need continued support, and those which should be radically altered. If matters of change are not opaque to workers, then there is no reason why they should be to ethnographers or ethnography. Perhaps, then, it is not change *per se* that is at issue for ethnography, but rather those occasions where issues around change are thorny – either because it is hard to see what should be changed, where proposed or possible changes are contentious, or where some, or all, alternatives imply negative consequences for some of those involved (e.g. redundancy – which may prick the conscience of the ethnographer). Arguably (but perhaps dubiously so) ethnography performs better at problematising the proposed changes rather than making positive recommendations, but this should perhaps be seen as a strength in so far as the problems raised are to do with aspects of the sociality of work that have been overlooked. We would agree, though, that ethnography can never *by itself* resolve issues of what should be changed, but argue that if we thought ethnography could, we would always be asking too much of it, because these are issues that are always and inevitably worked out as part of the wider organisational politics in which design is inevitably embedded. There is no reason, however, to expect that ethnography or ethnographers cannot provide details that cast light on the choices to be made, make recommendations, suggestions or caution against certain courses of action. Indeed, if we did not expect ethnography to influence design in these kinds of ways, then we would be asking too little from ethnography and ethnographers.

The question as to the role ethnography can play in design, whether it is best placed to provide requirements for software engineers, or to delineate alternative design spaces is perhaps too chimerical. Our own experience of using ethnography to inform design in a number of projects suggests that it can do both.[9] Again, the issue may not be that ethnography cannot, in principle, help define requirements at different levels of granularity, but that in some circumstances it is harder for it to do so than at others. Requirements arise from interplay between understandings about work (its conduct, values, and ambitions) with understandings about prevailing technological, organisational, and political possibilities and constraints. A detailed account of the conduct of work may immediately 'suggest' certain sorts of technologies as potentially useful, or guide the configuration of a pre-selected software package, although there is no reason to suppose that it will be able to do this in every case. It is perhaps just as likely that an ethnographic account leaves ethnographers and designers alike wondering over its implications for design. This does not mean to say the implications are not there (though they of course might not be) but just that it is hard to make the leap of imagination (perhaps based upon knowledge of how things have played out in similar circumstances elsewhere) needed to see the ethnography's relevance. That is to say that requirements are rarely 'read off', but are much more likely to have to been 'worked up', with the results depending on the skill and commitment of both the designer and ethnographer in relevancing the enquiry and its findings to the situation at hand.

3.7.2 Design Sensibilities and Ethnography

Use of ethnography potentially requires a further division of labour, that between specialist ethnographer and design team, which can create its own problems. In a project to inform the design of a prototype e-Science system, Warr et al. highlighted 'cultural differences' between the usability team (undertaking ethnographies) and the implementation team, charged with delivering working prototypes (Warr et al., 2007). The former expected the ethnographic findings to be the basis for discussion with the implementation team, who themselves were expecting requirements in the form of a recipe they could code to – the problem here is not description versus prescription, but an unwillingness to participate in the process of shifting from one to the other. While this points strongly to the need for designers to be open to the contribution of ethnographic data, the converse also holds that ethnography specialists have to be sensitive to the circumstances in which designers find themselves. It may well be possible to 'scare off' developers if we expect them to make sense of a mass of undifferentiated ethnodata (or a draft of usability requirements they have no hope

[9] An example of the former concerned the creation of a contacts database for psychiatrists working in a hospital ward, where a prior ethnography suggested both the need for such a database, as well as how it might best be organised and searched (Hartswood et al., 2000). An example of the latter concerned the use of a decision aid for mammography. Examining the details of breast screening practice showed up the tool's simplifying assumptions and suggested alternative avenues for design (Hartswood, 1999).

of addressing fully) – if we do not help with the sorting and sifting process. But, of course, and following Warr et al., if designers refuse to play the game, then all bets are off.

It may be that by the time ethnography specialists become involved in a design project, many decisions are already made that effectively limit the ability of developers to act upon ethnographic data or findings (e.g. Hartswood et al., 2005). The scope of the project may have already been fixed, decisions already made about one or more issues such as technology supply, configuration, project management, objectives, and delivery dates. It may be that the contribution of ethnography in these circumstances would be modest.[10] So while there are issues of working cultures, expectations about divisions of labour, established software development practices, and so on, it is also likely that in order to be receptive to ethnographic findings there has to be a sort of 'operational readiness' to take on-board the insights they have to offer – that certain sorts of finding become useful and actionable, and when the technical work has reached a particular stage in its maturation – where findings become apposite to the developers' technical concerns rather than competing with them. That is to say developers have to see not only the relevance of the findings, *but also to be in a position to do something about them*. We have to be pragmatic with our expectations from ethnography, as with any 'first pass' at a new system, it is unlikely that all the concerns of usability and use are likely to be met, as various technical compromises will inevitably have to be made to get the system working at all that will rule out meeting certain end-user requirements 'this time around'. Sometimes, we can at best only have an eye for building things in a way that makes for a smoother transition to a second version, where more usability objectives might be met.

One further question that arises in the question of meshing ethnography and design practice is how ethnography can be rendered predictable or manageable as a project activity such that a project manager can make judgements about, for example, the time it might take, the resources it might consume, and the benefits that might accrue. Ethnography does have certain properties that make it seem unruly, and perhaps even threatening such as (1) that ethnography (by providing 'neutral' description) could be seen as privileging the mundane activities taking place in the workplace, and standing in the way of other organisational agendas, and (2) that using ethnography might be akin to opening a 'can of worms' if it serves to cast doubt on the wisdom of courses of action that have already been committed to.[11] We can see how some of the variants of approaches discussed in this chapter more or less explicitly oriented to a requirement to make ethnography tame enough to be part of a normally managed project by giving it a focus, limiting its extent or finding ways to

---

[10] Taking the mammography decision support example again (Hartswood, 1999), as stated above, our study did identify 'simplifying assumptions and open up new avenues for design' but since we were evaluating a software package nearing maturity, there was little value to be gained from simply saying 'if you were going to start again, I'd do it differently', but much from assisting efforts deploying the existing system.

[11] Suchman (1995) draws out the equivocal nature of producing ethnographic representations to inform design.

generalise or 'package' its findings. We might see contextual design as at the extreme end of this continuum, orienting as it does to mechanising the conduct of ethnography, and reversing the privilege given to the 'shop floor' by orienting strongly towards delivering business processes, and viewing members practices as being in need of repair (Beyer and Holtzblatt 1998). The need to move from 'description to prescription' in as orderly way as possible can be seen as being driven in part by the imperatives of having ethnography as a managed component of a software project work, and, while spawning different ways of applying ethnography is not in and of itself a bad thing, the notion that the key to its success lies in discovering a magic formula for its application rather than building upon experience of its application in practice is, we believe, flawed.

3.8 Conclusions: Innovation in Use, Envisaging the Future

Designing working systems involves the crucial step of moving from work as observed to a vision of how work might be done in the future, using new technological artefacts and organisational arrangements. Traditional approaches to IT systems design assume that systems are designed by IT professionals, that at the end of the design process their implementation is finished and their properties fixed, at which point they are 'handed over' to users. This reliance on 'prior design' has turned out to be problematic, rarely providing solutions to the specific problems encountered in a setting and needing to be appropriated, fitted, extended, configured, grafted onto existing practices, etc. This observation is at odds with traditional supply-driven concepts of innovation, which saw finished artefacts emerging 'from the research and development laboratory as "black-boxed" technical solutions, already corresponding to user needs that could simply be diffused through the market to potential users' (Williams et al. 2005, p. 12). This presents a simplistic view of technological innovation that ignores the contributions of a wide variety of players, (end-users and various intermediaries) as well as the possible conflicts and significant uncertainties involved. The 'design fallacy' lies in 'the presumption that the primary solution to meeting user needs is to build ever more extensive knowledge about the specific context and purposes of various users into technology design' (ibid. p. 67).

The social learning perspective (Sørensen 1996; Williams et al. 2005) provides an alternative analysis that includes activities during implementation and use, and highlights the diversity of players, their active role in technological development, as well as their interactions and negotiations around design and use of technologies. Fleck (1988, 1993, 1999) coined the term 'innofusion' (innovation in technology diffusion) to describe the 'processes of technological design, trial and exploration, in which user needs and requirements are discovered and incorporated in the course of the struggle to get the technology to work in useful ways, at the point of application' (Fleck 1988, p. 3). In a similar way, the related concept of domestication draws attention to the various ways in which technologies are made sense of and accommodated or made 'at home' within larger socio-material arrangements (Williams et al. 2005, pp. 56–58). This involves, *inter alia*, acquiring necessary skills, exploring possible uses, and developing practices and routines.

Attempts to improve the requirements-gathering process and thereby the ‘fit’ of technologies with working practices during design will inevitably be frustrated by the changing circumstances of use – brought about, in part, by the introduction of the system itself. Systems are often used in different ways than originally intended by designers and the adaptations that people make range from the seemingly trivial to organisational process innovations of significant scope (e.g. Bowers et al., 1995). Such innovations may be traded locally, within organisations or may, in the extreme case, be traded in the marketplace or be fed back into design through mechanisms such as user groups or other forms of user–supplier relationships.

Underpinning this view is a departure from the traditional understanding of requirements as somehow pre-existing, as something that can be ‘captured’ through appropriate ‘requirements-gathering’ methods (Jirotka and Goguen, 1994; especially Woolgar, 1994). Rather, requirements are seen as being constantly evolving and in need of being ‘worked-up’ and regularly revised in the light of the situation at hand. Any step in the design process and any event might potentially lead to changes, to new or changed requirements being formulated. Requirements as an outcome of social activities are also not unambiguous but inherently complex, thus reflecting the different interests various people have. Working up a set of requirements that can be used to inform the development of a system therefore involves negotiations and bringing into alignment various parties, technological, and organisational arrangements. This is inevitably an ongoing concern rather than something that can be done once and for all.

It is therefore important for researchers to investigate the appropriation and use of IT systems and to attend to the potential for innovation during this phase. Attending to the local contingencies of technology appropriation and use offers a way to find candidate solutions to overcome the problems of local fit of generic offerings discussed above. However, this can only be achieved through a long-term commitment to developing and supporting local configurations of technological arrangements, through a partnership between IT specialists, end-users, and other organisational stakeholders (cf. Hartswood et al., 2000; Voss et al., 2000; Hartswood et al., 2007). Such a partnership can make the work of envisaging and realising new technological options more achievable through stepwise design and experimentation. An approach that we have called ‘co-realisation’ takes up this challenge by envisaging design as a longitudinal process that fosters accountability and capitalises on an ethnographic engagement with work practice (Hartswood et al., 2007).

In this chapter we have charted the emerging recognition of the importance of understanding the sociality of work and the concomitant emergence of ethnography as a means of its explication. We have also given a brief history of software development methods, and highlighted their sociality and how their application is also underpinned by collaborative practices. We have argued that questions arising as to precisely how and what ethnography can potentially contribute to design depends not so much on the limitations of ethnography *per se*, but on the circumstances of and the skill in its application, the timeliness of its findings in relation to the current phase of development, the willingness of designers to engage with the ethnographic process, and conversely, the willingness of ethnographers to appreciate designers’ constraints and priorities. We also noted that the specialist ethnographer brings yet another division of labour to the design team as well as drawing attention to the fact that ethnography attempts to span the already existing gap between the design and use context, rather than to close it off. One might speculate that the role ethnography might play in informing design will depend upon the future shape of design and design practices. In this final section we have explored how shifting the focus of design from *prior design* to *design in use* can further remove design activities from being undertaken in distinct phases. This perhaps provides for a continuous application of ethnographic sensibilities (with a greater or lesser emphasis at different times), reducing the problems that arise coordinating findings from a distinct ethnographic phase with the demands of design.

References

Anderson, R.J. (1994). Representations and requirements: The value of ethnography in system design. *Human-Computer Interaction*, 9, 151–182.

Anderson, R.J. (1997). Work, ethnography, and system design. In: Kent, A. and Williams, J.G. (eds.), *The Encyclopaedia of Microcomputing*. New York: Marcel Dekker, pp. 159–183.

Anderson, R.J., Hughes, J.A. and Sharrock, W.W. (1989). *Working for Profit: The Social Organisation of Calculation in an Entrepreneurial Firm*. Avebury: Aldershot.

Bannon, L. (1991). From human factors to human actors: The role of psychology and human computer interaction studies in system design. In: Greenbaum, J. and Kyng, M. (eds.), *Design at Work: Cooperative Design of Computer Systems*. Hillsdale, NJ: Lawrence Erlbaum.

Bannon, L. and Schmidt, K. (1991). CSCW: Four characters in search of a context. In: Bowers, J. and Benford, S. (eds.), *Studies in Computer Supported Cooperative Work: Theory, Practice and Design.* Amsterdam: North-Holland, pp. 3–16.

Bansler, J.P. and Havn, E. (1991). The nature of software work: Systems development as a labor process. Van den Besselaar, P., Clement, A. and Järvinen, P. (eds.), *Information System, Work and Organization Design*. Elsevier Science Publishers B. V., pp. 145–153.

Beck, K. (1998). Extreme programming: A humanistic discipline of software development. Astesiano, E. (ed.), *Proceedings of the 1st International Conference on Fundamental Approaches to Software Engineering*, Lecture Notes in Computer Science 1382, Springer, pp. 1–6.

Beck, K. (2000). *Extreme Programming Explained: Embracing Change*. Addison Wesley.

Bentley, R., Hughes, J.A., Randall, D., Rodden, T., Sawyer, P., Shapiro, D. and Sommerville, I. (1992). Ethnographically-informed systems design for air traffic control. *Proceedings of the 1992 ACM conference on Computer-supported cooperative work,* November 01–04, Toronto, Canada, pp. 123–129.

Berg, M. and Timmermans, S. (2000). Order and their others: On the constitution of universalities in medical work. *Configuration*s, 8(1), 31–61.

Beyer, H. and Holtzblatt, K. (1998). *Contextual Design: Defining Customer-Centered Systems*. Morgan Kaufmann Publishers Inc.

Bjerknes, G. and Kautz, K. (1991). Overview – A Key Concept in Computer Supported Cooperative Work. *Computergestützte Gruppenarbeit* (CSCW): 1. Fachtagung, 20. September bis 2.Oktober, Bremen. B.G. Teubner, pp. 153–169.

Blomberg, J., Giacomi, J., Mosher, A. and Swenton-Wall, P. (1993). Ethnographic field methods and their relation to design. In: Schuler, D. and Namioka, A. (eds.), *Participatory Design: Perspectives on Systems Design.* Hillsdale, NJ: Lawrence Erlbaum Associates, pp. 123–155.

Blomberg, J., Suchman, L. and Trigg, R. (1996). Reflections on a work-oriented design project. *Human-Computer Interaction*, 11(3), 237–265.

Boehm, B.W. (1988). A spiral model of software development and enhancement. *IEEE Computer*, 21, 61–72, May 1988.

Boehm, B. and Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *IEEE Software*, September/October 2005, pp. 30–39.

Bowers, J., Button, G. and Sharrock, W. (1995). Workflow from within and without. Marmolin, H., Sundblad, Y. and Schmidt, K. (eds.), *Proceedings of the Fourth European Conference on Computer-Supported Cooperative Work*, 10–14, September 1995, Stockholm, Sweden, pp. 51–66.

Bowers, J. and Pycock, J. (1994). Talking through design: Requirements and resistance in cooperative prototyping. *Human Factors in Computing Systems (CHI'94)*, Boston, Massachusetts, pp. 299–305.

Brooks, F.P., Jr. (1975). *The Mythical Man-Month: Essays on Software Engineering.* Addison-Wesley.

Büscher, M., Christensen, M., Grønbæk, K., Krogh, P., Mogensen, P., Shapiro, D. and Ørbæk, P. (2000). Collaborative augmented reality environments: Integrating VR, working materials, and distributed work spaces. *Proceedings of the 3rd International Conference on Collaborative Virtual Environment*, ACM Press, San Francisco, CA, USA, pp. 47–56.

Büscher, M., Gill, S., Mogensen, P. and Shapiro, D. (2001). Landscapes of practice: Bricolage as a method for situated design. *Computer Supported Cooperative Work*,10(1), 1–28.

Büscher, M., Eriksen, M.A., Kristensen, J.F. and Mogensen, P.H. (2004). Ways of grounding imagination*.* *Proceedings of the Participatory Design Conference*, Toronto, Canada, pp. 193–203.

Button, G. (1993). An organisational account of the question "Do users get what they want?".*ACM SIGOIS Bulletin*, 14(2), 35–40.

Button, G. and Sharrock, W. (1994). Occasioned practices in the work of software engineers. In: Jirotka, M. and Goguen, J.A. (eds.), *Requirements Engineering: Social and Technical Issues.* London: Academic Press, pp. 217–240.

Button, G. and Sharrock, W. (1995a). Practices in the work of ordering software development. In: Firth, A. (ed.), *The Discourse of Negotiation: Studies of Language in the Workplace*. Oxford: Pergamon, pp. 159–180.

Button, G. and Sharrock, W. (1995b). The mundane work of writing and reading computer programs. In: ten Have, P. and Psathas, G. (eds.), *Situated Order: Studies in the Social Organization of Talk and Embodied Activities*. University Press of America, pp. 231–258.

Button, G. and Sharrock, W. (1996). Project work: The organisation of collaborative design and development in software engineering. *Computer Supported Cooperative Work*, 5(4), 369–386.

Button, G. and Sharrock, W. (1998). The organizational accountability of technological work. *Social Studies of Science*, 28(1), 73–102.

Button, G. and Sharrock, W. (2000). Design by problem-solving. In: Luff, P., Hindmarsh, J., and Heath, C. (eds.), *Workplace Studies: Recovering Work Practice and Informing System Design*. Cambridge University Press, pp. 46–67.

Charette, R.N. (2005). Why software fails. *IEEE Spectrum*, 42(9), 42–49, Sept. 2005.

Crabtree, A. (2003). *Designing Collaborative Systems: A Practical Guide to Ethnography*. Springer.

Curtis, B., Krasner, H. and Iscoe, N. (1988). A field study of the software design process for large systems. *Communications of the ACM*, 31(11), 1268–1287.

Erickson, T. (2000). Supporting interdisciplinary design: Towards pattern languages for workplaces. In: Luff, P., Hindmarsh, J. and Heath, C. (eds.), *Workplace Studies: Recovering Work Practice and Informing System Design*. Cambridge University Press, pp. 252–261.
Fleck, J. (1988). Innofusion or diffusation: The nature of technological development in robotics. *Edinburgh PICT Working Paper No. 4*, Edinburgh University, Edinburgh.
Fleck, J. (1993). Innofusion: Feedback in the innovation process. In: Stowell, F. A. et al. (eds.), *Systems Science*, Plenum Press, pp.169–174.
Fleck, J. (1999). Learning by trying: The implementation of configurational technology. In: MacKenzie, D. and Wajcman, J. (eds.), *The Social Shaping of Technology*, (2nd ed.). Open University Press, pp. 244–265.
Garfinkel, H. (1967). *Studies in Ethnomethodology*. Prentice-Hall.
Geertz, C. (1973) Thick Description: Toward an Interpretive Theory of Culture. In his *The Interpretation of Cultures*. New York: Basic Books.
Glass, R.L. (2006). *Communications of the ACM*, 49(8), 15–16.
Greenbaum, J. and Kyng, M. (1991). Introduction: Situated design. In: Greenbaum, J. and Kyng, M. (eds.), *Design at Work: Cooperative Design of Computer Systems*. Lawrence Erlbaum Associates 1–24.
Grinter, R.E. (2003). Recomposition: Coordinating a web of software dependencies. *Computer Supported Cooperative Work*, 12, 297–327.
Harper, R.H.R. (2000). The organisation in ethnography: A discussion of ethnographic fieldwork programs in CSCW. *Computer Supported Cooperative Work*, 9, 239–264.
Hartswood, M. (1999). *Human-factors in computer-aided mammography*. PhD Thesis, University of Edinburgh, Edinburgh.
Hartswood, M., Jirotka, M., Procter, R., Slack, R., Voss, A. and Lloyd, S. (2005) Working IT out in e-Science: Experiences of requirements capture in a HealthGrid project. In *Proceedings of HealthGrid Conference*, Oxford.
Hartswood, M., Procter, R., Rouncefield, M. and Sharpe, M. (2000). Being there and doing IT in the workplace: A case study of a co-development approach in healthcare. In: Cherkasky, T., Greenbaum, J. and Mambery, P. (eds.), *Proceedings of the CPSR/IFIP WG 9.1 Participatory Design Conference*, Nov. 28th-Dec. 1st, New York, pp. 96–105.
Hartswood, M., Procter, R., Rouncefield, M., Slack, R., Voss, A., Büscher, M. and Rouchy, P. (2007). Co-realisation: Evolving IT artefacts by design. In: Ackerman, M., Halverson, C., Erickson, T. and Kellogg, W. (eds.), *Resources, Co-Evolution and Artefacts*. Springer, pp. 59–94.
Hemmings, T., Crabtree, A., Rodden, T., Clarke, K. and Rouncefield, M. (2002). Probing the probes. In: Binder, T., Gregory, J. and Wagner, I. (eds.), *Proceedings of the Participatory Design Conference*, Malmö, Sweden, pp. 42–50.
Heritage, J. (1984). *Garfinkel and Ethnomethodology*. Polity Press.
Hughes, J.A., King, V., Rodden, T. and Andersen, H. (1994). Moving out of the control room: Ethnography in system design. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, Chapel Hill, NC: ACM Press, pp. 429–438.
Hughes, J., King, V., Rodden, T. and Andersen, H. (1995). *The Role of Ethnography in Interactive Systems Design*. Interactions, April, 56–65.
Hughes, J., O'Brien, J., Rodden, T. and Rouncefield, M. (2000). Ethnography, communication and support for design. In: Luff, P., Hindmarsh, J. and Heath, C. (eds.), *Workplace Studies: Recovering Work Practice and Informing System Design*. Cambridge University Press, pp. 187–214.
Hughes, J.A., Randall, D. and Shapiro, D. (1992). Faltering from ethnography to design. *Proceedings of the 1992 ACM Conference on Computer Supported Cooperative Work*, Toronto, Ontario, Canada, pp. 115–122.

Hughes, J.A., Randall, D. and Shapiro, D. (1993). From ethnographic record to system design: Some experiences from the field. *Computer Supported Cooperative Work*, 1(3), 123–141.

Jirotka, M., Gilbert, N. and Luff, P. (1992). On the social organisation of organisations. *Computer Supported Cooperative Work*, 1, 95–118.

Jirotka, M. and Goguen, J.A. (eds.). (1994). *Requirements Engineering: Social and Technical Issues*. Academic Press.

Kling, R. (1991). Cooperation, coordination and control in computer-supported work. *Communications of the ACM*, 34(12), 83–88, Dec. 1991.

Lehman, M.M. (1998). Software's future: Managing evolution. *IEEE Software*, 15(1), 40–44, Jan.–Feb. 1998..Martin, D., Rodden, T., Rouncefield, M., Sommerville, I. and Viller, S. (2001). Finding patterns in the fieldwork. *Proceedings of the European Conference on Computer Supported Cooperative Work*, Kluwer, pp. 39–58.

Martin, D., Rodden, T., Rouncefield, M., Sommerville, I. and Viller, S. (2001). Finding Patterns in the Fieldwork. *Proceedings of the European Conference on Computer Supported Cooperative Work*. Kluwer. Pp. 39–58.

Martin, D. and Sommerville, I. (2004). Patterns of cooperative interaction: Linking ethnomethodology and design. ACM *Transactions on Computer-Human Interaction*, 11(1), 59–89.

Nandhakumar, J. and Avison, D.E. (1999). The fiction of methodological development: A field study of information systems development. *Information Technology & People*, 12(2), 176–191.

Nardi, B.A. and Miller, J.R. (1990). An ethnographic study of distributed problem solving in spreadsheet development. *Proceedings of the ACM Conference on Computer-supported Cooperative Work*, Oct. 7–10, Los Angeles, pp. 197–208.

Naur, P. and Randell, B. (eds.). (1969). Software engineering. *Report on a Conference Sponsored by the NATO Science Committee*, Garmisch, Germany, 7–11 Oct. 1968. Scientific Affairs Division, NATO, Brussels.

Ortner, S. (1997). Introduction. Representations, no. 59, Special Issue: *The Fate of "Culture"*: Greetz and Beyond. pp. 1–13.

Plowman, L., Rogers, Y. and Ramage, M. (1995). What are workplace studies for? In *Proceedings of ECSCW'95*, Stockholm, September.

Procter, R. and Williams, R. (1996a). Beyond design: Social learning and computer-supported cooperative work. In: Shapiro, D., Taubner, M. and Traunmüller, R. (eds.), *The Design of Computer-Supported Cooperative Work and Groupware Systems*, Elsevier Science, pp. 445–464.

Procter, R. and Williams, R. (1996b). Social learning and innovations in multimedia-based *CSCW*. *ACM SIGOIS Bulletin*, 17(3), 73–76.

Randall, D., Hughes, J. and Shapiro, D. (1994). Steps toward a partnership: Ethnography and system design. In: Jirotka, M. and Goguen, J.A. (eds.), *Requirements Engineering: Social and Technical Issues*. Academic Press, pp. 241–258.

Royce, W.W. (1987 [1970]). Managing the development of large software systems: Concepts and techniques. *Proceedings of the 9th International Conference on Software Engineering*, Monterey, CA, pp. 328–338. (originally published in *Proc. IEEE WESCON* 1970, pp. 1–9).

Rönkkö, K., Dittrich, Y. and Randall, D. (2005). When plans do not work out: How plans are used in software development projects. *Computer Supported Cooperative Work*, 14, 433–468.

Rouncefield, M.F. (2002). *'Business as Usual': An Ethnography of Everyday (Bank) Work.* PhD Thesis, Department of Sociology, University of Lancaster.

Russo, N.L. and Stolterman, E. (2000). Exploring the assumptions underlying information systems methodologies: Their impact on past, present and future ISM research. *Information Technology & People*, 13(4), 313–327.
Sauer, C., Gemino A. and Reich, B.H. (2007). The impact of size and volatility on IT project performance. *Communications of the ACM*, 50(11), 79–84.
Schmidt, K. (2000). The critical role of workplace studies in CSCW. In: Luff, P., Hindmarsh, J. and Heath, C. (eds.), *Workplace Studies: Recovering Work Practice and Informing System Design.* Cambridge University Press, pp. 141–149.
Schmidt, K. and Bannon, L. (1992). Taking CSCW seriously supporting articulation work. *Computer Supported Cooperative Work*, 1(1–2), 7–40.
Sharma, S. and Rai, A. (2000). CASE deployment in IS organizations. *Communications of the ACM*, 43(1), 80–88.
Sharrock, W. and Anderson, R. (1993). Working towards agreement. Button, G. (ed.), *Technology in Working Order: Studies of work, interaction and technology*. Routledge, pp. 149–161.
Sharrock, W. and Anderson, R. (1994). The user as a scenic feature of the design space. *Design Studies*, 15(1), 5–18. Also Rank Xerox EuroPARC Technical Report EPC-91-105, 1991.
Sørensen, K.H. (1996). Learning technology, constructing culture, socio-technical change as social learning. *STS Working Paper, No. 18/96*, University of Trondheim, Centre for Technology and Society.
Sommerville, I. (2001) *Software Engineering* (6th ed.). Addison Wesley.
Standish Group International. (1994). *The Chaos Report*. http://www.standishgroup.com/sample_ research/chaos_1994_1.php
Stewart, J. and Williams, R. (2005). The wrong trousers? Beyond the design fallacy: Social learning and the user. In: Howcroft, D. and Trauth, E.M. (eds.), *Handbook of Critical Information Systems Research: Theory and Application*. Edward Elgar, pp. 195–221.
Suchman, L.A. (1987). *Plans and Situated Actions: The Problem of Human-Machine Communication.* Cambridge University Press.
Suchman, L. (1995). Making work visible. *Special issue of CACM*, 38(9), 56–64.
Trigg, R.H., Blomberg, J. and Suchman, L. (1999). Moving document collections online: The evolution of a shared repository. In: Bødker, S., Kyng, M. and Schmidt, K. (eds.), *Proceedings of the 6th European Conference on Computer-Supported Cooperative Work.* Kluwer Academic Publishers, pp. 331–350.
Turk, D., France, R. and Rumpe, B. (2002). Limitations of agile software processes. *Proceedings of the Third International Conference on Extreme Programming and Flexible Processes in Software Engineering*, XP2002, May 26–30, Alghero, Italy, pp. 43–46.
Voss, A., Procter, R. and Williams, R. (2000). Innovation in use: Interleaving day-to-day operation and systems development. In: Cherkasky, T., Greenbaum, J. and Mambery, P. (eds.), *Proceedings of the CPSR/IFIP WG 9.1 Participatory Design Conference*, New York, Nov. 28th–Dec. 1st, pp. 192–201.
Warr A., de la Flor, G., Jirotka, M. and Lloyd, S. (2007). Usability in e-Science: The eDiaMoND case study. In *CHI International Workshop on Increasing the Impact of Usability Work in Software Development*, San Jose, CA, Apr. 28th–May 3rd.
Wastell, D.G. (1996). The fetish of technique: Methodology as a social defence. *Information Systems Journal*, 6, 25–40.
Williams, R., Stewart, J. and Slack, R. (2005). *Social Learning in Technological Innovation: Experimenting with Information and Communication Technologies*. Edward Elgar.
Woolgar, S. (1991). Configuring the user: The case of usability trials. Law, J. (ed.), *A Sociology of Monsters: Essays on Power, Technology and Domination.* Routledge, pp. 58–100.

Woolgar, S. (1994). Rethinking requirements analysis: Some implications of recent research into producer – consumer relationships in IT development. In: Jirotka, M. and Goguen, J. (eds.), *Requirements Engineering: Social and Technical Issues*. Academic Press, pp. 201–216.

Yourdon, E. and Constantine, L.C. (1979). *Structured Design: Fundamentals of a Discipline of Computer Program and System Design.* Prentice Hall.

Chapter 4

User–Designer Relations in Technology Production: The Development and Evaluation of an 'Animator' Tool to Facilitate User Involvement in the Development of Electronic Health Records

K. Neil Jenkings

Institute of Health and Society at Newcastle University, neil.jenkings@newcastle.ac.uk

4.1 Introduction

In this chapter we report on a prototype product, an 'Animator', and its use as part of a novel approach for facilitating user–designer relations in development of an electronic health record (EHR) for the UK National Health Service (NHS). We describe the development and initial evaluation of the Animator, a story-based communications tool designed to elicit users' expertise and knowledge, and its role in involving users in developing a vision for EHRs. Our case study provides a focus for exploring problems of user–designer relations in large, heterogeneous organisations where traditional participatory design (PD) approaches may struggle with the complexity and scale of the undertakings. Nevertheless, we show how our intervention can, in principle, generate insights useful for design, but how they still face problems of being heard in a heavily centralised procurement and implementation programme. We first set the scene by describing England's National Programme for Information Technology (NP*f*IT), an attempt to create the most comprehensive EHRs infrastructure of any health care system (Cross, 2006a). We examine the problems NP*f*IT has encountered in adequately engaging end-users of the systems it aims to deliver. We then discuss more generally the challenges posed to participatory approaches by innovations of this scale and complexity before describing our own experiences employing the Animator as a catalyst for user engagement.

4.1.1 The National Programme for IT

England's NP*f*IT, sponsored by the Department of Health (DH), is the biggest outsourced public sector IT project ever undertaken (Dyke, 2003). It emerged as a result of the Wanless Report (2002), which called for the NHS to shift to a national approach to ICT procurement (NHS 2002). NP*f*IT aims to deliver the following systems and services (as well as a variety of other management systems for general practice):

A. Voss et al. (eds.), *Configuring User-Designer Relations*,
DOI: 10.1007/978-1-84628-925-5_4,

The NHS Care Records Service, an individual care record for every patient in England accessible by patients and their carers as well as health care professional;

Choose and Book, an electronic booking service for patients to hospitals and clinics;

Electronic Transmission of Prescriptions (ETP) to replace paper-based services;

Contact, a central email service to allow the transfer of patient information and again replacing paper-based services; and

Picture Archiving and Communications Systems (PACS) for the static and moving digital image capture, storage, display, and distribution.

This chapter focuses mainly on the NHS Care Records Service and, in particular, the problem of user–designer relations in the deployment of many interlocking systems on a massive scale and to a heterogeneous group of end-users. It is evident that NP*f*IT represents a huge programme of work that intends radical transformation of the information and communication infrastructure of the NHS. Unsurprisingly, it raises a number of challenges, not least the engagement of end-users in its ongoing implementation. To make the picture more complex, NP*f*IT does not simply aim to replace existing functionality with more modern equivalents but aims to transform various aspects of care, and thus introduce new working practices and working relations into the NHS organisation(s):

> The care records service will create EHRs by combining central data about patients, including their identifying characteristics (the personal demographics service), administrative records, and important health alerts such as allergies, with summaries of care episodes drawn from local institution-based electronic patient records and, eventually, social care case files.(Cross, 2006b, pp. 656)

The NHS is not, however, a single hierarchical organisation, but rather is comprised of multiple, overlapping organisational power structures and decision centres, where any shift in the patterns of responsibilities, access rights, and ownership of data brought about by NP*f*IT will impact on existing power relations and practices. Some caution is required as using ICT to realise organisational change is not generally recommended, even if the introduction of ICT involves more than the replacement of existing procedures, documents, and tasks (Hendy et al. 2005). This point is worth emphasising as it highlights that we should anticipate major socio-cultural impacts as a result of NP*f*IT that need to be properly understood and managed from the outset (with the focus in this chapter on their being properly 'understood').

Perhaps for some of the reasons outlined above, there is a feeling among many observers that NP*f*IT is a technologically determinist programme foisted on the NHS by 'outsiders' with little knowledge of health care (Cross 2006a) and that sight has been lost of the impact it will have in the way people work and the ways in which services are delivered (Humber 2004). Humber sums up this mood of scepticism, highlighting the importance of user engagement alongside technical and managerial problems:

> yet if the national programme can address efficiently its political, logistical, technical and contractual issues and win over and sufficiently engage health-

care professionals, patients, and the public while taking their concerns and views seriously then it might just succeed… (Humber 2004, pp. 1145–1146 Emphasis added)

To date, however, the involvement of user groups has been seen to be poor, with the British Medical Association demanding greater involvement following the perceived failings of forums set up to represent and give voice to health care professionals and the public or patients (the National Clinical Advisory Board and the Public Advisory Board, respectively; Humber 2004). Hendry et al. also point out that 'the socio-cultural challenges to implementing the NP*f*IT are as daunting as the technical and logistical ones' (Hendy et al. 2005, p. 331), and goes on to report that senior NHS staff felt that socio-cultural factors had been neglected; that better communication was required (and not just in one direction, i.e., from NP*f*IT downwards); that genuine consultation was required and that only through this could a sense of ownership be fostered. The issues go beyond simply engendering a sense of ownership, of course, the systems must work and the implementations must be managed without alienating staff and disrupting the local provision of health care. However, concerns over the programme as a whole and Care Records Service, in particular, fill the professional journals. The following is a typical example:

"At this late stage we still do not know how much of the medical record is going to be exported to the records service. We do not know what control patients will have over what information about them is held there. We do not know if the amount of context required for a remote record to be meaningful exceeds or is less than what patients will consent to. We do not know how the passage of time will affect patients' and doctors' interpretation of events. In short, it is not clear how a centralised record system will sit with the dispersed relationships that constitute primary care or whether the government will get any useful return on its investment" (Robinson 2005, p. 315).

4.1.2 Why were Problems of User Engagement Not Anticipated?

There is a widespread feeling that the NP*f*IT has failed so far to engage users in a meaningful fashion and the question inevitably arises as to why problems associated with user engagement were not foreseen? The lack of user engagement is even more surprising given that pilot studies conducted prior to the Care Record Service implementation had flagged the need to better understand existing situated clinical practices, and human and technological resources. It was also recognised that these needed to be understood with an eye to how they would inform ICT design, development, and implementation. The precursor studies in question were a number of investigative research projects commissioned by the NHS under an umbrella programme of work referred to as the Electronic Records Development and Implementation Programme (ERDIP). This chapter concerns a tool developed under the auspices of ERDIP, which investigated the possibility of inculcating user–designer relations in the face of a mass workforce. We argue that use of the resulting Animator tool may have helped forestall some of the problems highlighted above had it been developed and deployed as part of the design and development of the Care Records Service.

Given that the problems of user engagement had been highlighted, and approaches to engagement trialled, an explanation for the subsequent lack of progress addressing them is never likely to be straightforward. One answer might be that they were never given the priority they deserved – that those pushing NP*f*IT through did not see the merit of understanding clinical work practice, nor the political necessities of keeping powerful stakeholders on board. One might see this as a particularly egregious error given the cautionary takes of adopting such an approach to socio-cultural factors in ICT projects (e.g. Southon et al. 1999). But, given the vast scale, scope, and complexity of the project, and the Titanic pressures, many of them political, to push ahead at ‘full speed’, it is perhaps not so surprising that issues of user engagement were relegated in this way. Public sector provision of ICT, instead of being driven by a detailed understanding of end-user requirements, is more often driven by technological, economic, and (especially) in the case of the UK’s NHS, political imperatives. Whatever their source though, such drivers frequently fail to engage adequately with the actual needs of end-users, and even less frequently to actually engage directly with users to obtain their expert knowledge and worldview in the crucial early stages of the design–development–implementation process. This is despite an increasingly widespread recognition that a central reason for the difficulties encountered in the design and introduction of ICT is the inadequacy of primary research into the everyday practices and concerns of health care professionals in the work places where the technologies are to be installed (Hartswood et al. 2003; Hendy et al. 2005; Jenkings 2007). Such involvement not only contributes to the systems themselves, but also to their acceptance (van der Meijden et al. 2000a, 2000b; Helleso and Ruland 2001) and, without user involvement, systems will at best reflect how medicine is taught rather than how it is practiced in the workplace (Sicotte et al. 1998). Furthermore, new systems typically greatly impact upon the social and professional organization of medical care at the workplace (ibid.). It is unsurprising, therefore, that health care professionals, and the bodies that represent them, seek political engagement and influence with programmes such as NP*f*IT.

4.1.3 Participatory Design and Large-Scale Implementations

For successful deployment of Electronic Patient Records (EPR) – the record of the periodic care given by one institution – it is important for systems to be tailored to daily routines and local practices (van der Meijden et al. 2000a). Although an EHR – the longitudinal record of primary care with subsets of information from various EPRs – may be viewed as being less local and more ‘national’, the same local user involvement is needed since the facilitation of (or at least impact upon) local activities is inevitable. So, while the number of potential users is much greater, many of the same problems as EPRs have encountered will still apply to EHRs (Bates 2002). This characteristic of EHRs that they are nationally implemented systems with local impact raise methodological questions about user engagement in their design. PD approaches, because of their ‘local’ character, are perhaps seen as not dealing so well with wider organisational constraints, including various regulatory frameworks, the need for interoperability, economies of scale, political horse-trading of various sorts, and so on. Critiques of this sort may be used as excuses not to engage in what are seen as site-specific studies, perhaps through concerns that findings would not generalise, or that too strong commitments might be made to local priorities. This is, however, to underestimate the utility of such studies as resources, or probes, for design – even if the findings have to be tempered in light of various practical and organisational exigencies. The aim can be seen as not trying to achieve a 'tailored' fit to each set of local circumstances, but a better informed, 'off the peg' model.

There is, however, an important distinction to be made between what might constitute potentially useful user engagement on the one hand and PD in its purest sense on the other. When designing a system for a small number of users it is possible, or at least more feasible, for the majority to be involved in a hands-on fashion in the design process. With an organisation the size of the NHS and an ICT project that will intentionally impact upon most staff meaningful PD by all members is impossible. Of course, this is not unique to the NHS but just more obviously the case. Rather than using this as an excuse to avoid wide-scale user involvement, it is suggested here that there *is* a possibility of undertaking useful user-engagement work in the very early stages of the technology design and procurement commissioning process.[1] If issues of user engagement are not addressed in large organisations, they too court failure, and size will not protect them.

It is in this sense of a tool enabling pan-organisation user engagement, rather than approaches involving hands-on user–designer workshops, which differentiates our approach from traditional PD approaches. While we did hold to the idea of 'grounding imagination' as promoted by Buscher et al. (2004), we did not adopt triangulation of more than one PD method. Although, as noted above, we suggest that different methods are appropriate for different stages of the design process.

Nevertheless, large-scale health organisations have problems engaging end-users due to the number and diversity of both their staff and clientele. Facilitating user–designer relations is a difficult problem, and one that becomes increasingly complex and multifaceted in line with the size of the organisation, the heterogeneity of its workforce and of work-practices. We would also argue that larger organisations with heterogeneous user groups, require different modes of user engagement that attend both to the variety of the practices to be supported and the unfolding demands of the implementation trajectory as it moves between procurement, development, configuration, testing, and deployment. User engagement may be more straightforward when a technological development is directed at, or deployed into, a single location or specific type of suborganization – although the 'knock-on effect' of technologies in

[1] This is not to dismiss the need for more focused PD methods at later stages, indeed, the initial user engagement maybe seen as feeding into later PD processes. User engagement should be throughout the life of a project, not just certain stages. If the user engagement is missing at the beginning of a project it may not be possible to repair the situation in light of later user engagement. Conversely, user engagement needs to be maintained towards the end of a project, rather than be replaced by proxies and user representatives where the benefits of initial user engagement can be 'lost'. There is a need for a more even approach to user engagement, and different types of such engagement, including PD-type approaches throughout the life of a project (Martin et al. 2005, 2008).

such areas is not to be underestimated. However, for an IT implementation programme on an unprecedented scale that will impact across virtually the whole of a complex organisation, as with the CRS, then problems of user engagement become much more fraught, harder to manage, and more difficult to profit from. Especially, when the intended technology will radically change the intraorganisational collection, sharing and use of patient information, and impact significantly on working practices, and inevitably be a source of disruption both during and following implementation. There are not only issues of user engagement to inform design and implementation, but also in informing the users of (and helping them to shape) the likely impact and 'future vision' of their post-implementation workplace. Recognising this widens the scope of the forms of required user engagement. It might be argued that communicating an organisational vision to users is somewhat separate from their engagement in shaping the more concrete aspects of the system such as user interface, work flow and dataset design. We would disagree with this and, instead, argue for modes of engagement that bring together both organisational communication and user–designer engagement as part of a unified strategy, partly because we recognise that users understanding the aims and scope of the system is a pre-requisite for engaging meaningfully in more detailed design work. Finding a strategy for engaging intended hands-on, 'coal face' users into the larger future envisioning at an early stage, and facilitating such a process in the development of an EHR is the challenge which the prototype Animator reported here attempts to address.

We argue that engaging users early on in this way may help forestall the sorts of problems that typically arise in technological, economic, and politically driven design processes. We do not, however, wish to suggest that there is a naïve technological determinism, or even political or economic determinism, at work in EHR system design, but that these are nevertheless significant, and often competing, drivers that can have a negative impact on user acceptance and fitness for purpose (Braverman 1974; Winner 1980). We are not suggesting that such factors can be eliminated, but we do argue that they can be tempered if robust information can be gathered through early user engagement and given sufficient weight to survive as a matrix of competing concerns. Concerns that are played out in the sites of system design decision making.

4.2 The Origins of the 'Animator'

This chapter reports on one part of an EHR demonstrator project, which was part of the recent development programme for EHRs in England and the forerunner to the current implementation NP*f*IT, now under control of the NHS Connecting for Health programme (NHS 2006). The momentum for these initiatives was established by 'Information for Health' (NHS 1998), which was published by the National Health Service (NHS) as a framework for the development of information services for the NHS. This report led to a range of activities, one of which was the ERDIP commissioned by the NHS Information Authority (NHSIA) (NHS 2001). The aim of the programme was to promote in-service development and demonstration of best practice and progress towards shared EHRs. In 2000, 17 projects were initiated in England: four research groups led by Strategic Health Authorities (Cornwall, South Staffordshire, County Durham & Darlington and Tees) began the development of 'pan-community' demonstrators to investigate how electronic records could be used to share patient information across health and social service communities. Each of these teams adopted different approaches to the problem. Thirteen additional 'focal' demonstrator research groups worked on smaller projects within ERDIP. The stated aim of this research programme was informing the development of policy and the national implementation programme (which would become NP*f*IT) and help the wider NHS in its local implementation of electronic records.

The County Durham & Darlington Electronic Health Record Project (DuDEHR)[2] project was not concerned solely with exploring potential EHR technologies and configurations. It was an attempt to ensure that the selection of candidate solutions was grounded in an understanding of the health and social care processes as experienced by practitioners in both a focused (for example, their situated work activities) and broad sense (for example, their knowledge of how their activities impact on, and are impacted upon, by other, micro and meso activities and organization) and not only as set out in official policy and guidelines (see Goorman and Berg 2000).

4.3 Project Development

The EHR, unlike EPR, is not a uniformly agreed upon concept, however, a reasonably consensual definition would be that an EPR details episodes of care in a single organisation, while an EHR would provide a longitudinal record of care across a variety of care providers (perhaps by pulling together data from the local 'EPRs'). We adopted the following working definition for the EHR: 'an electronic information resource for use throughout the NHS to support patient care by making relevant and appropriate information about patients available to the clinicians and other professionals charged with their care'.

Ethnographic research was undertaken at a number of healthcare sites within the Health Authority as part of the validation of architectural models developed and refined throughout the DuDEHR project. They were also used to help provide a focus on the wider problems associated with developing and implementing a pan-community EHR. Fieldwork sites included primary care, secondary care and mental health/social services. The ethnography aimed to produce detailed descriptions of the electronic and paper communications practices within each site, and their connection

[2] The County Durham & Darlington Electronic Health Record Project (DuDEHR), led by the Health Authority in collaboration with the Sowerby Centre for Health Informatics at the University of Newcastle (SCHIN) and a health care software supplier Eclipsys (now part of I-Soft), was one of the four pan-community demonstrators. The team involved in the Demonstrator project included Sarah Bell, Nick Booth, Andrew Izon, K. Neil Jenkings, Judy Kohannajad, Jasmin Latiff, Mike Martin, Paul Morgan and the IT team at SCHIN. The Animator storyboard was developed by Mike Martin. Special thanks go to Andrew Thompson.

to other agencies both internally and externally involved in the organisation and delivery of health care (e.g. general practices, departments within hospitals and mental health/social services offices). Even at an early stage of the research, a number of important issues emerged. These included a recognition of the diversity of practices employed in the sites visited (even within similar health care organisations) and that the practitioners and staff within these organisations had little or no conception of what an EHR was or could be – despite their considerable knowledge of the NHS. Consequently, it was decided that one of the aims of the project would be to develop a 'demonstrator' that would address this lack of awareness, be a data-gathering tool to complement the ethnographies in informing the architectural models, and to help foster better user–designer relations within the ERDIP programme, particularly with respect to what an EHR might be and how it could be achieved. This tool was designated the 'Animator'.

4.4 Development of the Animator

The project took the position that engaging potential end-users in EHR project programmes was both fundamental to successful development and implementation, and demanded communication processes based not solely around the written word (Buscher et al. 2004). The Animator was developed to provide an audio-visual presentation of what a pan-community EHR could look like, how it could work in practice, and to suggest some of the changes that it could bring about. When designing the Animator, we oriented to a broad target audience, including those involved in health care delivery, management, advice giving, and administration.

The first version of the Animator attempted to show clinical and administrative activity identified by the ethnographies, including the interactions and movement of personnel and information through the use of a diagrammatic representation of a surgery and animated symbols and storyline (Fig. 4.1). The same story could be depicted from a systems perspective where linkages between different systems are shown, with embedded photographs displayed throughout the various scenarios (Fig. 4.2). This version proved, however, to be overly complex, long, and not very audience friendly. In addition, informal reviews by members of the research team prior to field testing suggested that the initial version was overly technologically orientated. It was felt that the initial version was both too prescriptive and concrete in its depiction of EHR to successfully engage participants. Overly detailed content and reliance on symbolic representation of actors seemed not to provide enough space for the sorts of interpretation and elaboration that we hoped to provoke. Because we were orienting towards a broad audience, achieving the right specificity was key: too general and no one will see their role in the activities, too specific and it will not relate to the work of the majority of the audience. Additionally, if at the 'envisioning stage' we were being too 'concrete' about specific practices we would risk discussions getting bogged down in details rather than discussion of general principles. This decision to abandon the initial Animator version was not based upon piloting with end-users; it was more the case that the initial version was a staging-post in the continual to-ing and fro-ing of ideas within the research team.

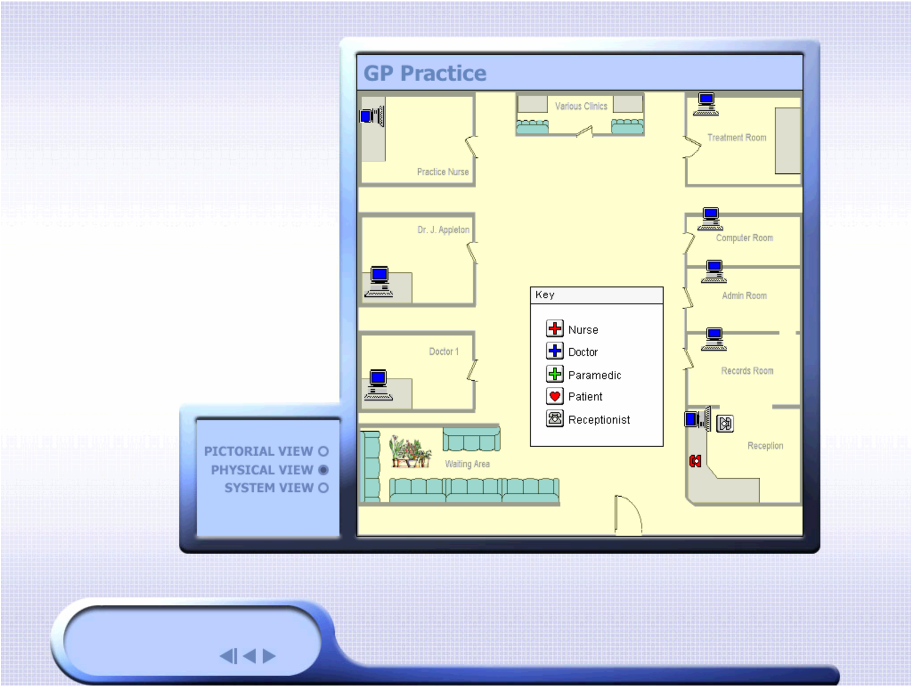

Fig. 4.1. Animator screenshot showing a 'physical view' of activity

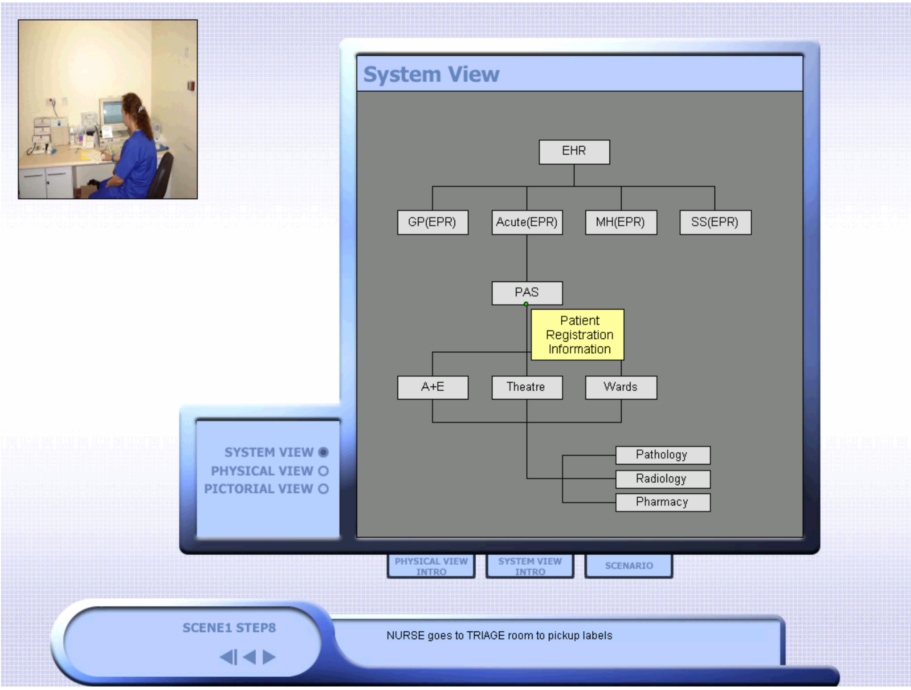

Fig. 4.2. Animator screenshot showing a 'system' view of activity

The second version of the Animator was based upon a story of a patient as he moved through an episode of health care and aimed to illustrate what health care *could* look like in an environment in which a working EHR operated (Fig. 4.3). Here, the ethnographic data and architectural models were not made explicit as in Version 1, but they were used to inform the depiction of the patient's experience. The Animator provides an audio-visual narrative of Mr. Jones's care (Fig. 4.4) through four linked scenarios. Each scenario was presented using a mixture of scripted dialogue between the actors, a representation of the activity of the various agents and health services and a technical animation of the messaging architecture, which showed the processing/exchange of information between the organisations involved. In the first scenario, Mr Jones phones NHS Direct – the 24-hour telephone helpline staffed by qualified nurses – from home complaining of chest pains. The story tells of how he is triaged by NHS Direct and how this is facilitated by Mr. Jones having an EHR that can be accessed by the health care call centre nurse. The Animator illustrates the type of information that would be potentially available through this record and how it is used not only to triage the patient, but also to transfer patient details to the ambulance crew, which the triage nurse has sent to the patient's home. The second scenario shows the ambulance crew with access to information from Mr Jones' medical record tailored to the requirements of the ambulance crew. A third scenario follows where the ambulance crew have contacted the hospital Accident and Emergency department (A&E) of their intended arrival and through the EHR transferred patient details and current treatment details: this has also allowed the pre-printing of A&E documentation necessary for the

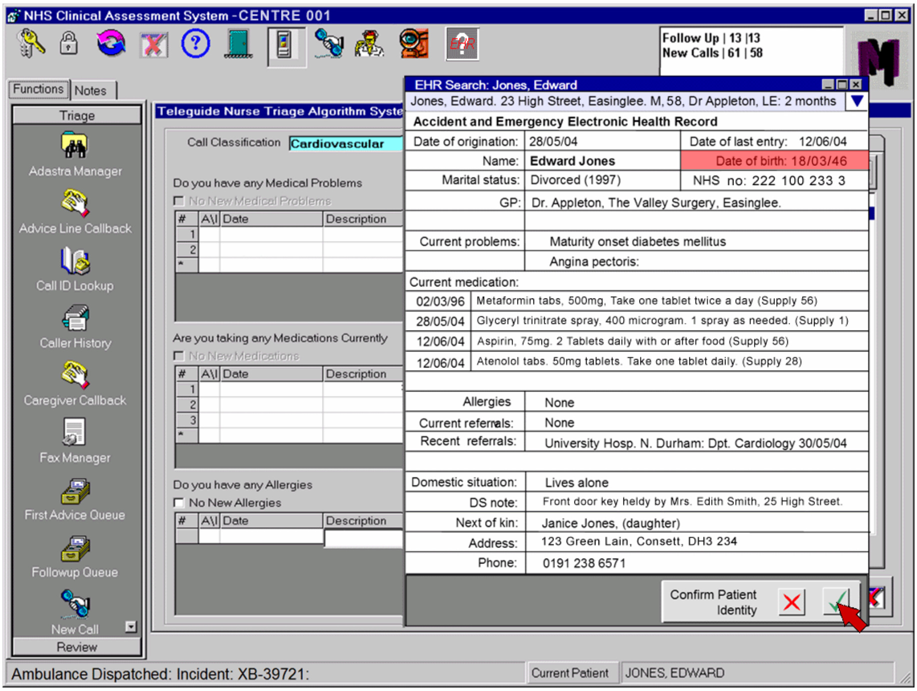

Fig. 4.3. Version 2 of the Animator showing a possible configuration of a health record

Fig. 4.4. The narrative depiction in Version 2 of the Animator

care of the patient. The ambulance is then shown arriving at A&E where the patient is then signed over to their care. The final scenario goes back in time 6 months to a general practice (GP) consultation where Mr. Jones is diagnosed with heart disease and asked if he would like to have his details on an EHR which, it is explained to him, would allow his medical details to be available to various health care professionals should they require them. Focus group participants were encouraged to discuss what they had seen and also to imagine how a similar EHR facility could impact upon their work and what the potential issues surrounding such a facility could be.

The aim was not to give a definitive account of the sorts of information that might be accessed by an EHR, nor how this might be done, but to provide a 'candidate version' based upon the architectural models and grounded in the details of the ethnography. The aim was to inform an audience to the potential of an EHR, and to allow them to critically engage with it from their own professional, or patient, perspectives. Thus, the Animator aimed to be a tool which both informed the audience and allowed them to respond to what they had seen and, through audio-data collection and questionnaires, allowing user involvement in the design process. It was hoped that this process would not only be about technical functionality, but also the workplace and social implications involved in the implementation of such a system and service within the NHS, both locally and beyond.

The Animator, a 15–20 minute multimedia presentation, was designed to be used in a focus group or discussion group environment, although we were also interested to test its ability to work in a one-to-one situation, and, until the recent closure of the NHS Information Authority, it was freely accessible over the Internet although no evaluation of its use there was undertaken.

4.5 Evaluation

Our evaluation aimed to assess the comprehensiveness and accuracy of the descriptions and explanations of health care processes in the Animator, but principally its ability to engender discussion of the DuDEHR vision of EHRs. We hoped to provoke discussion of the potential impact of EHR upon health care professionals' own work practices in the present and in the future, as well as to elicit comments on the wider ramifications of an EHR. By adopting this approach, it was hoped it would also be possible to assess the current awareness of these groups with regards to the NHS plans for a National EHR, the current developments within Durham and Darlington Heath Authority of the DuDEHR programme, as well as to engender user–designer engagement in the elaboration of plans for an EHR. The prototype was evaluated in a focus group setting to see what sort of discursive interaction and information would be generated and questionnaires were used to gauge participants' views of the Animator's validity. By validity, we mean in the sense of a comprehensibility of vision, absence of counter-intuitive features, and demonstrating a coherent knowledge of health care practice. It was not planned to use the Animator focus group sessions to inform the development of a specific EHR technology, as is often the case with the use of focus groups in technology design and evaluation (Lapinsky et al. 2001) (the ERDIP projects were all exploratory projects), but rather to evaluate the Animator's ability to fulfil the role outlined above as a proof of concept. It was anticipated that positive assessment in the questionnaire would both validate its effectiveness as a tool, and lend some credence to the results of the focus group discussions.

In all, there were 11 health care staff focus groups, held in a range of organisational contexts: five in secondary care, three in primary care,[3] one NHS Direct (the national 24/7 health call centre), one patients' council and one ambulance service. Open invitations to participate in focus groups were advertised in clinical work places and, as a consequence, participants were self-selected. The focus groups comprised of between 7 and 12 participants and included a range of health professionals and administrative staff (e.g. doctors, nurses, and secretaries). Participants were encouraged to engage in discussion and debate both before and after the Animator presentation.

Each focus group was conducted in four stages:

Baseline/pre-Animator discussion. What participants understood by shared electronic records: an open discussion with the focus group participants of what an EHR was likely to be was undertaken, including what issues they perceived to surround its form, introduction and development. The time taken for this varied between 5–15 minutes from group to group.

[3] The terms 'primary care' and 'secondary care' in the context of the UK NHS are used here as referring to: primary health care as a general practitioner or other non-hospital-based health professional; secondary health care as care provided by hospital medical specialists or staff members; tertiary health care being specialist hospitals.

Presentation. Watching the non-technical Animator presentation. A 15–20 minute presentation of the Animator in operation was shown to the assembled focus group. This was a ‘push and play’ preformatted programme designed to illustrate and provoke discussion of the issues surrounding the development of EHR.

Post-Animator discussion. Discussion of the Animator presentation and its depiction (accurate or otherwise) of the health care scenario, the role of an EHR, and the views of the focus group about this. The Animator was designed not only to raise issues, but also to allow the focus groups to develop the discussion with relevance to their own knowledge and experience.

A short evaluation questionnaire. This was a short questionnaire of four evaluation questions with an ‘Additional Comments’ box.

A focus group facilitator (KNJ) was used to promote group discussion. The aim was not to isolate individual responses, but to facilitate discussion and debate that could tap into the group’s collective responses. Using audiotapes, all focus group discussions were recorded then transcribed verbatim. Once transcribed, these tape transcripts constituted the main body of data for the Animator evaluation in terms of audience discussion and qualitative data generation. The development of the discussion in each group followed different patterns and topics emerged, which were linked to other topics in different orders by group members (Morgan 1997; Rawls 2005; Kitzinger 1993; Kitzinger 1994; Bloor et al. 2000). The analysis of the transcript data followed an inductive grounded approach (see Pope et al. 2000 for a brief outline) and produced a number of themes. It has been noted that focus groups have a contribution outside of their use in market research and have a role in participatory evaluation and practitioner evaluation (Shaw 1999) and that was the approach to their use here: although their evaluation was also implicit in that the focus groups only worked if the Animator achieved had the desired results, that is, raising discussion about EHRs.

4.6 Animator Evaluation Questionnaire Summary

Participants in the focus groups were asked to rate the following questions on a 6-point scale (Unacceptable, Very Poor, Poor, Acceptable, Good and Excellent):

1. Do you think that the scenario was realistic?
2. Was the visual presentation understandable?
3. As a way of informing you of EHRs, how do you rate the presentation? and
4. As a tool to help provoke discussion, how do you rate the usefulness of the visual presentation?’

The evaluation questionnaire responses from all the focus groups were overwhelmingly positive, the mean score for each question across all categories being approximately 4 (good). There were variations but none of these were statistically significant. When each of the groups was looked at individually there were variations across the sector of health care, but these were minimal with individual variations being explained via the individual respondent’s ‘Additional Comments’ box responses. For example, one respondent who gave a negative rating concerning the 'realism' of the Animator commented that this was because 'The "patient" was unrealistic. The speech was stilted/robotic, that is, he appeared to be sight-reading a script'.

NHS Direct's respondents' responses to this question ('Do you think that the scenario was realistic?'), while still positive, had 75% of responses in the acceptable category rather than reflecting the dominance of the 'good' category for other responses. This suggested less enthusiasm for the realism of the scenario by the NHS Direct focus group. One reason for this might be the depiction of NHS Direct by the Animator, or that NHS Direct staff are more 'switched on' to issues concerning the electronic handling of patient data. If we look at the free response comments for these questionnaires, we see the following:

Questionnaire No. 38 Additional Comments: It all seems very futuristic with negatives and positives for an all encompassing record. Initiated plentiful discussion, raised lots of questions without answers as yet; very interesting. Who would have ownership, who would be responsible for changing basic demos and who would decide how much information an individual could access?
Questionnaire No. 39 Additional Comments: Good demonstration of possibilities of EHR.
Questionnaire No.40 Additional Comments: A simplified care pathway which provoked its own questions Would be useful to see more that one pathway Useful to see how the EHR would be in an organised 'need to know basis' Would suggest further input from NHS Direct resources, that is, virtual centre, electronic records already created.
Questionnaire No. 41 Additional Comments: Good starting point – may need refining but good scenario.

Respondent 38 refers directly to the Animator's ability to stimulate discussion and restates some issues that had been raised during the discussion. Respondent 40 states that the Animator presents only a 'simplified care pathway' and suggests developing it to show more than one care pathway. The third comment made by respondent 40 concerns something they have learnt from the Animator and focus group discussion and their final comment suggests collaboration/input into the development of the EHR from NHS Direct itself. A similar positive appraisal with a need for refinement is also given in questionnaire 41. Overall, we can see that use of the Animator was viewed positively with scope for further refinement – which is what one would expect from a prototype.

Because of the generally overwhelmingly positive response to the four questions, we can feel confident that the content of the focus groups' discussion, reported below, covered most of the relevant issues for respondents. This is especially so in light of the positive evaluation of the Animator with regards its ability to 'help provoke discussion'. There were few dissenting voices, and those that did might have found the medical aspects or EHR concepts to be somewhat unrealistic. In a simplified storyline there is always the risk of an unrealistic plot and some people maybe more unhappy with this than others. However, the questionnaires indicate the vast majority found the storyline acceptable, which is perhaps best illustrated by the ability of the Animator to provoke debate, to which we now turn in our discussion of the results of the focus groups.

4.7 Focus Group Evaluation

The audiotape transcripts were analysed with the support of the qualitative data analysis package NVivo. The first few transcripts were examined for themes raised in the debates and discussions prior to and following the Animator presentation, and from these a coding frame was drawn up and then applied to all the focus group transcripts. This allowed the systematic retrieval of data by individual or multiple themes from some or all focus groups. These analytic themes allowed a systematic evaluation of the response to the Animator, while also picking up relevant development issues that could be fed back into the iterative development of the Animator and DuDEHR project.

There were 11 focus groups in all; five in Secondary Care, three in Primary Care, one Patients' Council, one NHS Direct and one Ambulance Service. The data from all these focus groups was coded with the common coding frame, as either 'Primary Care', 'Secondary Care', or 'Other': 'Other' being an amalgamation of the Patients' Council, NHS Direct and the Ambulance Service. This allowed individual themes to be retrieved under each of these groups or 'All'. The All category consisted of aggregated comments from all focus groups. This was done because, except for the main coding categories, the comments under any subcategory were too few to analyse by being split into focus group type.

A point worth noting is that the focus group discussions were free flowing. Discussions developed in each group following different patterns, raising topics in various orders and without being mutually exclusive, or fitting into only one coding category, a phenomenon that coding techniques can hide. Themes and issues would run into each other, be dropped, and then picked up again if further discussion caused reconsideration of earlier points of debate. Some groups were found to hold different views to others on the same topic, and sometimes opinions differed and changed within the same group as the discussions developed. The analysis did not track the actual development of ideas within the groups; however, it is an important finding that the Animator did provoke this type of discussion. The results were collected into the following categories: *Workload; Sharing Information; Access to Information; Record Content; Confidentiality; Patient Consent; and Implementation.* The seven topics under which the data was coded and presented are not discrete, but often overlapped. The analysis is presented in two parts: baseline and post-intervention**.** The majority of data was produced in response to the Animator in the post-intervention sessions.

4.7.1 Baseline

There was little expression of knowledge of NHS electronic records initiatives:

> …other than what I have heard from you in the packs of what we're doing, I haven't heard anything from any other sources outside, I can't even recall reading any articles or anything like that to be honest. (Focus Group 5)

The NHS Direct participants appeared the most knowledgeable and patients the least. NHS Direct had a knowledgeable view of what they wanted an EHR to be, while the Ambulance Service knew they had needs that could be potentially met by current technologies or those soon to be available. The Patients Forum were keen, but without much prior knowledge about EHR.

The baseline focus groups did stimulate some discussion, if minimal, as to what an EHR could be, for example, Primary Care members thought patients would have greater ownership of their health records, perhaps even materially in the form of a Smart Card, which would travel with the patient. They also believed that its practical use could be to make up-to-date information about the patient available to Secondary Care. Secondary Care members had views similar to Primary Care, but, additionally, that an EHR could have care pathways (formal evidence based plans of anticipated care) built into it, and that its practical use would require increased access to terminals or portable technology. Interestingly, NHS Direct focus group members raised initial issues that other focus groups raised after only having viewed the Animator, which illustrates two points: firstly, those already using advanced information systems are able to perceive further potential uses for an EHR, that is, what else an EHR might be able to do; secondly, the Animator allowed those who are not in this position to be stimulated into anticipating what the EHR in the presentation might meant for them.

4.7.2 Post-Intervention

As noted above, the Animator produced significant discussion about EHR issues that were thematically coded under: Workload; Sharing Information; Access to Information; Record Content; Confidentiality; Patient Consent; and Implementation. It is worth noting before looking at these topics that the issues raised are exactly the problems still facing the NP*f*IT programme and Connecting for Health's Care Record (Cross 2006b).

Workload

All participants agreed that the reduction of duplication potentially afforded by an EHR would be a great asset, both for staff and patients. Its impact being potentially greatest where there was a lot of printed information being sent via post or fax that could instead be sent and received digitally. Care Pathways built into the EHR were also seen as potentially eliminating duplication occurring in existing (largely paper-based) systems. Nevertheless, alongside these potential benefits there were serious concerns over a decreased workload and job losses, especially for those in Primary Care where staff often had to manually type or scan various documents into their EPR and where an EHR was seen as potentially removing these tasks. However, major concerns were also voiced about increased workload due to the administration of patients' consent; advising patients about the EHR and their records; and keeping related information up-to-date. In addition to issues of workload, there was concern expressed that training would pose significant problems in terms of differential training needs, the cost of provision, the time it would take and the difficulties of providing staff cover. These issues were felt to be especially pertinent for senior clinicians who were perceived to be less computer literate. It was also felt clinicians would claim not to have the time and that extra administrative support would need to be provided:

> it's going to impact on the secretaries' workload, it's not going to impact on the GP and consultants because I'll tell you the consultants at the hospital wouldn't put the data in, somebody would be taking it off and putting it on and writing it down for the consultants to read or printing out, they certainly wouldn't have anything much to do with it… but yeah, I think that the GPs would be very much the same (Focus Group 8)

Significantly, health care professionals also had workload concerns around the copious reading requirements of newly available records and the need for a balance: too much or too little information and health care professionals will not want to access it. These concerns were grounded in a fear of litigation brought about by substantially increased information in the patient records that could be missed at the point of reading and proved to be significant at a later date. One anticipated consequence of this was felt to be a need for additional time to produce and read notes to protect against potential litigation. Activities around reading and writing medical records, including obtaining patient consent, were also perceived by many participants as likely to require a need for a redesignation of work and changes to their organisational practices.

Sharing Information

Sharing information, the core idea of the EHR, was not without concern for participants in terms of how this might be done, by whom, and with what accountability. Sharing information across the Primary Care/Secondary Care Interface was seen as a positive development and that it could rectify the paucity of patient information available to the Ambulance Service. The timely sharing of information between the paramedics and the A&E staff suggested in the Animator was thought especially beneficial to patients' care in emergency situations. The sharing of information through an EHR was thought to facilitate developments that were already underway in Primary Care Out-of-Hours care, although a few concerns were expressed against sharing information in this arena. It was suggested, however, that patients might frequent A&E Departments instead of GP practices in the belief that Secondary Care had complete patient records. Access to records was seen as advantageous to NHS Direct but that it would require tailoring:

> I think I…to be honest I think the stuff with NHS Direct would be an improvement because now we've got nothing [yeah] the patient has rang NHS Direct but we've only got their word that they did that, we don't know what advice they gave, that sort of thing so that would be good. (Focus Group 7)

It was also felt that pharmacists could benefit from an EHR so as to access patient medication lists, allergies, and relevant active problems. It was suggested that an EHR could facilitate the perceived poor communication between Secondary Care and Mental Health Services, but it was not envisioned that this would be simple to implement in practice:

> If you are dealing with you know, drug…drug addicts, people with mental health problems where obviously you are going to be dealing with a lot of information systems, not strictly health perhaps. (Focus Group 4)

It was also envisaged that mental health patients might not sign up to an EHR if they thought their information would be shared with other agencies; that certain types of records in Mental Health may not work well in an EHR; and that there may by legal issues. However, it was believed that a shared National Service Frameworks (NSF)[4] might force the issue.

Health Services communications with Social Services was also perceived to be poor yet, while some advantages were noted with regard to child welfare, caution in this area was overwhelming, especially any link with welfare benefits and other state agencies.

> I think very few people would quibble if it's the local hospital, if it's NHS Direct, if it's the District Nurses, once you start saying 'Well of course, you know this will be available to Social Services, and that will be available to, you know, other agencies' then that's when you might run into serious problems. (Focus Group 4)

[4] National Service Frameworks are formal, long-term strategies for improving specific areas of care with set goals and set time frames. They incorporate national standards and identify key interventions for a defined service or care group.

However, while full access to records by Social Services was seen as infeasible, there was recognition that medical and social services are part of a continuum of care, although, surprisingly, there was no mention of clinical services being interested in social service records.

Access to Information

As noted above, the sharing of information invoked the issue of the structure of the NHS and overlapping boundaries and responsibilities. The practical issue of access to records involved issues of whom, when and where and in relation to what. At the same time, access was seen as tied in with confidentiality, provoking discussion of the need for restrictions, and audit trails to enable checks to be made on any access to any record:

> I mean it's theoretically possible to give access to each data field to have its own access erm…restrictions. (Focus Group 2)

While the technical possibilities were recognised, some participants were more concerned with the practicalities of access to information. A common position was that 'if people can't act on the information, why give them access to it?' But the concern here was the effort needed to segregate the data between relevant users and whether practical work activities were well-enough understood for this to be successfully accomplished. It was suggested that, in their current protocols, the work of many staff was not properly accounted for, for example:

> Yes, certainly potentially and we are now in a situation where we try to delegate a lot more data entry so that it gets done, obviously within a finite time, which means that people do have to have access to potentially sensitive information. (Focus Group 2)

If access was based upon stricter protocols, then administrative or secretarial staff may loose their current access, with the implication that clinicians themselves would have to in put more of their patients' data, a scenario seen as impractical.

The issue of patient access to their own personal records was seen as fraught with problems, including the existence of non-patient identifiers; the need for latency to prevent patients viewing inconsistent and unverified results, or viewing results before their meaning could be explained by a clinician; locations where patients could be given access; whether confidentiality from other family members would be practical, especially teenagers from parents (e.g. concerning contraception); retaining access payments; and the possibility of patient amendments to their own records. Related to this was concern as to the need for patient consent to allow health care professionals access to their records. Could patients deny record access to health care professionals and associate staff who currently have access now, but of whom the patient is unaware? Indeed, the present system was felt to involve a degree of trust that could be lost if the issue of access was mishandled.

Record Content

The content of the patient record provoked much debate, which was related to other issues, especially access. A common response was that a great deal of effort would be needed to find out what data each area of health care needs collected and to have available to it on an EHR. This seemed to reflect an awareness of the wide diversity of personnel, and variety of purposes, behind the accessing of records and documentation in the everyday practice of health care. The concern was that their current work is not accurately defined (or even definable) in a systematic way to create realistic access rights. This also reflected concerns about who did what in discussion of access levels. It was felt the record content and access rights may be easier within the development of National Service Frameworks (NSFs), but that, even here, NSFs could be an oversimplification of health care practices, especially for patients with multiple conditions. Consequently, while the role of NSFs in structuring an EHR was seen positively by some, it was felt that there could also be good medical reasons for not following an NSF, and that an EHR would need to support non-NSF treatment. A NSF content approach raised issues of sharing information as it would involve non-health service workers, as mentioned previously, thus the content could not just be 'medically' oriented:

> …now that we've got the NSF for mental health and the elderly, it very much involves social services, voluntary services, care workers, a very broad range and clearly they do need to be drawn into it. (Focus Group 2)

Nevertheless, complexity aside, there were positive attitudes to EHR content in that it could attend to the problem of timeliness:

> We have that problem with past...with paper records now…, we might have somebody discharged from a ward on Friday, they could come back into clinic and Coronary Care on Sunday, on Monday and they might have seen their GP on the weekend and things might have changed from the discharge that when they went home. (Focus Group 5)

Another issue raised was who would be responsible for data integrity and the veracity of content? Issues were also raised concerning ownership of the record content. Concerns were expressed as to whether a patient's right to access their records would also give them a say over the content. It was felt that if patients saw sensitive information about themselves on the EHR they may wish this to be screened and/or removed, including information necessary for patient care and staff awareness of potentially 'dangerous' patents.

Confidentiality

Confidentiality was another major and multifaceted theme. There was no aspect of content that could not, in some scenario or other, be envisioned as sensitive by participants. These included patient demographics, not just the difficulty of keeping them updated but what they actually contained, for example, marital status. It was felt that previously innocuous issues could become sensitive ones when they were no longer solely paper based or ‘restricted to the one location’, for example, if a patient lived alone.

In general, it was felt that there were data protection issues that needed to be addressed regarding the transfer of patient information, with different data fields needing different confidentiality status. While audit trails, as previously mentioned, could, in part, help ensure confidentiality, it was noted by participants that trust would still be an important aspect. Participants believed that they already operated within a framework of confidentiality:

> …I mean we are all bound by a confidentially anyway [sure, sure, yeah] so you know I mean you have to…we have to be very careful who we mention, who we talk about anyway so I think that would just [yeah] we should still go for that but I think erm…we are still responsible for ourselves [yeah]… (Focus Group 3)

In discussions about confidentiality it was noted that not only were patients increasingly protective of their confidentiality rights, but that they also have concerns with outside agencies and report their health issues accordingly: a concern reflected in increases in the non-disclosure of illnesses. Conversely, a concern about older patients especially was that they may just give their consent because a doctor requests and, without knowledge of the consequences, this may not be counted as be ‘informed consent’ (see below).

On the whole, participants were very sensitive to issues of confidentiality and, while professional misconduct sanctions for record misuse was agreed as necessary, so was the recognition that accidental access of a patient record was possible and needed to be recognised when interpreting audit trails. Further, while an audit trail of all persons examining a record was seen as necessary, there was concern as to how this would fit in with actual work processes, for example, a phoned request.

Patient Consent

Alongside the issue of access to patient records, there was also concern over the possible requirement to obtain patient consent for the creation of an EHR and the transfer of this information away from the site of the record’s creation. Resource provision for obtaining ‘consent’ was a concern, especially as patients were seen as needing to be informed in layman’s terms and often in their own non-English language. Consent raised issues around cost and workload (as noted above). There was discussion of whether a general consent would be needed, rather than for each data transfer. Also whether patients should have to opt-in rather than have to opt-out of an E&R. Some members focused on potential difficulties, but were aware that they were perhaps imagining the worst-case scenario, since, as the following example illustrates, this practice is already commonplace:

> People see us writing and we will say ‘Oh we are just putting the details on computer’ or ‘We are getting the details off the computer’, so, as far as they know, they’ve got something on record but at this point in time it is only the

> basic details, they don't really know that they get their X-rays looked at, that they've got the results on record, blood results that other people can access from other departments, I wouldn't imagine for 1 min that they know all of this, because it isn't explained. So we've already started doing it without their consent. (Focus Group 6)

Nevertheless, consent from certain groups such as the mentally ill was seen as problematic as this could involve the transfer of information out of the NHS.

Implementation

There was a sceptical attitude to the ability to implement an EHR stemming from a corporate memory of the previously promised systems that have then been delayed or cancelled – a scepticism, which has since been vindicated. However, there was also a reflexive awareness that negative attitudes prior to system implementation could be followed by a change of attitude post-implementation into 'how did we manage without it?' Nevertheless, there was caution expressed about the reliance placed solely upon computer technology and a perceived need for a paper-based backup and the implications this would have on workload and work practice.

There was an agreed need for a unique patient identifier to prevent the incorrect patient record being accessed: such an identifier has since been introduced.

4.7.3 Summary of Results of Focus Group Discussions

Although there was at first an apparent lack of EHR knowledge amongst health care professionals, this changed dramatically in the discussions following the viewing of the Animator. There was a massive change from the initial 'baseline' discussion and a genuine sense among almost all participants that a working EHR system, with all the problems sorted, would be a progressive development. At the same time, participants raised numerous practical concerns, many of which revolved around the practical transfer of records, and documents and access to them. Alongside these were issues of accessing too much information and that the use of the records would be too time consuming:

> I bet you there's not many people would start the first page of those [paper] notes and read every blood result, every test, every electrolyte right the way through you know for 3 inches worth of A4 paper, you know they would probably look at the last one and maybe the one before that and look for any read pages that are well thumbed pages to make sure they are not missing anything but you could have overkill you know if you are linking all these systems up, then people would just think 'Well it's not…not worth it'. (Focus Group 10)

Concerns were raised about incomplete, inaccurate, or unreliable information and also a belief that regardless of the above affordances, paper would be used anyway, alongside a computer record and this raised a practical issue of where such paper would be kept if it was not envisaged in the first place? These beliefs in the continuing utility of paper reflect those of Sellen and Harper (2002).

4.8 Discussion and Conclusions

4.8.1 The Need for 'Meaningful' User Engagement

By meaningful user engagement we do not mean user engagement at an essentially political level. Examples of this within the NHS include those occasions when prior research is undertaken to assess the requirements of health care professionals through the involvement of the various gatekeepers such as the Royal Colleges, executive committees, and union representatives who speak on behalf of their constituencies. Such bodies frequently have more political concerns in their 'rules of engagement' than the day-to-day use of technology by their members. Additionally, they are invariably not even representatives of the actual users of the technologies in question, but of the managers and employers of those who will. They therefore speak not on behalf of, but, in effect, instead of their members, that is, those who handle much of the inter-health care organisation and patient communication. These individuals may genuinely believe that they can describe the work of their members, but, in reality, may little understand the day-to-day local contingencies and the situated practices that constitute the reality of the work in question. Indeed, it is frequently in the interests of the actual user to hide these practices from these very persons and organisations. This issue is one recognised by health professionals themselves as John Powell as Chairman BMA Information Technology Committee stated:

> Consultation should not be limited to the select group of doctors with technical skills. The programme requires major changes in the ways all NHS professionals work, and the promised engagement should involve the "average" clinician in the "average" clinic (Powell 2004, p. 200).

Nevertheless, there are, at the same time, good practical reasons for the use of these types of 'users/representatives' within organisations such as the NHS, where 'traditional' forms of user involvement become problematic. The NHS is the biggest employer in Europe, and any idea of user involvement and relations between users and designers has to be, in the main, through some sort of intermediary – hence the space for some sort of representative or intermediary technology such as the Animator.

By 'meaningful' user engagement, what we mean is engaging with actual users, or intended users, directly. This, as stated, can be difficult in an organisation and potential user group as large as the NHS – but that is why there is a need for innovative solutions.

4.8.2 Animator-Assisted Mediation

Our aim here has been to introduce a prototype of a way of obtaining user engagement (and, to a limited extent, patient engagement) in the design, development, and, eventually, implementation process. The problem of traditional intermediaries, with their various agendas has been noted above, and we have attempted to develop and employ another type of intermediary to promote user involvement, the ‘Animator’. It may seem contradictory that we advocate getting directly at work practice and yet advocate the use of an Animator for extra workplace user engagement. However, it must be remembered that the Animator is based upon extensive fieldwork of relevant workplaces and is meant to supplement, and not replace, such methods. This use of an Animator, we suggest, is one that may be suitable when the potential user group is so large as to require supplementary forms of user engagement and PD.

The Animator is a novel approach, in both the development of the tool and its use and was well received by all the focus group members, scoring very highly on the post-focus group evaluation questionnaires. Prior to all the focus groups, apart from that with members of NHS Direct, there was little knowledge of EHRs generally, and the ERDIP projects in specific (including DuDEHR). Prior to the showing of the Animator, with the exception of the NHS Direct group, it was difficult for the participants to engage in a discussion of what the introduction of EHRs might involve at any level. However, following the Animator presentation, virtually all participants were able to engage in discussion about EHRs generally and what they might involve for their work specifically. They were able to participate in envisaging a future where such technology was in place, what the problems in getting to that stage might involve, and to be able to think imaginatively about what they would like an EHR to be able to do for them personally in their professional role and, just as importantly, what they did not require it to do. Of course, there was no one unanimous position with regards the EHR’s specific features and functionality, and many seemed to regard local tailoring of the system (in line with Bates 2002) as a necessary feature.

The Animator aimed to be a tool which both informed the audience and instigated further discussion in the focus groups – in this it seemed to succeed well, irrespective focus groups membership and size.

Most health care workers are expert users of documentation in their respective activities and, once encouraged, they have a wealth of observations as to potential positives and negatives of current and potential activities around EHRs.

4.8.3 The Animator in User–Designer Collaboration

The use of the Animator as a research tool constitutes a social phenomenon in its own right and any discussions it promotes, such as in the project reported here, are not independent of that local occasion, but indeed responses are designed for the rest of the focus group. What is said by participants cannot be said to be exact accounts of how things occur back in the workplace. Instead, they must be seen as responses to a specific set of circumstances: time, place, co-participants, Animator, focus group ‘leader’, and discussions arising. Nevertheless, this does not discount them as arenas in which ideas and issues can be explored, the results of which can be informative data for those responsible for designing and implementing an EHR (or NHS Care Record). At the same time, caution has to be exercised in the use of the Animator since it is an information tool and there is some potential danger of raising expectations: one has to be explicit with participants that it is a tool for exploring futures, rather than a demonstration of an actual system. Nevertheless, the Animator evaluation focus groups afforded an opportunity to inform the health care community that an EHR was being seriously contemplated, that it would affect their working practices and that they might have an opportunity to contribute to its development. The latter, would not be through the Animator, however, as it was not taken much beyond the prototype stage or deployed as part of the NP*f*IT programme. We believe this to be a lost opportunity since if this had occurred then the NP*f*IT design process could, from the start, not only have been about technical functionality, but also about the workplace and the organisational implications involved in the implementation of such a system and service. The importance of this can be seen from the fact that the focus group discussions reported above anticipated many of the issues that were to, and still do, bedevil the NP*f*IT programme of work. While use of the Animator beyond this prototyping focus group environment has not been assessed, local rather than national post-project dissemination work for the Strategic Health Authority about the NHS Care Record did use the Animator prototype in a similar fashion to the focus groups reported here and this was a direct result of its success.

As a research and data collection method, it is not a tool to be used in isolation, but, as noted above, in coordination with other user engagement techniques and ethnographic accounts. What is required is a two-way communication process, and for it to occur at an appropriate stage of the technology development, where meaningful insights can be raised, collected, and contributed to the organisational vision. It is in this regard that the Animator approach may contribute in fostering user-designer relations. However, this is not the same as influencing policy, design, and implementation in organisations such as the NHS. All research has a problem in this regard and research undertaken using the Animator is no different. Our experience was that the Animator could be designed for, and sponsored by, the NHS and, despite positive evaluation, not gain a place in practice for reasons of 'organisational politics'. The Animator, we suggest, can both engage potential users and inform them of developments and the need for their collaboration. But the commitment to meaningful user engagement in the design, development, and implementation of an EHR has to be both real and actualised. For that to occur in a programme of work such as the NHS's NP*f*ITs Care Record, the approach taken of 'retrospective user-engagement' has to be seen as the sticking plaster it is, and '…a more sophisticated approach is needed to gain the cooperation of frontline staff… better communication is essential' (Hendy et al. 2005). The literature suggests that detaching design and development from the local environment will be problematic and these problems will increase local resistance (Scott et al. 2005), something that NP*f*IT (recently re-branded as Connecting for Health) is proving to be true, and yet, potentially, could have avoided. Our position is similar to that of a recent letter writer to the British Medical Journal: 'Connecting for Health needs to hold its nerve but never underestimate the need to communicate with all the NHS staff throughout' (Young 2005). This lesson needs not just to be learned but applied in practice, and to achieve this we suggest that innovative tools such as the Animator could have a significant role to play. To be heard, users, including those in very large organisations, have to be given a voice, and the Animator-based focus group is potentially one way of helping to bring this about.

Acknowledgements

This chapter has benefited from various parties including Geraldine Fitzpatrick, Tim Rapley and Rob Wilson. I would also like to acknowledge the editors, especially Mark Hartswood.

References

Bates, D.W. (2002). The Quality Case for Information Technology in Healthcare. BMC *Medical Informatics and Decision Making*, 2(1). www.biomedicalcentral.com/1472-6947/2/7

Bloor, M., Frankland, J., Thomas, M. and Robson, K. (2000). *Focus Groups in Social Research.* Sage, London.

Braverman, H. (1974). Labour and Monopoly Capital: The degradation of work in the twentieth Century. *Monthly Review Press*, 192-5. Reprinted in *The Social Shaping of Technology*, MacKenzie, D. and J. Wajcman (1985) Buckinghan, Open University Press.

Buscher, M., Eriksen, M.A., Kristensen, J.F. and Mogensen, P.H. (2004) Ways of grounding imagination. *Proceedings PDC (Participatory Design Conference)*, *Toronto, Ontario, Canada, July 27-31,* 2004.

Cross, M. (2006a). 'Will connecting for health deliver its promises?' *British Medical Journal*, 332, 599–601.

Cross, M. (2006b). Keeping the NHS electronic spine on track. *British Medical Journal*, 332, 656–658.

Dyke, P. (2003). Healthy connections? *Public Finance* 2003; Sept. 24–26.

Goorman, E. and Berg, M. (2000). Modelling nursing activities: Electronic patients records and their discontents. *Nursing Inquiry*, 7, 3–9.

Hartswood, M., Procter, R., Rouncefield, M. and Slack, M. (2003). Making a case in medical work: Implications for the electronic medical record. *Journal of Computer Supported Cooperative Work*, 12(3), 241–266.

Hendy, J., Reeves, B.C., Fulop, N., Hutchings, A. and Masseria, C. (2005). Challenges to implementing the national programme for information technology (NPfIT): A qualitative study. *British Medical Journal*, 331, 331–336.

Helleso, R. and Ruland, C.M. (2001). Developing a module for nursing documentation integrated in the electronic patient record. *Journal of Clinical Nursing*, 10, 799–805.

Humber, M. (2004). National programme for information technology – editorial. *British Medical Journal*, 328, 1145–1146.

Jenkings, K.N. (2007). Implementation, change management and benefit realization: Investigating the utility of ethnographically enriched process maps. *Health Informatics Journal*, 13, 57–69.

Kitzinger, J. (1993). Qualitative research: Introducing focus groups. *British Medical Journal*, 311, 299–302.

Kitzinger, J. (1994). The methodology of focus groups: The importance of interaction between participants. *Sociology of Health and Illness*, 16, 103–121.

Lapinsky, S.E., Weshler, S., Mehta, S., Varkul, M. and Hallet, D. (2001). Handheld computers in critical care. *Critical Care*, 5(4), 227–231.

Martin, D., Rouncefield, M., O'Neill, J., Hartswood, M. and Randall, D. (2005). Timing in the art of integration: 'That's how the bastille got stormed'. In *Proceedings of ACM Group '05*, November 6–9th, Florida, USA, pp. 313–322.

Martin, D., Mariani, J. and Rouncefield, M. (2008). Practicalities of participation: Stakeholder involvement in an electronic health records (EHR) project. In: Voss, A., Hartswood, M., Procter, R., Rouncefield, M., Slack, R. and Buscher, M. (eds.), *Configuring User-Designer Relations: An Interdisciplinary Perspective*. Springer, London.

Morgan, D.L. (1997). *Focus Groups as Qualitative Resea*rch (2nd ed.). Sage, London.

NHS. (1998). Information for health: An information strategy for the modern NHS 1998–2005. NHS Executive, DOH, London.

NHS. (2001). NHS information authority – ERDIP: Evaluation of Electronic Patient Record Projects. January, 2001.

NHS. (2002). Delivering 21st Century IT support for the NHS, London, DOH.

NHS. (2006). *Connecting for Health* http://www.connectingforhealth.nhs.uk/ (accessed May 2006).

Pope, C., Ziebland, S. and Mays, N. (2000). Qualitative research in health care: Analysing qualitative data. *British Medical Journal*, 320, 114–116.

Powell, J. (2004). Changes must involve clinicians and show value to patient care. *British Medical Journal*, 328, 1200.

Rawls, A. (2005). Respecifying the study of social order – Garfinkel's transition from theoretical concepualization to practice in details. In: H. Garfinkel (ed.), *Seeing Sociologically*. Paradigm Boulder, Colorado.

Robinson, P. (2005). How do we set the records straight? *British Medical Journal*, 330, 315

Scott, J., Randall, T., Vogt, T.G., Kaiser, T.M. and Hsu, J. (2005). Permanente's experience of implementing an electronic medical record: a qualitative study. *British Medical Journal*, 331, 1313–1316.

Sellen, A. and Harper, R. (2002). *The Myth of the Paperless Office*. Cambridge, Mass, MIT Press.

Shaw, I.F. (1999). *Qualitative Evaluation*. London, Sage.

Sicotte, C., Denis, J.L., Lehoux, P. and Champagne, F. (1998). The computer-based patient record challenges towards timeless and spaceless medical practice. *Journal of Medical Systems*, 22(4), 237–256.

Southon, G., Sauer, C. and Dampney, K. (1999). Lessons from a failed informations initiative: Issues for complex organisations. *International Journal of Medical Informatics*, 55, 33–46.

Winner, L. (1980). Do artifacts have politics? *Deadalus*, 109, 121–136. Reprinted in MacKenzie, D. and Wajcman, J. (eds.) (1985). *The Social Shaping of Technology*. Open University Press, London.

van der Meijden, M.J., Tange, H.J., Boiten, J., Troost, J. and Hasman, A. (2000a). An experimental electronic patient record for stroke patients. Part 1: Situation analysis. *International Journal of Medical Informatics*, 58–59, 111–125, Sep.

van der Meijden, M.J., Tange, H.J., Boiten, J., Troost, J. and Hasman, A. (2000b). An experimental electronic patient record for stroke patients. Part 2: System description. *International Journal of Medical Informatics*, 58–59, 127–140, Sep.

Wanless, D. (2002). Securing Our Future Health: Taking a Long-Term View. London, DOH.

Young, R. (2005). Letter to editor. *British Medical Journal*, 331, 516.

Chapter 5

Lessons Learnt in Providing Product Designers with User-Participatory Interaction Design Tools

John V H Bonner

Department of Informatics, School of Computing and Engineering, Huddersfield University, UK, j.v.bonner@hud.ac.uk

5.1 Introduction

This study investigates how participatory design (PD) tools (adapted from human–computer interaction methods) could be provided specifically for industrial designers to conceive, develop and evaluate novel interfaces for new and innovative appliance technologies. The research was carried out over 3 years and involved a number of studies at a major European consumer product manufacturer's design group based in the UK. The challenge was to investigate if these adapted tools could be used effectively by designers (not trained in HCI or human factors-based methods) and potential users to develop novel interactive consumer product interfaces, particularly for microwave and cooker user interfaces.

The chapter begins by discussing the evaluation criteria that were adopted to measure the effectiveness of the design tools to form useful user and design requirements for a live commercial design project. The first study explored the use of a simple card-sorting tool and its impact on introducing product designers to PD methods. Lessons learnt from this study helped provide a direction for the second study that included a scenario design tool. The second study describes how the designer's confidence of using the design tools and engaging with user participants within the design process improves positively. More significantly, perhaps, is how the designers' perceptions of participants-as-designers also changes. This is followed by a discussion about the key findings from the two studies and the importance of organisational survival as a critical factor to the successful implementation of PD tools.

The fundamental philosophy of PD is to involve the users of future systems or artefacts in design activity (Greenbaum and Kyng 1991). PD empowers users by regarding the users as the domain experts and assuming that any changes to a system should improve their role within it (Schuler and Namioka 1993). Emphasis was placed on the design tools being able to consider the 'context' of user–product interaction as context influences purposeful activity (Brown and Duguid 1994; Bødker 1991; Nardi 1996).

A. Voss et al. (eds.), *Configuring User-Designer Relations*,
DOI: 10.1007/978-1-84628-925-5_5, © Springer-Verlag London Limited 2009

Structured methodologies analysing context and design have been developed (Beyer and Holtzblatt 1998). However, appropriate abstraction of contextual user–product interaction is notoriously difficult to capture, even for experts. Lewis et al. found that data capture and analysis methods had to be pragmatic to fit commercial demands (Lewis et al. 1996). Time constraints and lack of expertise required a more streamlined approach to using interaction analysis. They devised a coding scheme that was efficient with simple nomenclature. Both Lewis et al. (1996) and Beyer and Holtzblatt (1998) used a form of representational model first to evolve an appropriate and robust abstraction of the situation under study and then to test design proposals against it. Ehn (1992) suggested the use of common 'language games' between users and designers. The games need not have the same sense for users as for designers, but the rules of participation have to make sense to both. Design artefacts should not attempt to create 'pictures of reality' but should help users and designers articulate current situations and envision future ones.

Before describing the studies, a brief summary of the rationale underpinning the development of these PD tools is provided. The tools were to:

- be participatory – involving designers and users in the design process,
- be context sensitive – capturing as much of the rich context of use as possible,
- require no prior knowledge of user-centred design methods and be pragmatic in approach, and
- provide applicability and specificity to a given interaction design problem.

Making these design tools in this study suitable for non-experts made the development even more challenging. Good representational models seemed to offer a good starting point by which designers and users could articulate their needs and intentions. Therefore, the design tools were developed so that context-sensitive *design data* (i.e., user–product interaction evidence and information generated from the workshops, which may influence design decisions) could be generated in parallel to conventional design activity through the development of a series of *interaction design models*. Interaction design models would facilitate the transition from gaining design data from the real world to producing problem-specific design guidelines as a basis for usable interface design solutions.

The proposed tools would allow design solutions to emerge from users and designers working together, with users and designers, in effect, controlling the design problem and creating agreed solutions. In making judgements about the use and benefits of the tools, their effectiveness was evaluated using a range of validity and reliability criteria. Validity was interpreted as the degree to which designers could derive accurate and consistent interpretations of any design claims made through the use of the tools without misinterpretation. The reliability criteria were defined as the level of consistency in resolving the same type and number of interaction design issues across different design problems or workshops. As design data were now generated within the design process, reliability and validity needed to be examined closely, because there was now a stronger interdependency between users and designers to identify and resolve interaction design issues together. It was important that the design tools could provide a stable and robust environment in which this joint decision making could consistently occur.

Therefore, validity of these data was a concern and dependent on factors such as:

- *Design and management of design tool exercises* – this included tasks such as: the design of cards; selection and representation of objects and tasks; and the selection of workshop goals and motives.
- *Procedural control of the design tools* – the ability to: refine or adapt design tools to alter design data outputs; manage collaborative discussion between designers and user participants.
- *Quality of experiential knowledge* – how this is recognised and used.
- *Interpretation of design data* – effectiveness in interpreting data at different levels of abstraction with adequate depth and breadth.

Reliability could be affected by the following factors:

- *Design and management of design tool exercises* – this included tasks such as: accuracy of verbal and written instructions given to the user participants; recording of design tool procedures and exercises during and between their use; and choice and selection of users and designers between design tools and/or workshops.
- *Changes in procedural understanding of the design tools* – changes over time in skills to implement and control design tools by users and designers between design tools and/or workshops.
- *Consistency of experiential knowledge* – depth and breadth of designers' and users' knowledge used between design tools and/or workshops.

Little practical evidence had been found about how to assess the overall effectiveness of these design tools and therefore a wide range of evaluation criteria were derived. These included:

- Reliability and validity of the tools – as discussed above.
- Interpretation of the interaction design models – how designers and users convert outcomes into design requirements.
- Scope of the usability issues identified – what the tools uncover in terms of usability.
- Ability to support novel interaction styles – how effective the tools are in facilitating new interaction styles with products.
- Usability of the design tools – how quickly and comfortably did the designers and users work with the tools.
- Relevance of tools to designers – how did the design tools impact on current design practice.
- Likelihood of organisational acceptance – factors that would affect the adoption of the tools more widely within the organisation.

5.2 Initial Evaluation of a Card-Sorting Tool

In this study, card-sorting games were developed to explore how designers and users could collaboratively communicate their outline design and user requirements for new novel product interfaces for a future range of smart cooker interfaces. Evidence from the literature suggested that ‘card-sorting’ games might be a useful technique. Card-sorting games have been used by Muller et al. (1995), the first stage known as CARD (Collaborative Analysis of Requirements and Design) uses cards to facilitate the articulation of task and communication within working groups. The CARD approach has been modified by Lafrenière (1996) in the design of computer-based user interfaces (CUTA) to enable a simple, user-derived, task analysis to assist in interface design. Both the CARD and CUTA methods are based on cards depicting elements of tasks activity such as task objects, for instance, telephones and notepads, process-based activities like methods of working and situations; participants within the task activity are also depicted. Both methods require the participants to select task elements and place the cards in an agreed plan or sequence. Once complete, an agreed summary of their ‘representation’ is given by participants.

The attraction in adopting this form of technique was the ease with which such a method could be learned and implemented. A series of small, incremental card-sorting exercises were proposed for the first tool and were carried out in a workshop setting with designers and potential user participants.

Four different types of card-sorting tasks were designed each having a different contextual focus. The first exercise required the user participants to plan out the preparation and cooking of a meal using a series of cards describing cooking activities such as *Check carrots to see if they are ready*. Other supporting activities were described on cards, such as *Check to see if oven temperature is correct* using minimal references to technological support as possible. The intention was to allow the participants to discuss the whole cooking process and allow their personal habits and attitudes towards cooking to emerge. Participants laid cards out on the table in the form of a *task plan* describing how they would cook a particular meal.

The second exercise was designed to build on the first by inserting *function cards* containing descriptions of cooker features or technological support. Participants were instructed to add features to their task plan only if they felt it would be used. Participants would be asked to openly discuss the advantages and disadvantages of each feature before inserting a function or deciding to leave it out.

In the third exercise, participants were asked to think about a ‘week-in-the-life’ of a cooker and place cards depicting other cooking activities, ‘clean the cooker’ or ‘cook a quick snack’ under cards labelled with the days of the week. The purpose of this exercise was to explore if participants would be able to make design decisions or make inferences from their cooking habits that might affect broader or non-task-specific interface design issues. The intention of this exercise was to build up a frequency profile of usage and the type of tasks undertaken during a typical week.

In the fourth exercise, participants were provided with a series of cards containing character profiles describing fictional individuals with different levels of interest in cooking and technology. Participants ‘matched’ these characters against some of the function cards used in exercise 2. The intention of this task was to examine if the users could make ‘third party’ design decisions on behalf of fictional characters.

The card-sorting tool provided four main advantages. First, cards would provide a quick and cheap discussion mechanism or act as ‘transitional objects’ allowing more critical contextual thinking to occur about product interaction. Second, by providing a broad range of card descriptors it would allow novel concepts to be introduced without having to design the interface, allowing participants to interpret or define the cards on their own terms. Third, card descriptions could be divorced from defined or existing technology so future functionality could be discussed. Finally, the placed cards could act as a conceptual ‘interaction model’ for analysis. There was no need at this stage to have a single or coherent solution. The intention of the design tool was to arrive at a series of creative design proposals for subsequent refinement.

The researcher selected a scenario, for example, preparation and cooking of a Sunday meal. A set of speculative cards was prepared for the design team to reduce their time commitment to the study. The designers ‘walked through’ the prepared exercises, sorted the cards while acting as both designers and participants. This was to help familiarise themselves with the exercises and to look for anticipated problems or potential misunderstandings. For the main study, four groups of between 5–6 user participants were recruited for workshops sessions, which lasted about 2 hours. The participants were recruited by the design team and consisted of factory and office volunteer workers from one of the manufacturing plants. Each group was given some introductory explanation by the researcher who remained present throughout all workshops. They were asked to discuss the process as a group and arrive at consensual agreement if any differences in opinion were found.

After the exercises, participants and designers discussed their thoughts on personal cooking habits, perceptions of technology and the effectiveness of the card-sorting exercises. These were recorded on video, and notes were made during the workshops and the videotapes were analysed using the evaluation criteria.

5.2.1 Observations

All four workshops followed a similar procedural flow. The designers remained generally passive throughout the card-sorting exercises and only intervened towards the end of each workshop when more informal discussions began. In all groups, a leader or chair from the participants emerged acting as the ‘controller’ of the cards and also of group decision-making strategies. The assumed leader often re-evaluated their task plan to ensure coherence to procedural instructions.

Reliability and Validity of Design Data

The design data generated from the workshops were rich and variable, but the management and control of its production were negligible, thus resulting in poor validity. The exercises provoked wide and interesting anecdotal discussions about cooking methods, preferences and strategies for using cooking technology, but this was not controlled or steered by the designers; they did not take ownership of the problem, which could effect the production of consistent discussions about common interaction problems. This was understandable at this stage. Few facilitating procedures had been given to the designers prior to the workshops due to uncertainty about their involvement. The participants gathered contextual data, but not in a controlled or predictable manner. For example, there was a tendency to add ‘peripheral’ or unimportant cards to the task plan, like adding more utensils, rather than adding any real new tasks that provided deeper design insights.

The reliability of the design tool management was high. Participant behaviour across all four workshops was generally consistent. In all workshops there was considerable anecdotal discussion about cooking habits prompted by the card-sorting activities. Different sorting strategies were adopted within each group, but comments made by the participants were similar across all groups. However, decisions were less dependent upon the exercise task and based more on broad collective experiences. Comments were frequently based on family habits rather than being driven by the card-sorting exercises, which could affect the validity of the design data.

During all workshops, some discussion was given over to reviewing the task plan after a natural phase of cooking activity had been discussed or a reasonable ‘component’ of activity had been described through the cards. Omissions in task elements were identified through this process and improved the quality of capturing design data. This checking procedure revealed how cognitively different card-sorting activity was to real cooking tasks. Participants needed to remind themselves of procedural steps and ensure that these were accurately reflected in the task plan, as one participant reflected, ‘this is harder than doing the real thing’. The cards forced participants to deconstruct activity but not necessarily in a natural manner. In the example below, Participant 1 (P1) was confused whether a completed task (Potatoes cooked) had been represented.

P1: ‘you’ve turned them on…to put the potatoes on…didn’t you?’ [pointing to ‘Potatoes Cooked’ card]
P2: ‘Ah but the…’
P3: ‘you don’t need to put them on the hob yet’
P1: ‘should I put them in?’ [the oven]
P2: ‘go then yes…yes’
P3: ‘you’ve got to prepare your veg now’

The task plan did not naturally suggest where subtasks start or finish. Participants found it difficult to model time-related conditions that would be obvious during real activity.

During the second exercise, that is, inserting function cards into the task plan, participants took a less purposeful approach. Function cards were read out and participants arbitrarily inserted them into the task map without considering their importance or frequency of use.

P1: ‘Do you want an electric helper?’ [reading from card]
P2: ‘that would be at the beginning [of the task map] you’d want to know how to boil an egg’
P1: ‘Would you want that to be electric?’

P2: ‘Yer – she wants er…’
P3: ‘I need help when I cook’
P1: ‘Does it have to be electric or a book?’
P2: ‘A book’
P3: ‘A book might be easier to use’
P1: ‘Why – haven’t you read it? [Laughs]’
P2: ‘Do you mean an electric helper?' [Suggestion]
P1: ‘Would you use it?’
P3: ‘Yes I would’
D1: ‘What about an electronic book?’ [Suggestion – but related to one of the function cards provided]
P3: ‘That would be even better’
D2: ‘This might be an electronic note book’

This exercise, however, did prompt discussions around the activity of cooking, leading to more receptive discussions on the use of technology to support the cooking of new or unusual meals. In the dialogue above, a subgroup (P1–3) discussed how and when they might use some form of computer-based cooking assistant. They were unclear how it might function or how it might be used. The designer (D1) offered a more focussed solution but did not probe further. During one discussion of this nature, one designer was surprised to observe participants making contradictory demands for technology.

The third and fourth exercises were less successful in producing design data. In the ‘week-in-the-life’ exercise, participants generally added typical meal types under each weekday heading without discussion. The final exercise, ‘character profiles’ generated stereotypical comments about the usage of technology and again did little to reveal any insightful comments that could be effectively used as design data.

Effectiveness of Interaction Models

The task plan was intended to form the backbone of the interaction model with each exercise building up a contextually oriented representation of cooking activity. The plans, card settings and comments produced by the participants were intended as a record of design data. However, the designers paid little attention either to their construction or the completed plans. This appeared, at the time, to be disappointing. Design data, in the form of the task plan, were not formally documented but remained verbal, undocumented reactions. Little was shared between the designers about the knowledge gained.

Another purpose of the task plan was to create a common dialogue between designers and participants; clearly this did not always occur. In the example below, the designer (D) tried to identify why P had such little faith in the safety of her electrical products. Terms like ‘hot product’ are commonly used in this organisation but unfamiliar to the participant and discussion is not pursued because of this. Many of these deficiencies can be attributed to a lack of shared understanding between the designers and participants. The design tools need to ensure that this form of breakdown does not occur, this could be achieved perhaps by encouraging the designers to take more active part in the card-sorting activity.

D: ‘You mean you reset the clock every day?’
P: ‘Me electric goes off at the wall [at night time]. I could not afford for me house to be burnt down, for safety everything goes off at the wall’
D: ‘Is this because this is a “hot product?”’
P: ‘No – it’s all off at the wall’
D: ‘Everything?’ [meaning other electrical products]
P: ‘Every electric product’

Supporting Novel Interaction Styles

In designing the card-sorting activities, it was difficult to understand how novel concepts could be introduced or how they might be generated during the exercises. One approach emerged entirely by accident. Writing vague or ambiguous statements on the cards often prompted questions about their meaning or significance within the card-sorting exercises. While clarifying their meaning, suggestions were often put forward, which occasionally resulted in creative proposals.

Usability of the Design Tool

There was no doubt that the cards proved an effective vehicle for promoting discussion. The exercises themselves were not difficult to accomplish, although procedural problems were identified. For example, participants often needed reassurance on ‘rules of the game’. Task-planning exercises were very time consuming and often had to be reviewed for consistency and errors by the participants to ensure that the task map told a story.

For the designers, the level of engagement with the design tool was very low. They did not intervene in the card-sorting activities and only occasionally offered advice. When asked why they had not taken notes for use later on, they stated that they did not feel it was necessary as the process had already provided them with many new ideas. They felt they had a clear understanding of the direction they would take with future cooker interface proposals.

Relevance to Proposed Target Audience

The designers were very encouraged by the workshops and found the exercises extremely illuminating and worthwhile. One designer said ‘*in the five years I’ve been here I have never been able to gather as much useful information from users as I’ve been able to do here*’. The design tool provided an opportunity to involve users in a collaborative rather than consultative role. Persuading the design team to embark on such a process was, at times, difficult and required a great deal of ‘hand holding’ from the researcher. Many of the procedural elements of the design tool were untested, which contributed to their sense of unease about using them. Due to this uncertainty, it was difficult to assign the designers clear roles, therefore resulting in them becoming passive observers. Shifting ownership of the design tools to the designers became the next important iterative step.

After the workshops, the designers were asked to comment on the effectiveness of the design tool. Comments were very favourable but there was little evidence, apart from the task plan that other forms of design data have been captured. A summary statement was drawn up with the research investigator.

> It is important that the cooker interface instils trust to the user by providing ample feedback and information on the consequences of using any new or novel technology. New features will not be used unless the user fully understands the implications of such an action and can be confident that the action has been accepted by the cooker.
>
> The interface should avoid providing functions that are 'owned' by the cooker rather than the user, for example, timing devices where the cooker is allowed to own some time keeping tasks. Controls should always suggest that the user is in charge by permitting clear and positive feedback of their purpose but should also allow more adventurous users to feel 'master' of the cooker by allowing some controls to be configured to their own needs.
>
> The interface should provide a 'supporting' rather than 'expert' role either in terms of food safety and hygiene or in introducing the user to new methods of cooking or new types of cuisine.
>
> The key criterion, however, by which any novel features for a proposed interface must be assessed, is on TRUST.

The statement above reflected a user-centred tone with a strong emphasis on perception towards technology rather than on specific functional requirements. This was very encouraging as it was hoped that this insight would be achieved.

Likelihood of Organisational Survival

Prior to the workshops, all designers expressed concern about the card-sorting design tool and were hesitant about a process they were not directly in control of. They were unsure how participants would react to vague or unclear proposals and did not relish the prospect of deliberately placing themselves in a situation where they had uncertain or no design proposals to offer the participants. As one of the designers said, '*we don't want them going away thinking we can't design a cooker*'. Certainly the designers felt no ownership of the design tool before the workshops. However, later the reaction was very different. They were encouraged with how participants dealt with the situation and surprised at the participants' level of creativity.

5.2.2 Reflections on Study 1 and Changes for the Second Study

There was sufficient evidence to suggest that contextually based user–product design data could be gathered. The use of the design tools was clearly enjoyed by both participants and designers and acceptance for their use was achieved.

However, the interaction model (a task plan based on the layout and placement of cards on the table) was not used effectively as it could have been. The designers did not support or assist in the insertion of function cards at recognised stages or urge participants to consider the implications of adding functions into the task map. If this had been done, it may have helped participants to consider the implications of their actions more critically.

The relationship between objectives and outcomes of the card-sorting exercises needed to be more clearly defined and to be made explicit for both the designers and participants. Although a rich source of design data was gathered, the designers were not equipped to capture or control the type and quality of design data generated. More guidance by the researchers was required to help the product designers design the cards, for example, coding methods, illustrations, colour, and shape of cards. Card composition required more consideration to accurately trigger discussion about potential user behaviour and needs. The designers felt little compulsion to support their final design proposals using data gained from the card-sorting tool. To solve this problem, the designers decided to design the cards and the card-sorting exercises themselves. They streamlined the number of card-sorting exercises to one single activity, which they thought yielded the best results. The exercise in effect was a merger of the first two exercises in the first study – placement of task and function cards. To ensure more discrimination and active selection of function cards, a 'function filter' was introduced where each card had to be assessed on two criteria: frequency and importance of use. Functions that rated highly on both criteria were then introduced first into the task plan.

The quality and detail of the design data using this type of design tool was 'attenuated'. That is to say, the detail and scope of design issues discussed will inevitably be less detailed than using ethnographic or participative studies using trained designers or researchers. In the following study, attention needed to be placed on providing the right balance to gain the correct level of attenuation from the data-gathering process and the impact of using outcomes from these data to inform design decision making.

Therefore, to improve the accurate selection of important and relevant design data, a second, scenario-based, design tool was introduced. Scenario-based design methods have been used in HCI to help design complex system requirements. The main advantage of such an approach is allowing open-ended and ill-defined problems to be explored in structured and tangible ways. The complexity and subtlety of interaction makes comprehensive descriptions of activity difficult. The use of scenarios as a design tool has evolved in HCI as a mechanism to describe complex activity and allow designers to engage in and articulate design intentions (Carroll 2000; Jacobson et al. 1999) and have been successfully applied through the 'use case' approach in object-oriented software engineering (Constantine and Lockwood 1999).

A scenario-based design method appeared the most attractive option due to its flexibility and openness to interpretation. The scenario design tool permitted designers and participants to explore proposed concepts while acting or role playing within a selected scenario. The intention was that participants would be able to make more informed and context-sensitive judgements about the range of design proposals that had been suggested in card-sorting exercises. In order to make the design tools sufficiently malleable, participants used paper-based prototypes to enact their activity and walk through the interaction procedure. Amendments would be made in discussion with designers where possible new prototype variants could be rapidly introduced into the scenario.

5.3 Card Sorting and Scenario Design Tools Managed by Designers

In contrast to the first study, the designers now facilitated the workshops and made improvements, which they felt would increase their control over the process. To obtain a realistic understanding of the applicability of the design tools, experimental intervention was now kept to the absolute minimum. This was essential to ensure that natural organisational factors influence the effectiveness of the design tools, and not experimental procedure.

The design group had been commissioned to review their microwave product range, including the design of new or improved interface design functions. Two designers, who had been involved in the previous study, were provided with an hour-long tutorial explaining how to use the scenario-based design tool. A cooking scenario was selected, which the designers felt would require useful challenges in using a microwave, such as planning when and how to use the microwave.

Two card-sorting workshops were first carried out, each with five non-design employees. Figure 5.1 illustrates a card-sorting workshop in action. During the card-sorting workshops, the designers recorded any thoughts or comments on a large flip chart and photographed key events, such as the completed task map. The key outcome from the card-sorting workshops was a user-requirements brief in the form of a large (1.5 m wide and 1.0 m) board. This was used to form a tab board for the next design tool, scenario design, which was based on preferred function cards clustered into cells. Tabs were small, annotated sketches of preferred or suggested function

Fig. 5.1. Participants involved in card-sorting exercise

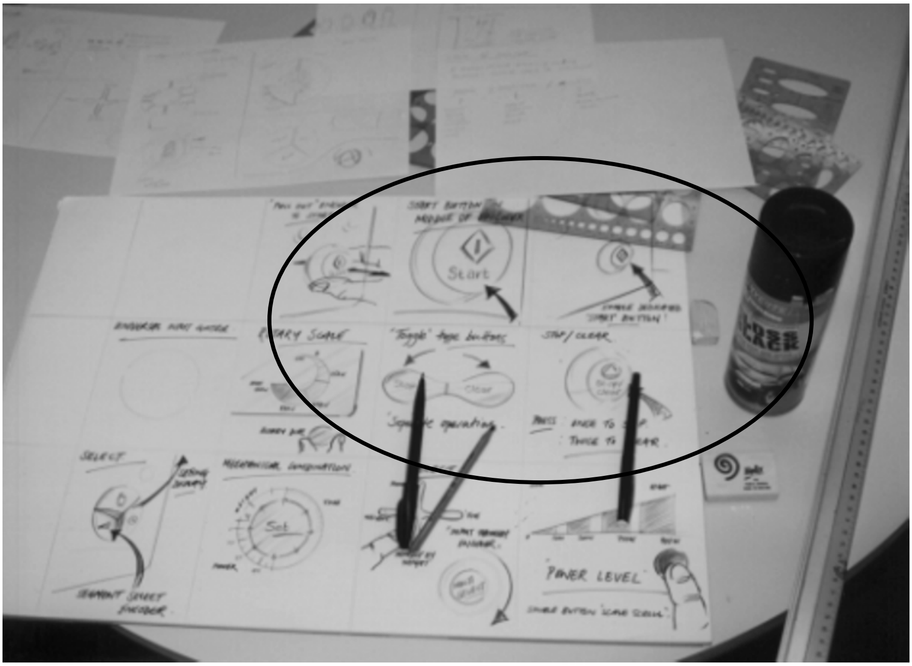

Fig. 5.2. Tabs created by designers

variants produced from the card-sorting exercise for participants to select in the scenario design workshops. Examples of tabs are provided in Fig. 5.2, illustrating variant ‘start’ controls.

The scenario design workshops were planned in a similar way to card sorting with two volunteer participants from the previous card-sorting workshops. Participants repeated the same task, but performed it as a real task using a working kitchen. Although they had to use a microwave oven, they could only operate the microwave ‘through’ the tab board and prototype interface (see Fig. 5.3, the tab board is at the rear of the picture). A conventional oven could not be used to force the participants to select and consider tabs (control and display components) and to discuss the usability of each component device.

Participants carried out the scenario by following a recipe and were encouraged to discuss their thoughts and ideas on the design of the proposed microwave interface. Alterations to any design proposal or tabs could be made at any time. The selected tabs were used to build up a paper prototype (see Fig. 5.4) based purely on the specific needs of the participants within a given scenario.

Fig. 5.3. Tab board in use

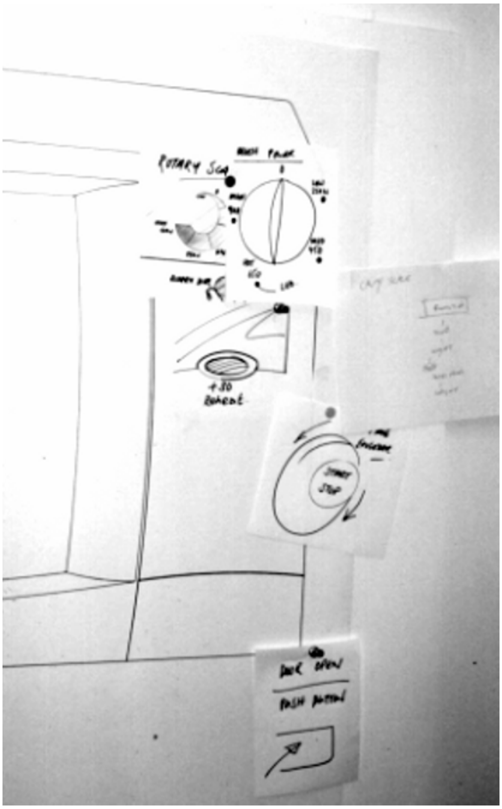

Fig. 5.4. Tabs placed on prototype interface

5.3.1 Observations

Reliability and Validity of Design Data

The design tools remained a very flexible, adaptable and potentially powerful devices for collecting design data. However, it is precisely these important characteristics that may affect the validity of the design data as this could increase the number of possible interpretations. Providing consistent instruction to the designers was at times difficult when so many of the exercises and procedures were untested. A handbook was provided with procedural instructions, but was not used in preference to being given personal instruction by the researcher. Although the designers were now facilitating the workshops, there was little evidence of managing collaborative design dialogue. They perceived their role as providers of design concepts for participants to test and to offer their interpretations on card depictions. Incomplete or ambiguous assumptions were not collaboratively discussed or jointly resolved, thus decreasing the validity of the design data. However, controlling validity was not an issue with the designers, they were more concerned with ensuring that the workshops appeared to be smooth running to the participants and that sufficient evidence was gathered to produce an internal report.

The design tools forced interaction design issues to be principally focussed at the physical device level, through a strong emphasis on depicting physical control and display elements on the cards and tabs. Other capturing methods were provided, for example, the layout of the task plan provided very useful information on where participants felt that functions could support the cooking activity. This type of design data could have helped in developing navigational support. However, the designers did not use these data even after the researcher had explicitly pointed out task plan patterns to them. A good example of this was how function cards clustered around activities at the beginning and end of the cooking task, thus giving strong suggestions to a possible navigational structure for controls and displays.

Designers created design data (cards and tabs) in quite a mechanical way. They produced simple comparative tabs, for example, different types of time controllers, for participants to comment on largely based on the outputs from the function filter used in the card-sorting tool. Tab options were refined through approving, rejecting or adapting them. Producing conventional prototype solutions reduced the likelihood of more novel and user-driven solutions to emerge. The designers were very effective at controlling tab selection. While participants were constructing their prototype interface, the designers often questioned their rationale for selecting or placing a tab. Often this was to highlight a syntax problem, but, in some cases, this was to allow the participants to consider alternative function variants. The designers did not record this form of design rationale. When questioned about this, they felt the prototype, as an outcome, offered sufficient constructive and concrete evidence. The designers also had little expectations about the design tools and were not seeking specific outcomes. For example, before the workshop, it was suggested to the designers by the researcher that cards should appear more 'rough and ready' to infer that changes and amendments could be made, thus allowing the participants to control the design of the cards. Professional pride prevented them from doing this.

The participants used the tab board (interaction design model) to great effect but, nevertheless, speculative decision making was observed.

P1 'If you cook something for 10 minutes and you take it out after 9 then as long as you don't start it again there will be a minute left in your memory'
P2 'Then you won't get your time and date back – you could leave it another day and all you would get, just sat there is 1 minute on the screen'
P1 'So when you put your next item in to cook it will automatically update itself, the clock would run in the background'
D 'I don't know, that's a feature you might decide to design, perhaps it clears itself after a period of time'
P2 'Yer, it gets bored'
P1 'auto clear'
P2 'but that might cause problems if you go away and answer the phone'

In this dialogue, a new 'automatic resetting' function was considered but decisions become based more on anticipated future behaviour rather than based on the context of the scenario. However, in another example, grounded *experiential knowledge* was used effectively. One participant struggled with the concept of representing weight on a display and suggested that it be in the form of 'bags of sugar'; this was more meaningful to her as she did not instinctively know how much a kilogram would weigh.

The *interpretation of design data* was sometimes arbitrary. Participants made too many speculative decisions that were not contributing towards effective design data. The selecting, placement and planning of cards inhibited useful consideration of real scenario-based activity. Decision making often related to stereotypical assumptions, for example:

P 'Well I could go for something as complicated as that, but I don't do a lot of cooking, but most women don't want a lot of buttons on a cooker, they want to turn it on and use it, you might be able to have a multifunction control but with a turn knob'

Or, they were based on stereotypical preferences rather than on issues generated through the design tool. This was particularly true while using the function filter.

P1 'We'll need a START or STOP won't we or STOP AUTOMATIC?'
P2 'No I have to…it's on, you normally click it up on mine'
P3 'When it gets back to zero it switches off'
D 'The bell rings'
P2 'Yes, you can stop it half way through'
P1 'So we are saying we don't want a manual – we'll do it through the timer'
P2 'So it's "quite useful" that we don't want to use it'

Here, automatic features were considered but references were made to other personal products rather than the design problem at hand thus adding to more variable design data. The graphical representation of tabs (control knobs and dials in this example) did cause problems with interpretation of particularly temporal aspects. Often designers had to explain how a function would operate.

P 'Is that … does that operate the dial?'
D1 'Set it and then activate it'
P 'Right, so you set the dial and then activate the dial'
D1 'No that's actually on the screen, it's above the dial in the screen… so you turn that round'
P 'So that's the knob on there, is that what you are saying?'
D1 'No'
D2 'Imagine that's your display it would be there'
P 'Oh Sorry'

In this example the participant has confused a display having an illustrative icon on it with a real control object that could be physically manipulated.

Adaptability of the design tools has been identified as an important aspect to improve ownership of the tools and was encouraged. Providing this form of control did, however, have an effect on the procedural understanding of methods adopted thus affecting the reliability of the outcomes. Between the first and second card-sorting exercise, the designers recognised by themselves that the first task-planning activity was too detailed and procedural. The designers had produced cards with prescriptive instructions resembling instructions from a cooking recipe. They also recognised that card depiction was 'text' heavy and more graphical images would increase interpretation and improve card recognition.

Interpretation of Interaction Design Models

When interviewing the designers after the card-sorting workshops, they appeared to have very clear views about what the participants wanted from a microwave interface. They asserted that the participants were reluctant to use non-tactile control devices, as one designer stated, '*that means no more touch screen interfaces*'. The designers concluded that any proposed interface should have no more than three control devices, although this had not been discussed explicitly during the workshops with the participants.

The function filtering activity (part of the card–sorting process to ensure that only functions deemed important by the participants are considered first) was thought to be the most productive interaction model in terms of establishing design guidance in the form of user requirements. Although the task plan and function filter were photographed, they were not analysed to produce further types of design data, such as gaining a navigation model from the task map (bottom left hand corner of Fig. 5.5). Only the preferred cards from the function filter (top right hand corner of Fig. 5.5) were retained to form tabs for the scenario design workshops.

With the tab board, both designers and participants were more critical in their evaluation, selection and use of the tabs. New tabs were devised if the participants identified an alternative or improved way of achieving a goal. In some situations

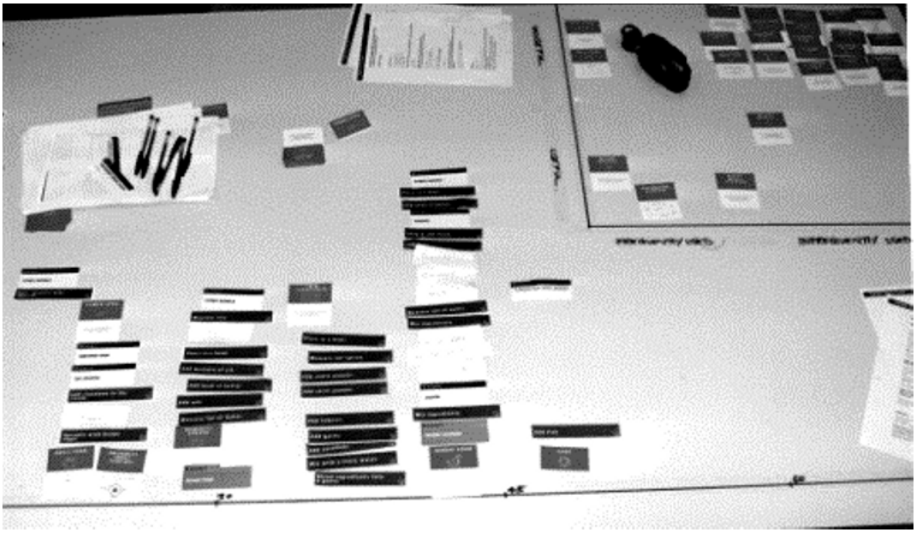

Fig. 5.5. Example of task map and function filter

participants made requests for functions that did not exist on the board. In this situation, the designer and participant would simply draw a new tab that suited their needs. The process of design, build and test could be achieved in a matter of minutes.

Scope of Usability Issues Identified

Usability issues were more comprehensively addressed with the scenario design tool, particularly issues such as consistency and compatibility. Participants did question the usability of control labels and provided more meaningful labelling suggestions; for example, replacing power levels, described in Watts, with a more contextual value such as full and half power.

Supporting Novel Interaction Styles

A range of novel concepts was introduced by the designers with the card-sorting tool, such as twin turntables, ready meal scanner, oven management system, universal input controls and menu cards. Most novel concepts were quickly rejected at the card-sorting stage. Some further degree of novelty in interaction styles was introduced in the scenario design, either by participants making requests for functionality that was not provided on the tab board, or by collaborative discussion with designers. Novel suggestions at this stage, however, were through the adaptation of control and display features already provided as tabs rather than the consideration of radically new interaction styles.

Usability of Design Tools

Early guidance and learning of the design tools came through instruction rather than reading the handbook that was provided. Progressively though, the designers used their own initiative and spontaneously resolved many procedural problems as they occurred. The designers gained confidence in using the tools and adapted the tools to improve their performance with them. Once the first workshop was complete, both designers expressed a clearer understanding of managing and conducting workshops and identified further improvements for the second workshop, for example, they revised the function filter.

Relevance of Design Tools to Designers

Initially, both designers expressed concern about the involvement of participants in design decision making. They were wary of suggesting vague design ideas to participants in fear of appearing unprofessional or inexperienced. This concern quickly evaporated once participants and designers became familiar with their roles. The relationship between the designers and participants was more consultative than participative in the card-sorting tool, but this quickly changed to positive active involvement between both groups in scenario design. External support and advice from the researcher was very important to the success and understanding of the design tools, but it was also very evident that the designers progressively gained in confidence and enjoyed using the tools. The designers were positive about the outcomes and thought they had gained useful insights that would not have been gathered otherwise.

Likelihood of Organisational Survival

The design manager and senior designer in the group were interviewed to discover how they perceived the efficacy of the design tools, the quality of the design data and interaction models and the quality of the final design solutions through reading the management report.

The design manager was extremely encouraged by the adoption of the design tools and felt that they reflected a recently implemented product development philosophy. This process was driven by the organisation's 'core values', including a user-centred approach to product development rather than by historical organisational production methods. The introduction of the design tools was also regarded as timely as the role and skills of the design team was beginning to change by developing more innovative and user-led product proposals. In order that the design tools could gain greater acceptance, approval needed to be sought at a senior management level. The design manager suggested that the design tools would only survive if the final design solutions were sufficiently creative and in line with current product development requirements. If this could be proven, the design tools could then be 'sold' outside of the immediate design group. It was important the design tools could be explained succinctly to other disparate and culturally different design groups within the organisation. This, he thought, could only be done through effective training and not through a handbook, which should only be provided as reference material. He added that the design tools would have to compete against a number of existing 'tools' used within the organisation.

The senior designer also expressed the importance of adaptability and again referred to the organisational changes that were currently underway. His view was that design skills were going to be diffused through the organisation and designers would become part of 'development groups'. In this sense, the design tools would have to be acceptable to a much broader skills base. He also agreed that the design tools could only survive if they could be integrated within their internal product development process. There was little incentive for any member to use methods not recognised as part of this process.

5.4 Reflections and Lessons Learnt from the Studies

5.4.1 Changing Designers' Attitudes towards PD Takes Time

Observations from the studies clearly indicate the designers' initial unease in involving users in the development process and allude to hesitancy and insecurity about their own role in the participatory process. The designers were familiar with the use of focus groups where consumers are consulted on their preferences to proposed design variants. However, involving consumers in the design process and designing through negotiation was anathema to their role as a designer. Trust needed to be created between the researcher and designers before these more 'radical' studies could take place. Many of the earlier iterative studies (not reported here) contributed indirectly to the building of this trust between the researcher and designers. Without these earlier iterative development studies, it would have been less likely that permission would have been gained to carry out these studies on a live project. As one of the designers reported, '*if all this goes well – it will be our idea – if it all goes wrong – it will be yours*'.

Nevertheless, despite gaining this trust the designers still needed a high level of support in implementing the design tools. At the introduction of any new design tool, they were anxious to be provided with a 'walk through' to ensure they had grasped the key concepts and flow of the workshops. The designers also spent time on careful preparation of cards and tabs. This was despite the researcher's strong encouragement to use rough cards that offered more ambiguous, less detailed information to prompt more open and potentially more fruitful collaborative discussion. This advice was ignored in favour of cleanly composed cards and tabs for reasons that were due mainly to professional pride. These resulted in some of the less-defined and potentially more interesting concepts, such as cooking management systems, being graphically represented as a complete solution, which gave the impression that they were not open for discussion. As the designers' confidence grew, and they became more comfortable with their relationship with participants, the designers were prepared to use more 'rough and ready' materials to encourage more critical debate and allowed designers and participants to changes and alter cards and tabs where necessary.

5.4.2 Interaction Design Models are Critical to Participation

Many of the reported transcripts are centred on the interaction design models: the task plan, tab board or the paper prototype interface. Much of the observational analysis highlights limitations in the dialogue between designers and participants where the interaction model has not been used as was anticipated, at least by the researcher. Despite this, each model offered an environment where candidate suggestions, with varying degrees of abstraction, could be proposed, designed and evaluated. Interaction design models are essential to effective user–designer dialogue.

5.4.3 Organisational Credibility More Important than Experimental Rigour

One of the key objectives of the study was to offer designers and participants a set of design tools that permitted a systematic way of analysing an interaction design problem and that could, through reliable and valid means, provide guidance for design solutions. One of the most interesting outcomes of these studies was the researcher's misplaced emphasis on experimental rigour. For the researcher, the critical factor for success was achieving reliability and validity in the use of the design tools. This was important because the aim was the eventual use of the design tools by designers who would not have external direction and support. It was assumed that the design tools would only be accepted by the organisation if they could be proved to provide consistent results. However, as the observations demonstrated, the designers and design manager were not concerned about this at all. Effectiveness of the tools was viewed entirely on the quality of the creative ideas generated from the participants and how these could be translated into commercially oriented solutions. Little regard was given to how this was achieved. The success of the tools was more closely aligned to satisfying complex organisational demands and product development procedures; they also needed to be marketable across different organisation groups, be quick to learn while also producing commercially appropriate design solutions.

The issue of organisational survival was investigated further. An opportunity arose where the design tools could be examined within another product development organisation. A discussion group session was set up with product designers within a user-centred design group at a telecommunications company where the feasibility of introducing new design tools into the product design group was explored. During this discussion, hurdles were identified that might impede survival of the design tools. Most of the product designers worked closely with human factors specialists, and traditionally, capture of user requirements was regarded as a human factors role. The designers needed to feel confident about using alternative design methods, which they perceived as having a human factors philosophy, before exposing their design methods to external criticism. Some concern was expressed about 'treading on human factors territory' and some of the group felt unsure about being able to provide a robust design rationale for any design proposals using this approach. Concern was also expressed about being able to gain access to users for workshop sessions. Designers were usually co-opted onto observation sessions and focus group meetings organised by other sections of the organisation. The group rarely organised participatory sessions themselves.

It was not possible at the outset of this study to know how the design tools would fit into the organisational culture but, certainly, a strong emphasis was given to providing a defensible design rationale. While this study has highlighted differing expectations between academic rigour and practical commercial needs, it has also revealed the more precarious nature of implementing any form of innovative or radical approach to product design. The successful implementation and use of PD tools must address these organisational factors before success can be assured.

5.4.4 Rules of Engagement Between Designers and Participants (and Researchers) Evolve Over Time

The reported observations illustrate how the roles of the three participant groups: researcher, design team and participants changed over time. The extracted dialogues demonstrate how participants move from 'game players' to active and engaging members of the design team. Early dialogues deal with interpreting and understanding the rules of engagement. This is replaced with more assertive discourse on how to create and use more innovative interaction styles, particularly in the scenario design workshops. The users felt empowered and gained confidence in making useful insightful comments, although this was often mixed with fanciful and unviable proposals. Nevertheless, as far as the designers were concerned, this was not a negative factor and simply added to their portfolio of possible design options. The designers were very encouraged by outcomes from the workshops and many of the participants' proposals were translated into design recommendations for a design management report.

Similarly, the designers moved from passive observers to active facilitators of a situated design process. The transcripts provide evidence of the designers' moving from supporting participants to active engagement. The designers often commented on how much insight was gained about user behaviour from the studies. Observations from the two studies highlight how designers were often surprised at the level of creativity and tolerance to ambiguity and contradictions in design decisions offered. Very often, this helped broaden the scope of possible solutions while also vindicating or rejecting preconceived design proposals. The designers, therefore, increasingly used the design tools as a mechanism for testing preconceived design proposals in preference to using them to create new interaction styles. They enjoyed the design process but, as often noted, felt little compulsion to systematically probe or document design decision-making activity.

The role of the researcher also changed. Initially, the researcher provided much of the impetus to design and implement the tools, but as the confidence was gained by the designers and participants this evolved to a passive and observation-based role. The 'summary statement' (p. 95) was an important indicator that the tools were beginning to work and the designers were beginning to view their products in terms of an interactive dialogue with consumers. Once the designers took ownership of the workshops, they could determine more precisely their own anticipated objectives.

These observations suggest the importance of recognising how rules of engagement between participants and designers will alter over time as understanding and confidence grows. In the formulation of PD methods, account should be made of evolving engagement.

5.5 Conclusions

Organisational factors eventually overrode the potential implementation of these PD methods. The organisational structure and the remoteness of the design group to key design management decision making prevented their adoption, despite local success. The studies, did, however, have a dramatic affect on the designers themselves. During an interview after these studies were carried out, two designers had reported how the studies had radically changed their perception of what a designer's role should be. They had recognised the limitations of their user knowledge and as one of them said, '*you'd think I'd know how to use a cooker*'. Furthermore, participation in the studies added to the growing disillusionment about their role and impact in product design decision making within the organisation. Two of the designers left shortly after this research (this study was not a contributing factor!) and one of them used this research as a case study in a subsequent job interview.

This study illustrates how design groups are often competing for funding, recognition and credibility amongst a number of similar or related departments. For user-participatory methods to gain broad acceptance within a manufacturing organisation, they have to be first accepted by the design group and, more importantly, they need to be confident of their acceptance within the organisation. User–designer involvement in design methods may only survive if they are organisationally marketable as well as liberating.

References

Beyer, H. and Holtzblatt, K. (1998). *Contextual Design: Defining Customer-Centred Systems*. Morgan Kaufmann, San Francisco, California.

Bødker, S. (1991). *Through the Interface: A Human Approach to User Interface Design*. New Jersey, Lawrence Erlbaum.

Brown, J.S. and Duguid, P. (1994). Borderline issues: Social and material aspects of design. *Human-Computer Interaction*, 9(1), 3–36.

Carroll, J.M. (2000). Five reasons for scenario-based design. *Interacting with Computers*, 13(1), 43–60.

Constantine, L.L. and Lockwood, L.A.D. (1999). *Software for Use: A Practical Guide to the Models and Methods of Usage-Centred Design.* Addison-Wesley, Reading, MA.

Ehn, P. (1992). Scandinavian design: On participation and skill. In: Adler, P.S. and Winograd, T.A. (Eds.), *Usability: Turning Technologies into Tools*. Oxford University Press, New York.

Greenbaum, J. and Kyng, M. (1991). *Design at Work.* Lawrence Erlbaum Associates, Hillsdale, NJ.

Jacobson, I., Booch, G. and Rumbaugh, J. (1999). *The Unified Software Development Process*. Addidson-Wesley, Reading, MA.

Lafrenière, D. (1996). CUTA: A simple, practical and low-cost approach to task analysis. *Interactions*, 3(5), 35–39.

Lewis, S., Mateas, M., Palmiter, S. and Lynch, G. (1996). Ethnographic data for product development: A collaborative process. *Interactions*, 3(6), 52–69, November/December.

Muller, M.J., Tudor, L.G., Wildman, D.M., White, E.A., Root, R.W., Dayton, T., Carr, R., Diekmann, B. and Dykstra-Erickson, E.A. (1995). Bifocal tools for scenarios and representations in participatory activities with users. In: Carroll, J.M. (Ed.), *Scenario-Based Design: Envisioning Work and Technology in System Development.* John Wiley and Son, New York.

Nardi, B.A. (1996). Studying context. In: Nardi, B.A. (Ed.), *Context and Consciousness*. MIT Press, Cambridge, Massachusetts.

Schuler, D. and Namioka, A. (Eds.). (1993). *Participatory Design: Principles and Practices*. Hillsdale, NJ, Lawrence Erlbaum Associates.

Chapter 6

A Break from Novelty: Persistence and Effects of Structural Tensions in User–Designer Relations

Sampsa Hyysalo

Helsinki Collegium for Advanced Studies, University of Helsinki,
Sampsa.Hyysalo@helsinki.fi

6.1 Reinvented Wheels and Real-Life Concerns in User–Designer Relations

Along with rapid technical development, the rise of ICTs has witnessed a rise of a family of approaches and methodologies that hold the promise in the creation of more work-affording and user-concerned technology. For instance, participatory design (PD) and user-centred systems design have demonstrated that designer–user relations can be successfully realigned in the production of new technology by involving shopfloor users in the actual design and not only treating them as informants or sources of inspiration for designers (Ehn and Kyng 1987; Greenbaum and Kyng 1991; Schuler and Namioka 1993). In the co-realisation approach, such joint design has been shown to be extendable to from early concepts to the gradual improvement of the product in the users' work practice. As the technology opens up new ways of working, it can be further modified to augment this development (e.g. Bucher et al. 2002; Hartswood et al. 2002).

However, as the organisers of the ECSCW 2003 workshop on designer–user relations put it in regard to novel methodologies: 'these approaches have not been used to their full potential', but 'used, in effect, as "patches" for more fundamental problems around user–designer relations.' A similar concern is visible in management studies. The poor understanding of user needs demarcating failed and successful innovations in 1970s innovation studies seem just as acute currently: similar problems in the coupling between design and use keep on appearing in one project after another (cf. Rothwell et al. 1974; Leonard 1995; Cooper 2004). Our experiences in two long-standing ethnographies from small high-tech companies support this view. Once we entered the ICT companies, it was striking how little the key personnel knew about the advanced methods in creating work-affording technology.

Much of the current consultancy and research literature assumes that the problem is one specific to the relative immaturity of ICT as a technological field (Norman

A. Voss et al. (eds.), *Configuring User-Designer Relations*,
DOI: 10.1007/978-1-84628-925-5_6, © Springer-Verlag London Limited 2009

1999; Kuniavsky 2003; Prahalad and Ramaswamy 2004). At the same time, ICT is often seen as a means to restructure producer–product–user relations. While there clearly is at least a 30-year continuity in how user–designer relations are organised, the question may well be posed why it is not structured by even more persisting conditions in industry. These continuities in user–designer relations are related to priorities in, and effects of, mass production, management regimes, funding dynamics of industrial organisations, increasing specialisation of expertise and so on. In being so, these structural constraints are not particularly specific to ICT or recent developments in the methods for bridging/organising design and use, but have prevailed at least since the early 20th century.

The question is not really whether or not ICT people are reinventing the wheel but what effects the wheel is likely have in CSCW. It may well turn out that precisely because 'IT systems become steadily more organisationally embedded' they tend to be mass designed and mass marketed, and be purchased in bulk in precisely the same manner as their predecessors. This may mean the waning of tailored systems in CSCW like it did for cars during the first years of the 20th century, computer hardware in 1980s or for steam engines, and men's suits much earlier (Freeman and Louçã 2001; David 1990). Future CSCW applications may be configurable and customisable, but the criteria for building them may not be derived from how well they afford work, but from the perceived competitive advantage for their producer corporations. To examine these issues, we first take an historical excursion to continuities in user–designer relations, and then examine how these continuities became visible in two development (R&D) processes in small health care high-tech companies.

6.2 Persistence of Structural Patterns in User–Designer Relations

Technology is designed, produced and used in many kinds of constellations. Yet, in the production of commercial goods, some form of industrial production has been prevalent since the proliferation of mass production in the 19th century. It is commonsense to assume that in industrial production, user–designer relations would change from one form of production to another, such as from Fordist mass production to mass customisation. The changes in the mode of production and marketing, the characteristics of products, and the changing patterns of usage would be seen to change the user–designer relation thoroughly from one type of production to another. It could be further argued that these changes would follow alongside 'long-waves,' reorganisation of industrial production that tends to follow upsurges of new generic technologies (Freeman and Louçã 2001, esp. 140–151).

Indeed, the last decade or so of literature on technology design and management gives an impression that it is specifically the rise of ICT that forces, or even allows, companies to focus on the customer relation as the prime part of their business offer (e.g. Beyer and Holzblatt 1998; Victor and Boynton 1998; Prahalad and Ramaswamy 2004). While ICT may have increased the importance of producer–user interaction, attention should also be paid to continuities that surpass several 'long waves' and reforms in industrial organisations. To illustrate these pervasive and slow-changing aspects of user–designer relations in industry, I shall elaborate how the key issues discussed, lamented, and offered as remedies for problems in user–designer relation today by far predate the transistor and all that came with it. Let us examine Fig. 6.1.

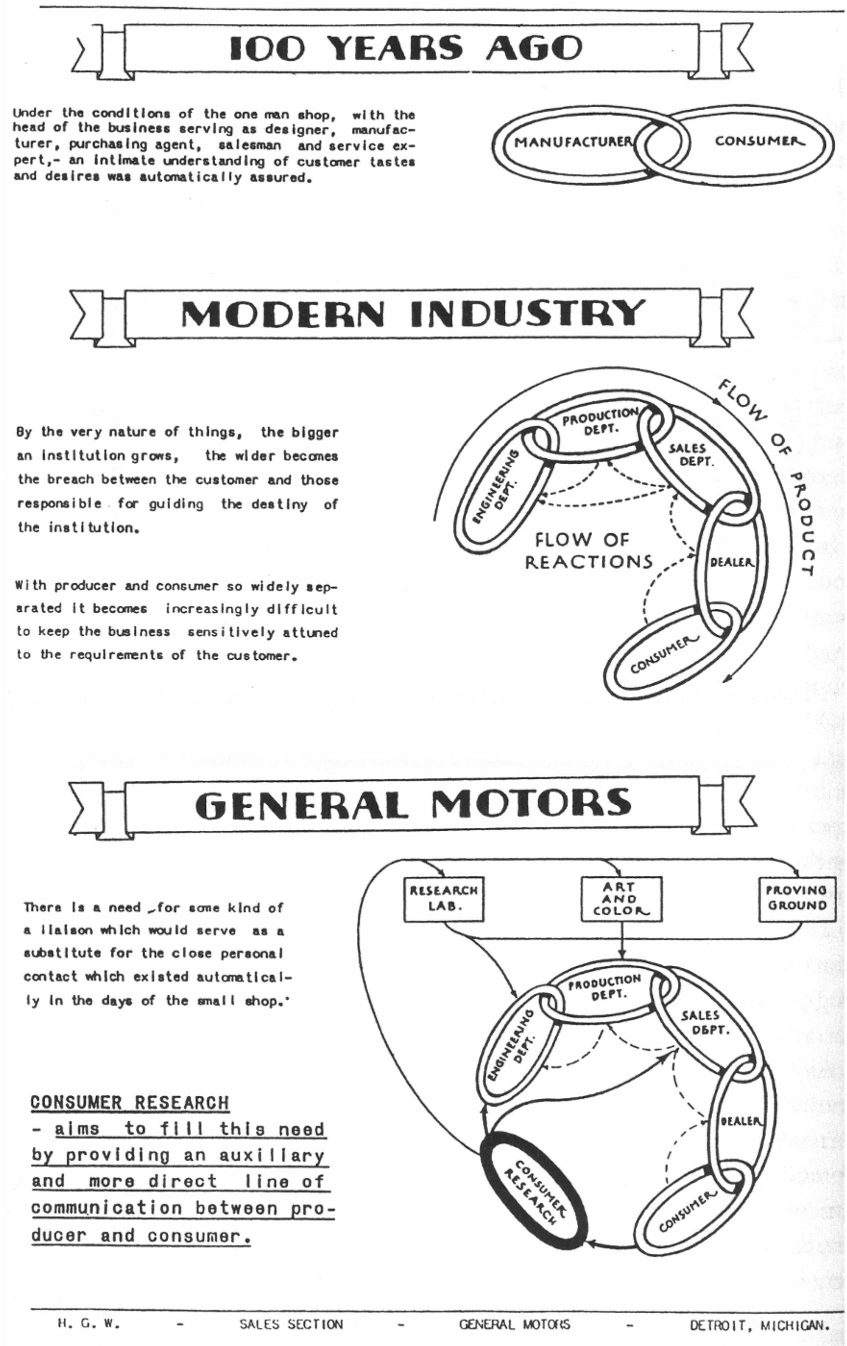

Fig. 6.1. Henry G. Weaver, diagram of consumer research[1]

[1] Weaver to O.E. Hunt, Detroit, 1932. Charles F. Kettering papers, GMI Alumni Historical Collection, Flint, Michigan. Reprinted with the permission of GM media archives.

If one did not pay attention to the caption and the lowest heading, ‘General Motors,’ the above picture could easily be mistaken for an illustration for a recent consultancy book on how industrial production should be rearranged to be more flexible, interdisciplinary, and customer centred (cf. Victor and Boynton 1998; Beyer and Holzblatt 1998; Prahalad and Ramaswamy 2004). The picture dates back to 1932. It was drawn by Henry Weaver, the manager responsible for ‘customer research’ at General Motors at the time.[2] Its resemblance to the current accounts (and to the findings of the previously mentioned comparative innovation studies in the 1970s) draws attention to the long-term continuities in the producer–user relation.

The diagram, as do most current accounts, begins with a portrayal of the golden age of handicraft production, when the designer and customer had direct personal contact and the product was customised for the needs of the client.[3] Notwithstanding how effective and common such a handicraft customisation has really been historically (cf. Ferguson 1992; Henderson 1998), the issue is the transition to ‘modern industry,’ that is, mass production, and growth in size of industrial institutions. In the middle section, Weaver implicates three reasons for the weakened ‘sensitivity to requirements of the customer.’ First, he points out the distance of the engineers to the customers (the picture) as well as to those who run the business (the text). In addition to implying increasing professional specialisation, the picture points to the increasing dispersion of knowledge within a large organisation. Second, it postulates an increasingly long chain (literally!) that mediates the product’s relationship with the customer and the feedback to the designers. Third, he implicitly draws attention to their major rival, the Ford Co., which had capitalised on the logic of mass production and the benefits of standardisation. Weaver claims that this production-centred logic runs contrary to the actual wishes of the customers. It diminishes costs but produces items that do not match the needs of any particular customer. He claims that this was so starkly undesirable by the customers, that it could be (and was successfully) used against a producer advocating high efficiency (Marchand 1998, pp. 85–100; Pantzar and Ainamo 2000). Even though these observations date back 70 years, the very same arguments can be found in most current books that seek to revolutionise industrial production.

The bottom section in Weaver’s diagram provides the answer to the separation problem: a new discipline of ‘customer research,’ which reunites design and use and intensifies the interaction within the producer company. Also this move has been popular ever since. For instance, Beyer and Holzblatt (1998) argue that marketing cannot provide a genuinely useful kind of data for design, and thus the approach of

[2] Weaver is usually credited as the person who transformed customer surveys from a scientific technique into a widespread method used by the industry (Marchand 1998). I wish to thank Mika Pantzar for drawing my attention to Weaver’s ideas and Marchand’s discussion of them.

[3] It is noteworthy that the same attributes and the shift from artisan to industrial production are lamented on similar grounds by 19th century authors, with a portrayal of a similar past golden age. Karl Marx’s lengthy discussion of the deskilling of labour in industrial production is a prime example (Marx 1990, pp. 439–553).

contextual design is needed to bridge customers and designers. Nielsen (1993) sees usability experts as this missing link. Leonard and Kuniavsky propose that there must be a range of techniques that human factors can employ to fit the nature of the technology but, in essence, she too sees that it should be the experts who master the various techniques who, once again, mediate between design and use (Leonard 1995; Kuniavsky 2003). It is also common to think of sales people as this missing link (Pinch 2003; Leonard 1995; on critical comments cf. Beyer and Holzblatt 1998).

The current candidates follow Weaver not only in his rhetoric but also in regard to how the brave new discipline should relate to users. While Weaver conducted large-scale surveys, he noted, 'The important things we have learned from guinea pigs have come – not so much from studying guinea pigs in the mass, but through conducting exhaustive experiments on a few prime specimens. GM should select out a group of "prime specimens" of its own. Such "special correspondents" or "motor enthusiasts" who took special interest in automobiles, could provide information of special value about technical issues and possible future trends.' (Marchand 1998, p. 95). There is remarkable a affinity to Eric von Hippel's 'lead-user method,' that systematised the process of seeking out and collaborating with an elite group of users (Hippel 1988; Hearstatt and Hippel 1992).

Finally, just like Weaver, the later approaches have struggled to show that their presence in the organisation truly impacts the success or customer acceptance of the products. Weaver was fully aware that the intuition of managers and engineers kept calling the shots in General Motors (GM), and that they often shunned the results he derived from the customers (Marchand 1998, pp. 103–105). In private, he confessed that much of his value to GM was derived through the marketing and public relations (PR) value that the customer research provided: the surveys were, on the one hand, clever marketing instruments and, on the other, the company appeared to be 'democratic' and 'customer centred' through the use of them (ibid. pp. 95–105). Like GM, current IT companies that emphasise users seldom fail to parade the banners of customer friendliness or a democratic attitude.

These illustrations of pervasive aspects of user–designer relations resonate with studies that have tried to characterise the general features of designer–user relations in industrial production (e.g. Grudin 1993; Suchman 2002). Producer–user relations are seen as characterised by a strict separation of production and consumption. Most products are designed and used in distinct activities by different people, means, and culture. The logics of mass production and mass marketing, the distribution of expertise, professional cultures, and boundaries, power relations within companies, and issues related to trade secrecy all effectively contribute to the prevailing separation of design and use, as well as that of designers and users (Grudin 1993; Hales 1994; Suchman 2002).

Within such an arrangement, it is typical that design precedes and predefines consumption. Even when users come up with the initial idea for a future product, as observed by Hippel (1988), most of the design decisions are done in an R&D organisation before the device reaches the hands of its eventual users. Particularly in designing for what is believed to be routine work, it has been customary for producers to see use as a question of educating the users in how to best appropriate the features of new technology (Lie and Sørensen 1996). In its starkest, the trust in the sufficiency of the design becomes visible when user training and after-sales support are outsourced, or capitalised as a further source of revenue, and in essence, regarded as deviations from the core of the technology business.

Design changes tend to follow after the market launch, as the product has faced problems in the hands of its users (e.g. Hasu 2001; Rosenberg 1982). Likewise, when people or organisations buy technology, they seldom put sufficient emphasis on the efforts and organisational changes it requires before the utility from the new technology is delivered. The performative myth about the sufficiently interacting nature of technology-in-itself may be questioned, even capitalised on in marketing and in an occasional design but, at large, the imperatives of efficient production and competition tend to assign the same clearly separated roles for users and designers.

The pervasive features in user–designer relations in technology production are particularly relevant, as they coincide with enormous technical change. The car industry in the 1930s and ICT sector since 1980s are instances where the need to bridge the gap has been clearly articulated, even though the technical constraints are vastly different in designing mostly mechanical cars and ICTs that allow multiple layers of user interface. Rather than the technological change *per se*, the question may be about the maturity of a technological field: good fit to users' practices and preferences becomes more important when competitive advantage is no longer achieved only by superior technical capability of the core technology or minimisation of its prize (Norman 1999; Rogers 1995). If we follow the theory of long waves in techno-economic development, ICTs are now at the same point at which electricity-based mechanisation of industry was in the 1930s when Weaver was writing his memos about GM and Ford (Freeman and Louçã 2001; David 1990; cf. Pantzar and Ainamo 2000). By the 1990s, transistor-based technologies had reached the point when the technologically driven early period of the development of a new kind of dominant technology, typically characterised by a technology push, was moving towards its end. As ICTs have started to spread to the fabric of society, their success may have become more dependent on meeting customer requests and on how well users' practices can be fitted to the potential of the technology in everyday arrangements (Freeman 1979; Freeman and Louçã 2001; Norman 1999). Similar patterns have been experienced with the previous dominant technologies, such as steam, electricity, and oil engines (Freeman 1979; David 1990; Freeman and Louçã, 2001).

On the other hand, it may be plausibly argued that during the last 30 or so years, the distance between designers and users has grown shorter from the rather abstract emphasis on 'customer friendliness'. This has been particularly the case with ICTs that moved towards more interactive systems from automating work routines in, for instance, office or manufacturing (Kari Kuutti, personal communication 21.10.2003). The narrow ergonomic and cognitive issues have given away to more a 'contextual' understanding of the actions and practice of users, and even towards various forms of co-design (Kuutti 1996; Star 2001). This has also been institutionalised in the teaching curricula for the engineering and design of software and 'smart products'. The emphasis on meeting customer needs is not only 'design wrapping' to create better appeal, but relates to the way ICT products require, but also enable, significant recustomisation by their end-users, which must be partially prepared for by their designers. In information systems design, a number of academically visible paradigms have emerged for thinking about and organising product development that try to avoid the common way of deploying technology from producer to users, and from management to workers (Hirschheim and Klein 1989; Gregory 2001).

An increasing amount of tools has emerged for investigating use, in traditional market research (questionnaires, interviews, segmentation, and differentiation of products), as well as in design techniques such as usability and user-centred design. However, their real use and impact to the products remains unclear (cf. Grudin 1993; Williams et al. 2000, pp. 110, 112–113). Sørensen et al's study of 63 Norwegian IT professionals suggests that the industrial use of these techniques continues along lines already weaver subscribed to: 'All the company could expect from the layman was insight into "his reactions to engineering developments" as they affected "his physique, his nerves, his temperament and… his pocket book"' (Marchand 1998, p. 105). The current emphasis on more interactive producer–product–customer relationship may simply follow from the fact that when new ICTs tend to be more interactive, it has also become more natural to discuss the user–designer relations in interactive terms (Normann and Ramirez 1994; Victor and Boynton 1998; Freeman and Louçã 2001).

6.3 Historical Continuities in Present Day Concerns: Two Cases of Health care Technology

To examine these continuities more in-depth, let us turn to our two recent longitudinal studies with ICT companies in the health care sector. They provide insight into how the above identified continuities and dynamics in user–designer relations play out at the project level. Both cases are Finnish hi-tech start-ups that started an innovative product development during the early or mid-1990s. Both companies sought niche markets that required in-depth understanding of user practice. Their business cases rested on possessing the latest technical expertise and vision of how it would bring valuable improvements in user practices. It could be argued that small start-up companies may have flexibility in relation to organising their R&D and their user relations, as they do not have a complex organisational structure and are less likely to be conservative. But new hi-tech companies also have strong pressures to get their products to market, grow, and internationalise to cover their R&D costs.

Below, the development of these two innovations is analysed in terms of their 'biography': the phases of development and turning points in either the nature of product, user–designer relations or company strategy (cf. Pollock et al. 2003; Pollock 2004; Miettinen et al. 1999). The 'biographies' of the two cases have markedly different starting points, but nonetheless featured similarities in their development. Attention is particularly drawn to the tension between the need to achieve economies of scale and packaging, and achieving a configuration that would work sufficiently in work practices of various users in multiple locations and regions.

6.4 The PDMS Development Project: Why Does PD Wane in the Wild?

The first development project sought to develop an illness-specific electronic health record for diabetes health care professionals. The database was initiated by medical researchers of the department of Public Health and General Practice at the University of Oulu. They analysed manually over 100,000 patient sheets for diabetic retinopathy at the turn of the 1990s. As a follow-up study loomed in the future, they were eager to computerise the patient records. A municipal diabetes clinic joined the pursuit as they wanted to have a statistical tool that would make it easier to follow the treatment balance of their patients.[4] With the help of a programmer from Oulu University Hospital, these users created a preliminary database with Microsoft Access.

In 1996, a small software company, ProWellness Ltd, was founded in Oulu to create an Internet-based archive for medical records. The city of Oulu recommended that the parties should engage in collaboration. ProWellness saw diabetes as a good starting point, while the users saw promise in the expertise of the cutting-edge programming firm. While users provided the details of diabetes care and practice, the company brought their programming skills and their experience in designing programs and databases for time-pressured work.

In the first phase of the design collaboration, both parties came to an understanding of what information should be included in the database and how it should be handled. The contents were solely specified by the users, who also spent time in educating the designers about diabetes treatment and the details of their work. The structure of the program evolved in the course of the collaboration. The main form of collaboration was ordinary, albeit intensive, communication between users and designers. Ideas were exchanged in face-to-face discussions, email, as well as in simple handwritten notes and drawings about the data contents and potential interface solutions. The ideas were iterated first on paper and then worked into software prototypes that were tested and developed further. The designers made the final decisions about how to incorporate the various features; however, their decisions were wholly dependent on the expertise of the medical participants. All in all, the parties were mutually dependent on the complementary resources of their counterparts.

The collaboration also quickly refined the goals of all parties. The company realised that their original archive idea had been too ambitious and too difficult to realise. Their business idea was refined into creating 'PDMS-like' expert systems for other long-term illnesses in connection with developing citizens' self-health programs for these diseases. Users appropriated the designers' idea that, using Internet technology, the database could be filled in by all key personnel and would facilitate

[4] Diabetes is an incurable long-term illness. In the long run it leads to, for instance, kidney failures, heart attacks, and blindness. These complications can be countered by maintaining 'a good treatment balance', mainly right blood-sugar level, with diet and medication. A large amount of documentation is produced and used to control the disease over the years. For this purpose, paper forms have been the main tool, currently sought to be replaced by software.

coordination between the various physicians, nurses, auxiliary nurses, as well as the specialised care given in the local hospital, thus allowing for real coverage also of type II diabetes patients. Additionally, the company envisioned a further module for the home use of patients. In this way, the database program grew to incorporate all the data generated in the treatment and monitoring of diabetes in a hospital district. The first part of this program was the physician's and nurse's screens that were piloted and further improved in the Oulu diabetic clinic, beginning in 1998.

In this early period of collaboration, participants managed to create tools and procedures that facilitated efficient collaboration between them, even though none of the participants were aware of any participatory design or user-centred design methods. However, the users already had a history of trying to create their own applications and thus had some experience in how to computerise their work practice. This shows that in certain conditions, successful in-depth user–designer collaboration can, indeed, be accomplished without specialised means. The project also shows that collaborative design is a feasible way of working for a commercial company. Had the company tried to gather all the knowledge needed about diabetes by itself, it would have travelled a long and rocky road without the needed content expertise, not to mention the consequences of them committing to their original unfeasible archive idea.

When the first version was up and running the collaboration network was extended with the help of the professional contacts of the user-partners. The new participants were physicians and nurses in the diabetes clinics in the central hospitals of Tampere and Kajaani, who were giving special care to diabetics. This extended collaboration also proved successful. In 2 years, the specialised needs of the personnel in special care were incorporated, and the usability and statistical functions of the program were significantly improved. During the year 2000, the program was bought by a number of hospital districts in Finland and the new user sites were incorporated into the development team. When the co-design work had been going on for 4 years, the program had gained a promising market share in Finland. The database had established itself as a *de facto* standard in Finland, proliferating next to all hospital districts. This is a noteworthy because previously there had been altogether 21 failed attempts to create software for diabetes care in Finland (for more detail see Hyysalo and Lehenkari 2003).[5]

But along with this success, changes also ensued in the developer conglomerate. Instead of the previously swift action to incorporate new ideas for improvement from users, the company started to take a more reserved stance towards the various wishes for customisation and new features that were voiced, particularly by new user sites. It became apparent that the company had taken the view that the program was essentially ready, and it accepted only those changes that were absolutely necessary in order to realise or fulfil deals with hospital districts.

[5] The original participants in the PDMS collaboration knew only a few of these, and the whole scope of the previous attempts in different hospital districts became visible only when we took it upon ourselves to go through all the hospital districts in Finland (Hyysalo and Lehenkari 2003).

In part, this shift in company approach reflected their increasing frustration with 'managing the cacophony of opinions' in what should and what should not be in the program. In part, it reflected the fact that company sought to wrap PDMS up as a package that could be sold without further work and costs for the company. This was further motivated by the company shifting its resources to the internalisation of their business and into making end-user programs for patients with long-term illnesses. In addition, the company management had partially changed and the new products were now being developed 'with the leading experts', instead of a multiprofessional collaboration. The company also believed that they, meaning the company, now had the in-house competence to develop databases for illnesses, not only for diabetes, but also in general.

However, during the early 2000s, PDMS was not being used in any regular health care centres in Finnish hospital districts, where the vast majority of diabetics (of type II) were being treated, and where the major impact of the program for diabetes care, as well as the company revenue (charged per patient entered into the system), resided. The expert network developing PDMS saw its use in health care centres as a matter of motivation and training. The collaboration was not extended to regular GPs and nurses, as the conglomerate believed they knew what "has to be in the program" in terms of the treatment of diabetes. But primary health carers seemed to shun this conviction by simply not accepting a program they felt was still too complex to be used along side other software in their reception work.

At the same time, the medical practitioners considered the PDMS program to be expensive. Even the early developers had to pay handsomely for the program they had been developing. Moreover, the work that users had put into the development work was acknowledged only with a brief and anonymous referral, 'developed in collaboration with users'. The company did re-establish somewhat closer user collaboration later on in 2003, and also with primary care GPs and nurses. We do not have first-hand evidence about how restricted roles users came to play, but it seems that they were not only testers (as was the case in developing the citizens screens earlier) but neither did they came to enjoy once again the position as valued sources for core ideas of development.

The major phases in the biography of the PDMS program are summarised in Table 6.1.

6.5 Outline of the Wristcare Innovation Process

The second case study, Vivago-Wristcare, is of a 'semi-medical' technology. The device was designed to monitor users' physiological state via the wrist and to generate an automatic alarm in case of medical emergency. It also has a manual alarm button and all alarms are mediated by a receiver unit and telephone network to a predestined end: to relatives, to an alarm centre, or to the nurse on call. The recipient then makes the decision on the appropriate action, for instance, calling the user, her neighbours, maintenance, or ambulance. The use of the device is thus based on forwarding the alarm within the network of care.

Table 6.1. Major phases in the PDMS product biography

	Prod-uct/concept	User–designer relation	Key concern in company strategy
Background	Earlier data-base programs	Range from close collabora-tions to over-the-walls	–
Initiation	Large reposi-tory/research tool	Mutual depend-ency, close collaboration	Creating sufficient functionality and understanding
Early development	WWW data-base covering entire diabetes care	Mutual depend-ency, close collaboration	Creating sufficient functionality and understanding
After first market launch	Database for diabetes pro-fessionals	Enlarged ring of participants	Covering the entire range of diabetes care, establishing sales
Second generation release	Database for all diabetes care	Diminished intensity and scope of collabo-ration	Proliferation and sales, packaging of product, internationalisation, other products
After second generation	Diabetes pack-age, minor customisations in Finland.	Re-intensifying of collaboration, extending it to primary care and international settings (UK, Canada)	Internationalisation, Development of other products, incremental improvement of PDMS

The concept took shape during the years 1992–1994. In 1993, the company, International Security Company Ltd. (IST), was founded to develop it. The idea arose from its inventors' experience with the development of Safety phones and alarm systems. Key solutions were also drawn from industrial automation, other monitoring devices, paramedics' diagnostic practice, elderly care, as well as technology business that the inventors were involved in previously (Nokia Ltd. and Sostel Ltd.). There were a number of internal and external investigations that assisted in defining the concept: technical feasibility and monitoring were studied with the technical research centre of the state (VTT), European markets were investigated by Strategy Analysis International (SAI) and Helsinki University of Technology (HUT) and the concept was 'test-marketed' in interviews with the inventors' elderly relatives. The investigation results were unreserved about the potential of Wristcare. Only the unorthodox methods of technical measurement raised doubts.

At this point, Wristcare was to be a one-for-all design, specially focused on the private home use for the still relatively active elderly who may face heightened risk of medial emergencies. This conception of use served as a reference for further technical development. During the years 1995–1997, the prime concern for product development was finding the right sensors, ways of measurement, and adequate algorithms. Further insight about users was generated in a design and usability study that was conducted in 1995–1996. It had hardly any immediate effects, even though it warned against some of the core assumptions made about the use of the device. The designers had already proceeded far with the design, and believed in it. The implementation of the results also coincided with a funding crisis that the company had in 1996–1997. The founder's personal assets started to run dry, and the project failed to attract investors. There was also a costly technical setback, as more sensitive sensors could not be made reliable enough and R&D had to backtrack to its earlier development path.

The first working prototypes were manufactured in 1997. First pilots started in 1997, and the product was officially launched in 1998. The product developers announced that the device was a success in technical terms. However, there were an unexpected number of false alarms and unreliabilities that had to be worked on, along with the numerous technical bugs typical for first versions of a technological device. Pilots also showed that there was a mismatch between the requirements in the use of the device and the abilities of the users. To work flawlessly, the device required sensitive use described in its manual, including specific, even though simple procedures in wearing, removing, and storing the device; cancelling false alarms, cleaning, etc. These instructions grew from 7 to 25 pages during the first 2 years of use. Even though some users were happy with the device, some had problems even in understanding how to work the single, manually operated button, not to mention handling Wristcare the way its design required. Much of the reliability of the device was on the shoulders of secondary users, nurses, and alarm centres, but the device fitted poorly with their work practices and existing instrumentations.

Between 1998 and 1999, the company made numerous adjustments and new developments, ranging from adjusting the algorithms to user training. The common denominator in these improvements was that the company recognised that the Wristcare technology had to fit better to the security service that users were fundamentally after. The product was expanded to include diagnostic software for alarms, which was soon complemented with online graphical-monitoring software 'acti-graph.' Use of Wristcare in institutions such as rest homes was augmented by developing an integrated system, which included a number of receiver units and wrist devices. During this period, experience from usage led to questioning many of the previous assumptions, such as who the users and clients were, how they worked the technology and how their condition could be monitored. In the midst of struggling to fix and improve the technology under tight finances and schedules, IST managed to sell about 1000 devices by late 1999. They also won both domestic and international innovation awards, received positive press coverage, gradually attracted increasing external investments and made distribution contracts (both domestic and international).

At this point, design of a new generation Wristcare was initiated because of problems with the existing design and its component parts. Attention was also paid to the appeal and usability of both the wrist device and the monitoring software. Gradually, partnerships developed with several user organisations and they began to be used explicitly for testing and gaining ideas for improving the design. Strategies for how the technology was presented in marketing, user training, and in dealing with the medical community were changed. The change in strategy in relating to users enabled the company to improve all aspects of the product system, particularly its control software that was also a key feature for cancelling false alarms and difficulties in fitting work practices in different rest homes and alarm centres.

However, as was the case with the PDMS program, the gradual, user-driven improvement of the technology proved to be a temporary phase in company strategy. The company deemed that the various local adaptations of the product were impossible in financial terms. The company sought to prioritise its product development efforts to most urgent developments and to achieve a standardised (even if in-site customisable) product offering to allow for the needed growth in sales and internationalisation of its business. In effect, the company tried to standardise the various local modifications made during the second round of development into well-defined product packages. Though the company still received information from the key user sites to alter the existing design, it simply saved them for a possible next generation device.

The major phases in the biography of the Wristcare product are summarised in Table 6.2.

Table 6.2. Major phases in the Wristcare product biography

Phase	Product concept	User-designer relation	Key concerns in company strategy
Background	Safety phones stabilised and wide spread in elderly care.	Strict separation of design and use.	–
Initiation	Automatic monitoring and alarming in a safety phone.	Some conversations, interviews and market studies.	Re-aligning and expanding safety phone market.
Early development	One-for-all, foolproof, Standalone Security device.	Market relation, all other ways of interacting to be minimised.	Technical realisation, economic feasibility, rapid internationalisation.
After first market launch	Own software, Institution and home versions, one-for-all.	Market relation, fixing bugs, and other maintenance.	Technical performance, reducing false alarms, internationalisation.
Second-generation release	Locally configurable product with enhanced usability.	Collaboration with some user organisations.	Persisting problems in reliability and fitting use, Internationalisation.
After 2G releases	Locally configurable system, but only from packaged parts.	Collaboration and mass sales separated moments in strategy.	Profitability, Internationalisation, fit with new environments of use.

6.6 Examining the Commonalities in User–Designer Relations of the Two Cases

There are several issues that are particularly worthy of attention in these two cases. First, neither of the companies could get by without forming collaborative relationships with several user sites. In PDMS project collaborative design had evident power in meeting the difficult design challenge. Only by surpassing the typical constellation in user–designer relations were the participants able to create a system that succeeded where 21 previous projects had failed. In the Wristcare project, designers sought to minimise contact with users, and based their design on their intuition, vision, and models derived from their professional experience. Contact and collaboration with user sites began when the original strategy failed to produce a technology that would be sufficiently fitting and reliable from the customers' perspective. This collaboration proved crucial in improving the reliability, meeting the need to customise the system, and improving its fit to users' life and work.

Second, it should be noted that in both cases the gradual improvements after the first market launch were perhaps even more significant in achieving work-and-sales affording technology than the initial design period. Thus, although user-centred design methodologies might have proven valuable, an isolated, snap-shot action, no matter how sophisticated, would have not resulted in a satisfactory result on its own.

Third, both companies were unaware of the methodologies developed for collaborating with users (e.g., user-centred design and participatory design) but, instead, relied on self-invented and often improvised means to organise partnerships and to collaborate through them. While these forms of collaboration proved beneficial, there are grounds to consider what improvements could have followed quickly and more efficiently with the use of adequate tools and ways to organise partnerships. This is particularly evident in the Wristcare case, where mutually acceptable solutions started to emerge only 3 years after the market launch and beginning of frequent interactions between designers and users. In the Wristcare case, also the investigations of use during the early stages of the product development process suffered from the same lack of knowledge, and failed to inform the company of their need to alter some of their core assumptions about use and users.

Fourth, the companies moved away from a collaborative strategy as soon they as they could, that is, after they had gained confidence that they had established sufficient functionality in their product. In PDMS, this took place despite the evident success and cost efficiency associated with receiving from users input about the contents and specifics of their work practices, which could have served also as a model for the development of other products. There are several dynamics behind this shift that draw attention to show continuities in user–designer relations come to have an effect.

Both companies faced a formidable pressure to expand their business to cover for the development costs and achieve profitability. This imposed on the companies, on the one hand, the pressure to 'generify' their product offerings, that is, to move away from site-specific versions, and to package and standardise them (Pollock 2004). On the other hand, this re-instated the tension visible in Weaver's account and other studies in user–designer relations: the companies had to package their products to be able to bring down costs but, at the same, ensure that users could sufficiently localise the products so they would perceive themselves receiving benefit for the work they needed to put in to making the technology work in their daily practices. In another words, as Weaver formulated it: people needed product variation to be able to achieve the downstream value they desired from the technology in the first place.

In regard to user–designer relations, this tension meant the companies trying to cease product variation to minimise development and production costs, while trying to answer to the localisability demand by seeking to make their generic package sufficiently on-site configurable. In effect, this first meant diminishing users' say on the design in order to bring the product more into control of the producer so as to achieve these reformulations (Aaltonen 2004). While IST had retained this control all through its collaborations, ProWellness had to acquire it from its user partners that had formidable power over the development work and the reputation of the system in the eyes of prospective customers.

However, the need to generify was by no means the sole cause for abandoning user participation in developing the product. Both companies expressed conviction that user participation was useful only for a limited period. Even if not particularly time consuming, it took some resources, which are scarce in small start-ups, away from other company actions, and the potential benefits were deprioritised when they were not seen as absolutely necessary for the immediate product generation at hand that seemed to be the horizon of actual shop floor action in both companies. Again, this dynamic is hardly new: the complexity of product, the high level of dependability required of mass production (the larger the production run, the more expensive it is to patch up any shortcomings or bugs afterwards) led the companies to focus on the 'core' of their professional expertise – in engineering and marketing – as the pressure related to it was recognisable (and obviously severe in both companies).

Moreover, the conviction that user participation should be a limited strategy was rooted in the belief in the sufficiency of the companies' technical competence in delivering successful technology. This was strongly associated with an interesting kind of organisational forgetting. When new management took over in both firms, the immediate reaction was to reinstate the need to expand and internationalise the business as rapidly as possible. This was associated with the order to standardise as much as possible, to reduce as much of local variation and customisation as was possible, and to downplay the efforts of learning and user collaboration, thus returning to the very same management ideals both companies expressed in their very beginning. And, in both companies, this shift by management was later accompanied with a shift back to more active relations with users. In terms of organisational forgetting, it is also noteworthy that the relatively high turnover of staff responsible for customer and user-related issues and/or changes in their job descriptions effectively prevented any strong policy emerging on nurturing user collaborations. New personnel (typically product managers, customer representatives) had to discover anew that knowledge acquired from users was indeed useful for design, as well as whom to contact in user sites, how to obtain information, along establishing trust anew in each site.

6.7 Small Steps That May Make a Difference

The case studies suggest that the tensions and dynamics associated with the economies achieved with generified products are likely to prevail in the industry, albeit taking different forms. This recognition leads to the advocating of three relatively modest measures that might make a difference in, at least, how big the 'patch' is that is used to remedy more deeply rooted organisation of designer–user relations:

1) Acknowledging and assisting de facto methods of collaboration and their enhancement. The use of intuition and informal (not methodologically assisted) collaborative relationships may still be the dominant ways that relate design and use. Design and use are also related by normative ways for thinking about use available in engineering and management practices, as well as 'traditional' ways to investigate use such as market surveys (Hyysalo 2003). The safety device case showed that even small steps in improving these ways of user–designer relations may have significant effect in the quality of the product. Measures such as designers personally visiting user sites, usability tests with actual users in their work site, organising pilot sites so that the goals and means of project were mutually arranged and agreed upon, were highly instrumental in providing designers with the understanding of users' work practice that allowed them to reconsider their previous assumptions and priorities as well as enabled them to learn from the use.

In the PDMS case study, plain but frequent discussions served as principal means to generate the needed functionalities. The use of lists of functionalities, paper sketches of layout, and the early piloting of prototype assisted this interaction. These ethno-methods of producers should not be overlooked even if they appear unsophisticated from the academic perspective. Furthermore, both cases bear witness to the power of ordinary communication and shared ground between designers and users in achieving beneficial collaboration: it is by no means only the specialised methods alone, but it could be argued that sustained interaction in itself maybe more important than by-the-book application of an isolated, snapshot inquiry, whatever its kind.

At the same time, it seems obvious that both companies would have benefited from even elementary knowledge of how to make their learning from use more efficient. In PDMS, the use of post-it notes and paper prototypes to construct screens would have likely saved some time and iteration from the participants. Later in its development, our exercise of mapping the dynamics of the participatory process proved instrumental in reorienting the participants so as to avoid possible problems in the future (Hyysalo and Lehenkari 2001, 2002). In the Wristcare case, early investigations of use and markets could have been sufficiently enhanced by use of more adequate descriptions and methods in gathering users' perceptions (Hyysalo 2003). During the first 2 years after the initial market launch, the measures taken later by the company could have saved time and iteration in meeting the required dependability (reliability of alarms + means available to deal with them). Moreover, during the latter period, the usability investigations, designers' observations, recording of problems and intended changes could have benefited from the use of more research-based tools.

2) Finding grammar and scalability for different methodologies. Ethnographic and PD methods often require expertise not readily available for hi-tech companies. Specialised methods for mediating design and use also tend to appear as universal fixes for user–designer relations. It is not easy for designers and engineers to determine which methods suit the different technologies best, phases in product development, environments of use, and the resources available to them. This may form an extra barrier from moving from the home-baked methods to potentially more-advanced and adequate research-based approaches. Finding a grammar and some kind of guidelines for differentiating between different approaches should be further encouraged. Equally, challenges to the continuities in user–designer relations are so formidable that some scepticism needs to be entertained in regard to trusting that R&D should move to a remarkably more collaborative direction. This calls for attempts to render various approaches to overcoming temporal, spatial, professional, cultural, etc. gaps between designers and users more appropriable for prevalent industrial R&D management regimes, as was suggested by the case studies above.

3) Paying attention to layers of configuration in CSCW. As Stewart and Williams note, current ICTs are often configurational technologies, composed of different component systems and often distributed to different organisations (Stewart and Williams 2002). More precisely, most ICTs are based on multiple layers of configuration, starting from compilations of components of hardware, forming systems from those components; integrating hardware and various software; creating working ensembles in a workplace from different devices, applications, tasks, and procedures; performing meaningful actions in work practice with the help of social and material mediators, and so on (Hyysalo 2004). All these layers require some form of design; at least configuration and artful integration of lower-level systems and tasks. The relations of design and use in technology production are thus divided into multiple steps of appropriation and design that need to be aligned to bring downstream value and upstream cost efficiency. The key empirical questions that arise from case to case are: (1) In which layers and in what ways should producer companies and their vendors involve themselves? (2) What is the appropriate level of design for users to involve themselves for their own benefit? (3) In which layers and in what form is collaborative improvement of technology with users or a research intervention most needed? These concerns are rooted in the fact the value achieved and distributed among technology producers, vendors, service providers, and end-users of the technology and related services are dependent of one another, and require resources and work from all the involved parties to be realised (cf. Normann and Ramirez 1994; Hyysalo 2004).

6.8 Conclusions

An historical anecdote was used here to suggest that there are significant, overarching continuities in the development trajectory of CSCW or, indeed, computerisation. The pervasive features of industrial organisations might begin to affect CSCW increasingly with the maturation of the field. In a sense, this holds a promise. Maturation of a technological field is typically associated with competitive advantage being increasingly gained more through meeting customer requests and desires than through increase in raw technical performance. But that maturation is equally associated with price competition through increasing standardisation and production volumes, growth of producer companies and their concentration within fewer players. For user–designer relations, this is likely to mean that producers seek to focus on competitors and their product offers, and rely on variation and the gradual emergence of 'best designs' in creating 'sales affording technologies', rather than aiming to improve user–designer relations in creation of 'work-affording technologies'. Efforts in creating more participatory design approaches may well end up being compromised in the future as CSCW becomes standardised into an engineering problem. Meanwhile, the temporal and spatial separation between design and use that remains typical in industry becomes re-enforced, along with the separation in roles of designers and users. It would not be surprising that new mediating professions would emerge, with some designer roles shifting to users after the emergence of dominant designs and users' gradual accommodation to them.

The pervasive ways of organising user–designer relations, and some priorities and rationale behind it, were visible in the small health care hi-tech companies studied. The emphasis on work affording took place only through necessity and was abandoned with the rise of sales-affording actions in the companies, even when these two were not always in obvious opposition. Issues visibly associated with this discomfort with user collaboration included, but were not limited to, company control of its product and processes, striving for packaged and easily distributable products, concentration of resources within technical development and marketing, prioritisation of pressing short-term goals over long-term planning, unfamiliarity with sustained collaboration with users, and unfamiliarity with methods and actions needed in such collaboration.

At the same time, the case studies bear witness to the importance of user collaboration and the emphasis on creating work-affording technology: the designers' vision and intuition alone were not able to bring about a successful technology. As start-ups typically focus on what is seen as 'core' design and production, they seldom have personnel with competence in user-centred methods. As a consequence, these firms easily fall back on 'common sense'. That is, prevailing models of organising design and production, and more or less improvised methods ranging from the analysis of previous products, to troubleshooting, scattered customer feedback, market research, and adjusting their product after its shortcomings have emerged in its first implementations. This brings us back to the opening of this chapter: user–designer relations are likely to persist as a key issue for technology production, both in terms of business strategy as well as a concern for the effects of new technology in the work and life of people.

References

Aaltonen, A. (2004). Asiakaslähtöisen tuotekehityksen edellytyksistä rakennusteollisuudessa (on requirements for customer centered design in construction industry). In: M. Hasu, T. Keinonen, U.-M. Mutanen, A. Aaltonen, A. Hakatie & E. Kurvinen (Eds.), *Muotoilun muutos – näkökulmia muotoilutyön organisointiin ja johtamiseen (transformation of de-*

sign – perspectives to the organisation and management of design work) (pp. 45–77). Helsinki: Teknova Oy.

Beyer, H. and Holzblatt, K. (1998). *Contextual design: Defining customer centered systems*. San Francisco: Morgan Kaufmann Publishers.

Bucher, M., Shapiro, D., Hartswood, M., Procter, R., Slack, R., Voss, A. and Mogensen, P. (2002). Promises, premises and risks: Sharing responsibilities, working up trust and sustaining commitment in participatory design projects. In: T. Binder, J. Gregory & I. Wagner (Eds.), *Proceedings of the Participatory Design Conference*. PDC 2002, June 23–25 (pp. 173–183). Malmö, Sweden: Computer Professionals for Social Responsibility.

Cooper, A. (2004). Inmates are running the asylum – why high-tech products drive us crazy and how to restore the sanity. Indiana: Sams.

David, P.A. (1990). The dynamo and the computer: An historical perspective on the modern productivity paradox. *The American Journal of Economic Review*, 80(2), 355–361.

Ehn, P. and Kyng, M. (1987). The collective resource approach to systems design. In: G. Bjerknes, P. Ehn & M. Kyng (Eds.), *Computers and democracy: A scandinavian challenge* (pp. 17–58). Brookfield, VT: Gower.

Ferguson, E.S. (1992). *Engineering and the mind's eye*. Cambridge, MA: MIT Press.

Freeman, C. (1979). The determinants of innovation – market demand, technology, and the response to social problems. *Futures* (June), 11, 206–215.

Freeman, C. and Louçã, F. (2001). As time go by: From the industrial revolutions to the information revolution. Oxford: Oxford University Press.

Greenbaum, J. and Kyng, M. (Eds.). (1991). *Design at work: Cooperative design of computer systems*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Gregory, J. (2001). Scandinavian approaches to participatory design. In: C. L. Dym & L. Winner (Eds.), *Mudd design workshop iii, social dimensions of engineering design*, 17–19 May 2001. Claremont, California, USA.

Grudin, J. (1993). Obstacles to participatory design in large product development organizations. In: D. Schuler & A. Namioka (Eds.), *Participatory design: Principles and practices* (pp. 99–119). Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Hales, M. (1994). Where are the designers? Styles of design practice, objects of design and views of users in cscw. In: D. Rosenberg & C. Hutchison (Eds.), *Design issues in cscw* (pp. 151–179). London: Springer-Verlag.

Hartswood, M., Procter, R., Slack, R., Voß, A., Buscher, M., Rouncefield, M. and Rouchy, P. (2002). Co-realisation: Towards a principled synthesis of ethnomethodology and participatory design. *Scandinavian Journal of Information Systems*, 14(2), 9–30.

Hasu, M. (2001). *Critical transition from developers to users*. Academic dissertation. Helsinki: University of Helsinki, Department of Education.

Hearstatt, C. and Hippel, E. v. (1992). From experience: Developing new product concepts via the lead user method, a case study in a “low tech” field. *Journal of Product Innovation Management*, 9, 213–221.

Henderson, K. (1998). On line and on paper. Visual representations, visual culture, and computer graphics in design engineering. Cambridge, MA: MIT Press.

Hippel, E. v. (1988). *The sources of innovation*. New York: Oxford University Press.

Hirschheim, R. and Klein, H.K. (1989). Four paradigms of information systems development. *Communications of the ACM*, 32(10), 1199–1216.

Hyysalo, S. (2003). Some problems in the traditional approaches of predicting the use of a technology-driven invention. *Innovation*, 16(2), 118–137.

Hyysalo, S. (2004). *Uses of innovation. Wristcare in the practices of engineers and elderly*. Academic dissertation. Helsinki: Department of Education.

Hyysalo, S. and Lehenkari, J. (2001). An activity-theoretical method for studying dynamics of user-participation in is design. In: S. Bjornestad, A. Morch & A. Öpdahl (Eds.), *Iris 24,* *24th information systems research seminar in Scandinavia*, August 11–14, 2001. Ulvik in Hardanger, Norway.

Hyysalo, S. and Lehenkari, J. (2002). Contextualizing power in collaborative design. In: T. Binder, J. Gregory & I. Wagner (Eds.), *PDC 2002, participatory design conference*, June 23–25, 2002 (pp. 93–104). Malmö, Sweden: Computer Professionals for Social Responsibility.

Hyysalo, S. and Lehenkari, J. (2003). An activity-theoretical method for studying user-participation in is design. *Methods of Information in Medicine*, 42(4), 398–405.

Kuniavsky, M. (2003). Observing the user experience. A practitioner's guide to user research. San Francisco: Morgan Kaufman Publishers.

Kuutti, K. (1996). Activity theory as a potential framework for human/computer interaction. In: B. Nardi (Ed.), *Context and consciousness: Activity theory and human computer interaction* (pp. 45–67). Cambridge, MA: The MIT Press.

Leonard, D. (1995). Wellsprings of knowledge: Building and sustaining the sources of innovation. Boston, MA: Harvard Business School Press.

Lie, M. and Sørensen, K. (Eds.). (1996). *Making technology our own? Domesticating technology into everyday life*. Oslo: Scandinavian University Press.

Marchand, R. (1998). Customer research as public relations: General motors. In: S. Strasser, C. McGoverns & M. Judt (Eds.), *Getting and spending: European and American consumer societies in the twentieth century* (pp. 85–110). Cambridge, UK: Cambridge University Press.

Marx, K. (1990). *Capital* (vol 1). London: Penguin Books.

Miettinen, R., Lehenkari, J., Hasu, M. and Hyvönen, J. (1999). Osaaminen ja uuden luominen innovaatioverkoissa. Tutkimus kuudesta suomalaisesta innovaatiosta (know how and the creation of new in innovation networks. A study of six finnish innovations). Vantaa: Sitra ja Taloustieto Oy.

Nielsen, J. (1993). Usability Engineering. Boston: Morgon Kaufman.

Norman, D. (1999). The invisible computer: Why good products can fail, the personal computer is so complex and information appliances are the solution. Cambridge, MA: MIT Press.

Normann, R. and Ramirez, R. (1994). Designing interactive strategy: From value chain to value constellation. Chichester: John Wiley & Sons Ltd.

Pantzar, M. and Ainamo, A. (2000). Nokia – the surprising success of textbook wisdom. Paper held at the *Strategy Processes, Innovation and Creativity stream of EGOS conference*, July 2–4, 2000.

Pinch, T. (2003). Giving birth to new users: How the minimoog was sold to rock and roll. In: T. Pinch & N. Oudshoorn (Eds.), *How users matter. The co-construction of users and technologies* (pp. 247–270). Cambridge: MIT Press.

Pollock, N. (2004). Universtiy or universality – on the establishment of the “organizationally generic”. Paper presented at *Understanding socio-technical action* – Conference, Napier University, Edinburg, UK, July 3–4, 2004.

Pollock, N., Williams, R. and Procter, R. (2003). Fitting standard software packages to non-standard organizations: The ‘biography’ of an enterprise-wide system. *Technology Analysis & Strategic Management*, 15(3), 317–331.

Prahalad, C. K. and Ramaswamy, V. (2004). *The future of competition: Co-creating unique value with customers*. Boston: Harvard Business School Press.

Rogers, E. M. (1995). *Diffusion of innovations*. New York: Free Press.

Rosenberg, N. (1982). *Inside the black box: Technology and economics*. Cambridge: Cambridge University Press.

Rothwell, R., Freeman, C., Horlsey, A., Jervis, V., Robertson, A.B. & Townsend, J. (1974). Sappho updated – project sappho phase 2. *Research Policy*, 3, 258–291.

Schuler, D. and Namioka, A. (Eds.). (1993). *Participatory design: Principles and practice*s. Hillsdale, NJ: Lawrence Erlbaum Associates.

Sørensen, K., Aune, M. and Hatling, M. (1996). Against linearity – on the cultural appropriation of science and technology. In: M. Dierkes & C. v. Grote (Eds.), *Between understanding and trust* (pp. 237–260). Amsterdam: Harwood Academic Publishers.

Star, S. L. (2001). Computers/information technology and the social study of science and technology. *In International encyclopaedia of social and behaviour sciences*. Amsterdam: Elsevier Science.

Stewart, J. and Williams, R. (2002). The wrong trousers? Beyond the design fallacy: Social learning and the user. *Presentation held at the EASST Conference*, University of York, 31 July–2 August 2002.

Suchman, L. (2002). Located accountabilities in technology production. http://www.comp.lancs.ac.uk/sociology/soc039ls.html, 2002 (28.02.2002).

Victor, B. and Boynton, A. (1998). Invented here: Maximizing your organization's internal growth and profitability. A practical guide to transforming work. Boston, MA: Harward Business School Press.

Williams, R., Slack, R. and Stewart, J. (2000). Social learning in multimedia, *Final report to European commission*. Edinburgh: Research Centre for Social Sciences, Edinburgh University, High School Yards, Edinburgh.

Chapter 7

Practicalities of Participation: Stakeholder Involvement in an Electronic Patient Records Project

David Martin[1], John Mariani[2] and Mark Rouncefield[2]

[1] XRCE Xerox, 38240 Grenoble, France, david.martin@xrce.xerox.com
[2] Computing Department, Lancaster University, Lancaster LA1 4YR, UK, jam@comp.lancs.ac.uk, m.rouncefield@lancaster.ac.uk

7.1 Introduction

This chapter considers some of the everyday practicalities of achieving participation and managing user–designer relations when delivering an electronic patient record (EPR) project within an NHS Hospital Trust. Using ethnographic, observational data, we provide examples pertinent to understanding how user–designer relations play out in an NHS setting; documenting the problems encountered by both users and designers in this process, determining how these are resolved, and looking at the process of argument and negotiation. We also seek to emphasise that user–designer relations play out within a wider project and organisational setting that impinges on the possibilities for: (1) achieving the desired levels of user participation in design and (2) achieving a design that sufficiently supports pre-existing endogenous user practices.

In his influential paper 'The Computer Reaches Out: The Historical Continuity of Interface' , Jonathan Grudin heralded the 'turn to the social' in computing:

> …with the advent of "groupware" and systems to support organizations, we are beginning to see the focus of user interface design extend out into the social and work environment, reaching even further from its origin at the heart of the computer. (Grudin 1990)

Subsequently, we have seen a widening of the remit of systems design. Design is seen less and less as a solely technical, data-centric enterprise. Instead, research has urged for the need to understand the social and organisational context within which the use of computers takes place. Computer use is embedded within social and organisational practice and procedure, both formally specified and informally realised. Technological changes do not just plug into an organisational socket and run unprob-

A. Voss et al. (eds.), *Configuring User-Designer Relations*,
DOI: 10.1007/978-1-84628-925-5_7, © Springer-Verlag London Limited 2009

lematically. New technologies impact existing practices and vice versa, sometimes to devastating effect. Concern for understanding the operation of the existing socio-technical system and organisational context stems from a desire to avoid problematic technologies, disruption to working practices, uproar on the 'shop floor', and so forth.

Such a line of argument will doubtless be familiar to those in the participatory design (PD) audience since techniques using socio-technical approaches to design have increasingly become a part of the PD repertoire (Carroll et al. 2000; Hartswood et al. 2002; Hartswood et al. 2007). Of course, PD is not just about allowing social scientific researchers into workplaces to 'study the workers'. Since its origin in Scandinavia, the notion of PD–active user involvement in work-oriented design has undergone a number of changes as it has been adopted and developed in other parts of the world (Bjerknes and Bratteteig 1995; Blomberg and Kensing 1998; Chapter 2 by Törpel et al., this volume). PD has become popular as part of a demand to attend to the lived realities of 'being user in an organisational setting' (Grudin 1990). As computer systems increasingly dominate workplaces, the problem faced by designers lies in understanding how to evolve software systems to maintain an adequate fit with existing work practices, while also achieving the desired organisational transformation. At the same time, the challenges posed by software evolution have not been matched by organisational strategies for software configuration. These difficulties are especially significant in the context of organisationally complex, large-scale work settings, where there is often a need for decision making about what level of local variation is feasible and desirable in information handling and administrative practices. While PD has been characterised as 'a heterogeneous enterprise employing a wide range of practical techniques for enabling active user participation in design' (Crabtree 1998), it conventionally deploys a number of common, even stereotypical, techniques for increasing and supporting user involvement in work-oriented design. Apart from workplace studies, most of these are devices for increasing, eliciting, and ordering user involvement in the design project through 'future workshops', 'mock-ups', 'prototyping', and 'scenario' construction.

For us, the important ideas and issues to come out of PD centre on how participation is conceived and managed, changing notions of 'the user', and adequately understanding the relationship between design, use, and the changing social and political context. PD is generally united by an ethos of empowerment and 'meaningful' involvement of stakeholders in the design of the systems they will use (e.g. Ehn 1988). This is partly politically motivated and partly pragmatically determined; it is not only about social democracy but also about the systems that stand more chance of a success when the users are able to have a stake in their development. Of course, this is necessarily a gloss of the varied and nuanced approaches and techniques employed under the banner of PD, but we feel this captures most of the essential elements.

As researchers with an established tradition of employing ethnographic studies, which we believe should be integrated within a PD approach, the previous statement certainly corresponds to our ethos of design. We use ethnography as a means of capturing details of the lived particulars of 'real-time real-world work' (cf. Hughes et al. 1992) and making these available for the purposes of design in a number of settings (e.g. banking (Martin and Rouncefield 2003); the steel industry (Clarke et al. 2003)). This needs to be complemented with consultation with stakeholders about our explications of their work and we believe they should also be properly involved in the design process; being involved in procurement, design, evaluation, and so forth. Thus stated, our position is clear, and the wealth of studies within the PD and related computer-supported cooperative work (CSCW) and human–computer interaction (HCI) literature that use a similar approach with positive outcomes would suggest that our position is shared by a growing number of researchers and practitioners. However, despite this apparent success, we still see industrial and commercial systems being built on a regular basis where we can analyse problematic outcomes as at least partially a product of a lack of a systematic attempt to understand the wider social context of use and involve stakeholders 'properly' in design. Of course, we acknowledge that many things may be implicated in the project's 'success' or 'failure' (Blythin et al. 1997) and that there are no independent or scientific criteria for making these judgements. Nevertheless, given our belief that a lack of systematic approach to PD, incorporating ethnographic study, is implicated in failure, we are bound to investigate the reasons for the failure.

7.1.1 A National Health Service (NHS) Trust

This chapter reflects on the fieldwork carried out as a part of 3-year project, beginning in 2003. The purpose of this project was to investigate the implications for the NHS of the introduction of EPRs within all hospital 'Trusts' (comprising a few hospitals in a geographical area) in England and Wales. This system-deployment work is being undertaken as part of the NHS connecting for health (CfH) programme, which aims to provide EPRs for all Trusts that will integrate with general practitioner's (GP's) systems and a national database, often called 'the spine'. The UK is still very much in the throws of this process and a recent series of articles in the press have drawn attention to the fact that the cost of CfH has doubled to £12.4 billion and the delivery of EPRs is well behind schedule. Most EPRs are to be delivered through a national structure of service providers but some Trusts were allowed to proceed with implementations of EPRs with previously selected service providers. The study reported in this chapter relates to one of these 'early adopters', a Trust located in the north of England. Reflecting back, we can see how our research did indeed, as we had hoped, provide findings of relevance for the future development of integrated care records in the NHS. Unfortunately, they drew attention to the problems rather than providing easy solutions or a case study of best practice. Phase 1 of the Trust's system was delivered a year late in February 2005, while the more complex, and potentially medically useful aspects of the system such as medical records online, electronic drug prescription, care pathways[1], and access to medical information systems and services intended to be delivered by 2006 are still some way off. Indeed,

[1] These are standardised, computer-based processes for delivering treatment consistently, in line with best practice 'standards', for all patients. Computer-based care pathways are an integral feature of EPR systems are they are currently conceived.

our research instead pointed to the fact that the NHS had underestimated the complexity of the design problem, had been overambitious, and had not appreciated the tensions that lay between standardisation and the need to support local work practices (Martin et al. 2005; Martin 2006).

7.1.2 Delivering the Electronic Patient Record

Satisfying the growing demand for improved coordination and cooperation between health care providers presents a major challenge for health care planners. CfH sets out to meet this challenge through the adoption of EPRs and other technologies. The patchy success record of past electronic medical record projects, however, makes it clear that the problems the EPR faces are numerous and complex. Integrated healthcare records systems are seen as making available more and better-quality data, leading to better treatment and the realisation of 'seamless' health care. But progress has fallen short of expectations and studies cast doubt on whether the EPR can actually deliver the anticipated improvement in information collation, distribution, and use, and promote service integration (Ellingsen and Monterio 2000; Hanseth and Monteiro 1998; Hartswood et al. 2003).

One strand of the criticism has suggested that the lack of a socio-technical approach to design involving active user participation has been implicated in the failure of such projects (Berg 1997; Heath and Luff 1996). The basic argument is that the problems stem from an inability to understand the social organisation of current practices with paper records and a failure to design in concert with medical and administrative staff. We believe these difficulties are well-documented and -known. As a consequence of this, and in-line with our opening remarks, we did not want to simply conduct another study that compared ethnographic explications of current practice with ill-fitting technological solutions in an ironicising exercise. Instead, we were offered an opportunity of access to the project team, charged with the delivery of an EPR, to support health care delivery and attendant administrative processes across two hospitals. This allowed us to investigate the nature of *participation* in 'the wild' and to see whether through understanding it we could begin to understand why socio-technical approaches to design and user participation were underutilised and unsuccessful in the design process for complex systems.

What is clear from our fieldwork is that the designers and analysts aspired to user participation, understanding, and involvement from the stakeholders, and documenting how work was carried out locally. This was not just the official line from the NHS or the government – such talk permeated the work of the project analysts, in their own discussions and their work with stakeholders. However, to the observer (as well as to the personnel themselves), this is not recognisable as PD as written about, nor as ethnography. Interestingly, this is not because similar types of activities do not occur, because they do, rather because there is no systematic approach. Participation and activities designed to understand current practice are distributed sporadically across areas of the project. They are varied and can sometimes be seen as ad hoc. It would be disingenuous to suggest that the reason for this is ignorance or carelessness, as project members are seen to regularly bemoan the compromises they must make concerning the work they would like to do on these very topics. They complain that participation and understanding of practice and how it may be transformed for the better is good in some areas, and bad in others. What this clearly points to – the key analysis of this chapter – is that participation, in this real-world setting, is socially and organisationally contingently constructed. What participation is, what form it takes, and where are matters for local negotiation? Explicating this allows us to consider the reasons why socio-technical PD is not comprehensively realised in a complex setting like this. Subsequently, we can outline the challenges for our community.

7.1.3 Setting, Study, and Method

This chapter presents findings from a study that has been investigating some of the everyday practicalities of delivering an EPR project within a hospital Trust. The emphasis is on the EPR as a project – which needs to be managed in order to be successful. It recognises that many high-profile health care IT projects, like the EPR, are carried out within an organisational and political environment that commonly threatens to overwhelm and sink the project. We look at the everyday work of the project, of the mundane and routine concern with addressing organisational contingencies and constraints.

The often reported features of the NHS, such as the complexity of the institutions involved, their lack of modernity in terms of structure and operations, and the limited resources available for IT development (not only financially but also the difficulty in recruiting top-level expertise) impact EPR projects. Legislation dictates a public–private partnership (PPP)[2] be signed for the contract to deliver the EPR and support it. Bespoke solutions are too pricey to be considered (although might have been rejected anyway); so the job of procurement is to find a provider with a pre-existing system that can be tailored for the hospitals' purposes. We studied the Trust during Phase 1 of a 3-phase £8.3 million EPR project, which was to be delivered as a PPP in partnership with an Anglo-US company, we call 'OurComp'. The EPR system delivered was a commercial off-the-shelf (COTS) system OurComp had originally developed for the US health care market. Phase 1 involved the configuration and implementation of the core administrative system and connected reporting system, accident and emergency (A&E), theatres, order communications, and integration of the system with legacy pathology systems. After the system went live, a year late in February 2005, the project moved onto phase 2, which involved electronic medical records and drug prescription. However, this is still to be delivered, and phase 3, which involves, for example, care pathways is some way off.

Our research uses ethnographic methods, with their emphasis on workplace studies and the 'real-world, real-time' everyday character of work. Using ethnographic, observational techniques we document how and in what ways the orderly character

[2] In a PPP arrangement a private supplier works in concert with a governmentally owned institution to deliver a building, a product or a service. In this case both the Trust and the supplier – 'OurComp' – work to configure and implement OurComp's COTS system. OurComp will then administer and support the system for 9 years.

of EPR project work is achieved and delivered with a particular focus on how work practice is understood and how participation is managed. The defining feature of this kind of study is the immersion of the researcher in the work environment, where a non-presumptive record is made of many aspects of the day-to-day work over an extended period of time. This involved ‘shadowing’ the internal project team leader as they went about their everyday work as well as observing internal implementation analyst meetings, joint US/UK analyst meetings, project leaders meetings, IT communications strategy meetings, and meetings with medical staff in their departments. In this way, a ‘thick description’ (Geertz 1973) -of the situated working practices associated with the EPR project is produced. The project manager has the responsibility for information provision and distribution, and coordinating activities amongst internal teams and with the system provider. Implementation team meetings are the arena in which practical project activities are reported, discussed, negotiated, planned, and decisions made. They are attended by the project manager, analysts from all implementation teams, programme support representatives, trainers, and US analysts via a teleconferencing system. These meetings tend to be fairly technical in nature and involve reporting on progress, issues, and concerns. These team meetings provide an opportunity for people to orient to the project as a totality and provide some correspondence between what project members should be and are doing. At the same time, the project manager uses team meetings to keep people informed, thereby keeping any progress or problems visible.

7.2 Managing Participation and Understanding Work Practice

The manner in which the project is conceived and managed clearly has strong implications on how the design process is achieved. The approach and the tools of management partially dictate how descriptions of current work process and practice are produced and the manner in which they are used in design. The same is true for how stakeholder participation is realised. However, although the design method and the tools of management are the same across project areas, as is the desire to implement these in a uniform fashion, a range of organisational contingencies work against uniform implementation. In the following sections we will describe some of the ways in which understandings of work practice and the nature of participation are determined by both project management and organisational contingencies.

7.2.1 Getting a Project to Work

Our observations of the implementation of an EPR project indicates a number of ways by which the contingencies and uncertainties of organisational and project life can be handled. Most obviously, planning is a way of managing contingency – but, of course, plans do not implement themselves, but have to be made to work in ‘real world, real time’. As Button and Sharrock note, organising a project into ‘phases’, for example, is intended to ensure that tasks are worked on until completion, to achieve a paced sequential progression in the work and provide for the recognition of uncompleted steps (Button and Sharrock 1994). All phases are planned in advance in terms of what they consist of and when they will take place – identifiable major phases in this project include: procurement, award and signing of contract, 'data collection', 'database building and configuration', 'application testing', 'integration testing', and finally 'go-live and transition management'. Phasing exhibits some sensitivity to timelines of practical decision making by specifying considerations relevant to a decision prior to any deliberation. Phases may be (almost certainly will be) delayed, tasks reallocated, items of the contract, and hence the phasing, renegotiated and redefined. Nevertheless, phasing remains a key resource for the ongoing practical management of the project – enabling the distribution and coordination of work, allocating responsibilities, keeping track of activities, and measuring work progress.

Phasing also relates to another aspect of practical project management, the methodical handling of tasks (or at least maintaining the semblance of method) and some way of measuring progression – how they are doing, how much has been done, where they are, and what remains to be done. This involves maintaining the agenda of tasks, ordering, sequencing, allocating, managing, and keeping track of progress and problems through the 'issues' and 'risks' logs. In this fashion, the project manager can determine where they are relative to the project schedule, and whether the work, going at the pace it is now being conducted, will be done by the scheduled date. The following four quotes, from statements at project meeting made by the 'project manager' illustrate attempts to keep a project 'up-to-speed':

> And if I can just ask everyone to keep doing that I think we have to be very pro-active and keep emailing your analyst and say what do you want me to work on, what d'you want me to do… I'm getting nervous for a variety of reasons … I'm just not sure what they're going to throw back at me… I just want to make sure we're covering our bases as well.

Of course, 'slippage' from the plan is a 'normal, natural trouble' and its importance or magnitude is measured against the schedule:

> …there was fifty three days where we were looking at database configuration and I've said that now there's, not to scare anyone, twenty eight days left … twenty eight business days left before … its in the plan its identified that we're going to start testing, we've not done any configuration

Where 'slippage' does occur, contingency plans are made by reference to possible implications:

> …it may be that we'll have to go with the idea that they don't interface in phase one… but we'll carry on in discussing it um, further just to sort of look at all of the implications around it and I'm hoping that its not as… Its more annoying than anything right now if the truth be told, but in term of the scope of the overall project I think there's ways we can get around it without making it… um… to specific too much of an impact on the end user

Such solutions often involve considering various workarounds:

> …we need to start thinking about …how we would deal with that if-if we can't get Telepath linked um, we just need to start thinking what are our options whether people continue ordering micro or on paper, or whether we have …electronic ordering ... I think we just have to look at all the different options … of how to deal with it without, sort of, causing sort of too much, damage, to the microbiology staff but also without too much impact on the end user

Taking this approach to design means that participation is framed in particular ways and the same is true in terms of the methods used to document and understand current processes and practices. Participation in particular phases is given a particular character. For example, the procurement and award of contract phase is largely handled by the Information Management and Technology (IM&T) programme board. This group comprises the chief executive of the board, senior clinicians, senior managers, and senior IT staff (including the project manager). Thus, at this stage, participation is managed through senior 'representative' stakeholders who are intended to stand for the best interests of their staff and the hospital. It is only during the subsequent phases that we begin to see a wider range of stakeholder participation. Successively lower levels of management are appointed to working groups in a hierarchical project structure. Firstly, we have a series of groups that have project-wide responsibilities – such as for determining training requirements, selecting technologies, and providing information to the staff. Secondly, we have a team of eight project leaders –; each responsible for managing a single implementation team. Each implementation teams has the responsibility for the 'design', configuration, testing, and implementation of a single module for phase one of the project. Our main work has focused on the 'the sharp end of the scalpel' in production; the work of the implementation teams, their managers, and the project manager.

Implementation teams comprise a team leader (usually clinical, or from a clinical background), at least one UK analyst and one US analyst, and various user (clinical or administrative) representatives from that area. The user representatives may be from various levels in that area and are termed 'expert users' during building and testing, and then renamed as 'super users' once the system goes live, as their job moves from informing design to training others. They are the key personnel whose job is to provide process descriptions and give details of practice to the analysts for the purpose of database building and configuration and subsequently, they are the participants in testing who evaluate the design for usability, 'fit' with work practice, and so forth. When the system goes live, their job is to train, advise others, and troubleshoot. In this fairly traditional project structure, the more strategic a decision the more likely it is to be taken by a group further up in the hierarchy, where participants are less likely to be typical, everyday users of the system. 'Ward floor' workers are involved once the technologies and the means of designing and deploying them are already selected, meaning that participation has a particular delimited character. However, given that, we can see that the mechanisms are in place with the intention of involving 'ward floor' stakeholders in the requirements gathering, build, configuration, testing, and deployment of the system. Theirs is the participation intended to communicate what current practices and processes are, how to fit this to how the system is set up, how activities may be better realised or supported by the system, how well any subsequent design works, and so forth.

Therefore, the ways in which the project is formally managed has a significant influence on the nature of participation and how an understanding of current practice and process is achieved and used for design. For example, nurses and secretaries do not have input into the selection of suppliers, the communications strategy, or how training should proceed; these are jobs for more senior clinical or administrative managers. The approach to design does not involve an ethnographic study of current practice before selecting a COTS product. Indeed, here the COTS system was selected with only a minimal data-centric specification of the system requirements, and certainly no detailed description of the types of endogenous work practices that local users would like it to support. At this stage, we might wish to debate whether this formal organisation is the best way to manage participation, or the best way to proceed with a socio-technical design. To do this, we could seek to evaluate the system in some manner, specifically to see whether we could implicate the design approach or the form of participation in the system's success or failure. However, we take a different approach, one which acknowledges the manner in which the project is set up is eminently sensible and rational, given the contingencies of running such a complex, large-scale project. By selecting a proven (albeit in other countries) health care solution, by involving representative stakeholders in all parts of the process, by allocating activities and responsibilities to different stakeholders in different phases, etc. seems a sensible way of getting a handle on and managing such a complex project. Indeed, all projects of this size require some formal organisation, phasing, and allocation of tasks. Our purpose here, however, is not to assess this method, or approach to design. Instead, what we are interested in is how, given the intended systematic and coherent nature of the formal project organisation, the actuality of socio-technical design and participation is multifarious in character. As we shall show, design and participation vary in nature because of the practicalities of putting a formal plan into action under the influence of various local and organisational contingencies, and that this seems to influence outcome at least as much as the quality of the formal plan.

When phase 1 eventually went live a year late, there were a number of problems with the system. One of the clear difficulties was that, in particular areas of the design, the system did not fit well with the needs and practices of the users. There were various reasons for this, and we have documented in other papers how user requirements were consistently de-prioritised as 'technical' requirements, and requirements for standardisation took precedence (Martin et al. 2005; Martin 2006) in pressured and difficult-to-resolve circumstances. However, it was also clear that some of the problems in the usability of the system could be traced to problems in achieving productive and useful participation of stakeholders and in gaining the required information on the application domain. In this chapter we concentrate on understanding these issues.

7.2.2 Keeping Users in Mind

Throughout the project there is an obvious need to involve stakeholders, to keep ‘users’ in mind – though this may get submerged in a myriad of demands of keeping a project on track. In these circumstances, a focus on practice as well as process (a socio-technical perspective) and an understanding that a ‘domestication’ process (Williams et al. 2005, 2000) that fully involves the various stakeholders is required. Of course, ‘users’ come in various forms, and on occasion it may be that the interests (or convenience) of one set of users (e.g. administrative staff) may clash with those of another (e.g. clinical staff or patients). Tensions (professional and design related) exist between different user groups. Even within the clinical user group there are many different subgroups:; consultants, doctors, nurses, physiotherapists, occupational therapists, radiologists, lab staff, etc. Each has different work--oriented perspectives and this can have implications for the EPR design. Nevertheless, the EPR system, as an infrastructural backbone of the organisation, requires a close match with organisational structure, process, and practice. The system is inextricably linked to all work activities, so it is of crucial importance to understand and take users’ everyday activities into account. One problem that arises therefore is in considering the relationship of the EPR to other organisational changes, where there may be a lack of understanding of just what the implications of the EPR are on everyday organisational workings. In these circumstances, the problem emerges of human factors effectively being downgraded, being dumped (perhaps by necessity in this type of project) down the schedule, or treated as ‘other’ types of problem and are perhaps not adequately addressed.

In this project there is clearly a desire for a deep understanding of current procedures – as part of an attempt, in some cases, to transform them. In this case, as described, user involvement comes in the form of ‘expert’ users who are involved in specifying current configuration and procedure. The main involvement of these expert users comes during testing, which it is envisaged will highlight various human factors problems. Though identifying the problem and its solution can be difficult with a piecemeal documentation of current practice. The influence of these expert users is partially reliant on their relationship with their UK analyst (and that analyst’s experiences of the UK health care system). So, for example, a ‘pathology analyst’ who had worked to develop systems with users over a long period and was good at championing their cause, notes:

> …if there’s no way to get the information, from the microbiology system into (the EPR system) then people will still have to go to multiple places to get the information they want and that defeats the object (of the project)
> if we have a single sign on procedure, to get both onto the network and onto (the EPR system), we may run into problems in the laboratory with our connections to all our other analysers

In the case of the pathology team, we have observed many more discussions concerning how the system needs to be set up in order to support the manner in which personnel get their work done – a clear acknowledgement of user needs, and of the importance of current practice and process. This can be contrasted with other areas of the project where participation and socio-technical design concerns are less successfully realised. For example, the analyst in charge of the project of migrating the existing patient administrative system (PAS – for booking and recording appointments and treatment episodes) complained, across a series of meetings, of the difficulties in getting definitive information on the current processes. He could not locate the person (or persons) who could provide the information and, instead, had been given piecemeal and outdated data; and this was occurring at a stage when the new database and workflows were meant to have been completed.

Here we relate a number of reasons why the experiences of the two analysts were contrasting, and in doing so, reveal some of the reasons why participation and design vary so much. Firstly, PAS is arguably a more complex application to build, not in terms of functionality but because it covers administrative working across the whole Trust – covering thousands of workers in many different areas with diverse local processes, or diverse ways of achieving ostensibly 'the same' process. In the case of pathology, although functions may be more complex, and pathology may connect with the whole Trust, the group of users is relatively small (50 or so) and is primarily co-located. For each 'module', however, the 'core' design team was the same size.

Secondly, the make up of the teams varied considerably. The criterion for analyst selection was that they had to have previous experience in systems analysis and/or health care experience. The pathology analyst had both, and work experience with the various stakeholders in that area, that is, the stakeholders were far better at verbalising what was required and what they would not stand for. It was therefore clear to the analyst when the requests or complaints were serious. In the case of the PAS analyst, although he had worked in the hospital previously, it was in the finance department, which meant he was now working in a new area with unknown people. This led to a considerable amount of time being taken up for establishing productive relationships. One cannot ignore the political background against which such work can proceed and 'resistance' and 'unhelpfulness' by expert-users can blight relationships. Indeed, the project team roughly characterised expert-users in various ways such as 'volunteers' or 'volunteered' and part of their work had to be a matter of assessing and sorting the information provided from various stakeholders as it could conflict with other information they received. It is worth noting, however, that sometimes conflicts with 'expert' users (and their superiors) occur due to slippage. The regular business of the hospital proceeds and expert-users must be drawn away from other duties in order to help the project team. When planned 'project time' was realised, disruption usually was minimal. However, when the dates for module testing slipped at the last minute, for example, the rescheduling work that this provoked (again!) was substantial. The slippage was not really anyone's fault but it potentially damages the participatory relationship.

A clear outcome of the difficulty in achieving the desired participation in the design process is that the system envisaged as a means to enhance and transform current process and practice is shelved, and instead, design reverts to replicating the current data flow. Design becomes conservative because it has to proceed with what is available to work with. This was put succinctly by the PAS analyst:

Extract from field notes taken at analysts' meeting 17-10-03

... for scheduling of clinics we need many different users to test it as it is different for different areas. We're basing our build on the call centre (*takes the bookings etc.*) information. The problem is that the build comes from either PAS or 'how you do it'. The how you do it information has not been provided in full or in a format I can use so we will just have to go on how PAS does it. We wanted to set up clinics the way they work. It would have been magnificent. But we have to go to PAS instead. No one in this hospital is capable of providing a list of clinics…

Such comments emphasise the desire of the project team to understand not just the data and data flow (as on PAS for instance) but what people actually do and how they do it, and to try and transform practice for the better. The basics of design are reverted to reluctantly.

The situation regarding the A&E team provides another interesting example of how various contingencies conspire to determine how successful any project development is. Some use-related issues, if unproblematic, may be entirely dealt with within the respective teams. However, the analysts' meeting is often called to resolve more serious concerns, with the analysts as a 'champion' for their users. The following example demonstrates one way in which human factors enter into the project and are given serious discussion came with a debate concerning 'logging-on' procedures – in particular, issues of security and authority that take account of the particular circumstances of medical work:

A: Because if they've got to log out people will not log out of it they don't now
B: But maybe they won't have a chance because the log in time out will…
A: Well I understand that … but if it doesn't time out before someone gets their hands on the keyboard that next action is taking place under someone else's signature
B: Mm hm
A: And that's a problem
C: Mm hm it is a problem
A: And in A & E, in that chaotic, you know, environment, they will not log out
C: Well and again that is something I mean again this is one of the reasons why we've asked for the IT trainers here as well so that this is. Yesterday I met with the IT trainers and we started talking about some of the issues that we need to make sure that everyone is aware of and one of them, you know is this issue now we'll add that to the list that this is one of the key ones, making sure that people log out and understanding the implications because in a fact it's an electronic signature, and that's going to give a-a print, of where you've been on the system and if you don't log out you're allowing someone else to use that that signature
A: But it's not a training issue **
C: Mm
A: The fact is that the log out procedure will not be looked upon as important as treating a patient
C: Sure

A: And in that environment they're not going to turn round, and log out, every time they walk away from a PC, I can guarantee that
C: Yeah so … we need to look at it, I agree it's not completely a training issue I do think it is partially a training issue
A: Well I understand that, yes

In the above excerpt, A is the A&E analyst, B is the pathology analyst, and C is the project leader. Apart from illustrating the concerns about how the system will mesh with medical practice are taken seriously, and showing the setting and form which these discussions commonly take, the quote aptly illustrates the manner in which problems are conceived in such exchanges and that how to solve the issue is a problem for those involved. When human factors issues are raised with the proposed design, the project team decides whether the issue merits redesign (potentially the more costly option) or can be dealt with through other measures such as training or 'change management'. The question as to whether there is a serious issue with usability or whether it is a case of 'resistance' is a common one and can lead to disputes like the one above where the A&E analyst rebuffs the characterisation of the problem as a training issue. The 'correctness' of the characterisation may only be understood at a later date. The situation for A&E is interesting in other ways as due to the specifics of the work environment, which is intense and chaotic, it presents particular challenges for an EPR. Whereas in other areas, treatment (and therefore, administration) is more methodical and consistent, in A&E staff may be relocated from one duty, patient, etc. to another according to emerging requirements. As indicated above, this means the system interaction is regularly interrupted. Thus, concerns over privacy, logging out, and data loss become greater, and the flexibility of the system to cope with the specifics of the work was especially salient for this area.

The A&E experience also illustrates another contingency that has implications for the emerging design. Both the US and the UK analysts for this area resigned during the project. Staff losses and resignations are likely within a project of this size and duration. However, planning for these is difficult and dealing with them inevitably takes time as steps are taken to employ and familiarise new team members, especially when we consider the importance of the team-working relationship as discussed earlier. When this is considered against the tight (and tightening and slipping) schedule we can again see how the aspirations for design suffer.

7.2.3 Escalating Problems

An interesting outcome of the problems experienced in A&E during the database building and configuration stages was that the difficulties became the main agenda. For nearly 6 weeks, little work was done and the staff and managers in A&E were militant in their criticism of the system. The escalation process was used to ensure that the problems were considered at the level of the IM&T programme board, where, pressure could officially be placed on the supplier, and that serious and sustained effort was made to resolve these difficulties. The subsequent focus on the plight of A&E produced a major turnaround in that area in the following months.

Orienting to the project as a totality also necessarily includes an attention to the methodical handling of tasks, handling the project agenda (especially in meetings with technology providers), and escalating things in the correct fashion. It also includes some notion of keeping track and measuring progression, negotiating, and renegotiating responsibility, and having some awareness of the correct routes by which tasks should be accomplished. This is quite clearly seen in the issues surrounding the escalation of problems – how can a problem be raised as an issue to ensure it is addressed while maintaining otherwise cordial professional relationships? Interestingly, this material draws to our attention that user–designer relations is not simply a matter of users on one side and designers on the other, but involves a more complex criss-crossing of relationships where tensions flare within and across groups. In this case, we also have two sets of analyst-designers. For instance, the US-based designers must maintain an orientation to both OurComp and to the needs of the Trust, and the Trust analysts may be both sympathetic and frustrated by expert-users. The Trust project manager needs to be sympathetic, while achieving a means of getting things sorted out according to procedure. The following quotes from the project manager to UK analysts and an expert-user illustrate how she does this 'emotional labour', while minimising the risk of disputes and ensuring that problems get dealt with according to the correct procedure.

Within the EPR project, there is a managed process for escalating problems – a staged process:

> In some of the escalation process stuff …I try and do everything as a staged process …and I do try and keep things away as much as possible so that you're not having to get involved in the in the fight part so to speak

There are ordered 'issues' and 'risks' logs – issues become risks when they are deemed to be a threat to the planned delivery of the system:

> …it's already on the Risk, Log we uhm probably up the risk number at this stage cos' its obviously increased in possibility or likelihood

When problems cannot be readily solved between analysts they are removed from the discussions:

> I'm trying to as much as possible keep the grappling over this with OurComp at the level of me because I don't want to impair your working relationships with your analysts

The logs (particularly the risk log) are used as a means of escalating the problem to be dealt with at a higher organisational level – in this fashion attempting to ensure that harmonious working relationships can be maintained at a lower level.

> I have said I wanted the data to be issues at the risk log now because I said this delay and um the direction so um not that I want anyone to get into an argument with them during the conference call but just so you do know I have escalated this one because I am very concerned

just to reassure you tomorrow's IM & T steering group you can bet that this issue is going to come up at that because I already know and Y knows the issues around the code of connections.. once I've got the IM & T steering group fully aware of all of these issues … they'll take a stand, in a sense an official stand from the hospital perspective which will make it a little easier for me to put a put more pressure on OurComp but I do want to keep you aware of sort of how things are going and again I do try and … keep … that argument side of it away from this group cos' I don't feel you need to I don't want you to have to worry about that side of it if that makes sense because I know you have enough on your plates without sort of having to get involved in that but I'll try to keep you better informed as to where things are going with that.

As we see from the excerpts, the issues and risks logs are the means through which problems are noted and ranked and that their ranking is used to justify a set of actions aimed at getting the problems solved. As a problem persists and is moved up the rankings, it provides the authority to raise the issue at a 'senior' meeting or with the supplier. Of course, it is important to acknowledge that issues and risks are socially constituted – just what to place on the lists, at what position, and the subsequent actions are matters for discussion in the various teams involved in the project. Consequently, the attention given to particular risks depends on matters such as the vociferousness with which they are voiced, how they relate to timetabling, how agreed upon they are, how many areas of the project they affect, and how difficult they are to be solved.

Furthermore, as we can see from the quotes, both the formal apparatus of project management – procedures for problem escalation, management groups with the authority to settle disputes, etc. – and the sensitivity of the project manager, are crucial in maintaining good working relationships between various designers and users. It is a difficult balance for the project manager – showing equanimity, patience, and professionalism at all times – but tensions are bound to spill over and, in a project when problems will necessarily occur, these are important features of project management if working relationships are to be maintained.

7.2.4 Keeping Track of Issues

Getting a project to work requires that the project leader keep track of issues and problems as they arise and are prioritised and dealt with. Issues, when they do arise, are conventionally managed through formal and informal conversations aligned with the use of various forms of documentation (schedules, logs, and meeting minutes):

Project Manager: I think we just raise it so that its minute-ed that we've raised it see what their response is…

Nevertheless, items can fall off the agenda causing problems – '*I'm worried that this one has fallen through the cracks*'. Sometimes 'others' – usually the suppliers – have let the project down in some sense by not conforming to agreed deadlines.

A: "…it was identified that this should be in place by June so we thought we were merrily, things were progressing the way they should but now the last information that we received, contradicted that so-so I'm going to start ah doing some phoning today. – and see what we can do…"
A: I went back to the minutes from the initial Z Co meetings and, X had said very clearly
B: Yeah he was quite confident it be ready by June
A: And we would be the first ones installed and so from the initial reports that's why I've never got too concerned and again that was a
C: Mm
A: Fatal mistake

Key: A – project manager, B – pathology analyst, C – pathology expert user

As the above extract shows, deadlines are no guarantee that work will be done and consequently the project manager needs to maintain some overall awareness of progress – to orient to the project as a totality. And problems may return:

Project Manager: No I think that's a real concern and as I've said I have raised it earlier and have actually added it to the issues log earlier and we have got some movement then but we're still we're we had some creep back.

Discovering that something has slipped off the agenda (and that work that was assumed to be progressing has not been) is the type of nasty surprise the project team has had to get used to. Again, this has serious effects for design, often meaning that a 'satisficing' workaround needs to be devised, one in which the desired transformative aspects of the system are compromised. In terms of user–designer relations, these examples again point to the crucial role of project management in intervention in the relationships between users, Trust analysts, and designers/analysts from other companies (both OurComp, and in some of the examples legacy pathology suppliers). Keeping track is not only a matter of making sure everything is progressing as desired but also a means for the project manager to detect problems within working relationships and to intervene before these break down.

7.2.5 Domestication and Legacy Systems

As health care organisations seek to deploy the EPR as an infrastructural technology, that is, as a backbone for organisational activities, the need for a close match with the organisation increases (Hanseth and Monteiro 1998). A growing body of research has pointed to the difficulties involved in designing systems that match the complex and particular needs of organisational users. To work and be useful, such systems have to be adapted in the course of implementation and use to match them to users' technical and organisational contexts (ibid.). To be successful, such processes of configuration, 'design in use' (Hartswood et al. 2000) or 'domestication' (Williams et al. 2005, 2000) require contributions from a wide range of organisational members as well as technical specialists.

Therefore, for example, underlying technical and organisational issues, problems connected with 'legacy' systems hinder the development and deployment of the EPR. This is not only about linking software from different systems, but also about understanding how the organisation works. An appreciation of legacy needs to move away from a purely technological stance – with its emphasis on aging systems and code – to admit the importance of a subtle appreciation of factors that may appear distant from the technology, including the fine detail of everyday working practice. Any attempt to resolve legacy issues will depend on understanding that organisational change will necessarily have to confront legacies as the practical issues of daily work. This involves understanding how technologies become embedded, and are oriented to, within everyday working practice and a subtle appreciation of the practical meshing of organisational structure, processes, and technology.

The configuration challenges for the EPR are numerous and significant. There are a large number of issues concerning the detailed design of user interaction with the system:

> **Project Manager**: Yes and just to reassure you I have sent an email … and I've said very clearly that the expectation has always been that well we would participate in configuration it was on the understanding that they would be directing that configuration … and I have said that there was fifty three days where we were looking at database configuration and I've said that now there's, not to scare anyone, twenty eight days left before um twenty eight business days left before we uh are its in the plan its identified that we're going to start testing. We've not done any configuration so I have said I wanted the data to be issues at the risk log now … so you do know I have escalated this one because I am very concerned and especially when I've been expecting more clear information about what we're doing and I've just I'm worried that this one has fallen through the cracks…

Perhaps more importantly, many implications of information integration, that is, more rapid information flows, novel information representation, and record-keeping practices, will only be understood through experience. Integration may change existing – or create new – work dependencies between, for example, clinical and administrative departments in unexpected ways. This is seen, again, for example, in the debates around security policy:

> **Pathology Analyst**: … can I ask a quick question … what's gonna be the policy with regard to time out, functionality of the software?… if we have a single sign on procedure, … we may run into problems in the laboratory with our connections to all our other analysers, if somebody initiates a data transfer, … for reviewing and authorising results as they come up on analyser and the network connection is cut because the time out's kicked in … you could end up locking a lot of results that takes a long time to actually retrieve…

It is vital that system implementers be aware of such changes, evaluate their significance, and match them to options of system configuration. For this to happen, there must be effective feedback mechanisms to implementers and appropriate policies in place for negotiating how this is acted upon.

7.3 Discussion: Project Work and Organisational Issues

This concluding section attempts to link the everyday concerns of managing a major IT project with other important organisational considerations. The EPR project is characterised by ongoing negotiations about tasks and responsibilities, and substantial ongoing effort to coordinate work across a diverse interorganisational teams across sites and timezones. This is managed through both formal (contracts, schedules, meetings, and visits) and informal (email, telephone calls, etc.) means. Working with and working out these relationships between organisations:

> in a sense our thing is with, ** the ..(OurComp's) manager and theirs is (with) H-Ware
>
> …involves learning how they are structured:
>
> I've got the numbers to start phoning myself and trying to pursue it we – we're a bit in a situation where we're at the mercy of different organisations because … it's ZZ … a-and so we're trying to liaise through various layers of people to try and get this to move on, so …. I have made the IM & T steering group aware of this at the last meeting which was approximately a month ago to raise it as a concern

… and (of course!) with interorganisational working others' failings can suddenly become your problems:

> it is our issue but its … not us holding it up on this one … but it will not be seen that way in the Trust they'll see it as the EPR not meeting a target

System design in a large NHS Trust (and the associated processes of analysis, configuration, testing, integration, evolution, etc.) is a complex, messy business. Within the EPR project reported here, this Trust design was proceeding in tandem with the implementation of a new network infrastructure. In these circumstances, issues such as hardware provision, data point placement, database configuration and population, interface design, and training are inextricably linked to other projects and organisational working associated with modernisation and investment in IT. At the same time, the NHS environment can be said to be (continually!) characterised by upheaval and changing circumstances, policies, even governments. Furthermore, given national, governmental targets, and priorities, there is a sense in which this is a project that cannot afford to fail – unlike the software projects documented by Button and Sharrock (1994) and, despite the long history of IT failures within the NHS, there is a strong sense that this is a project that must succeed, that abandonment to work on another project is not an available, or a thinkable option (which can be seen by the continuation of the project in the light of major problems and long delays). Of course, resource also remains a problem within the NHS environment. Variations in resource coverage are due to histories of systems use, problems in attracting technical staff, differing systems expertise, different mechanisms for clinical input, varied relationships with clinical staff, etc. – all of which bear on the success of the project and its associated work. Similarly, reliance on many providers adds even greater complexity to working relationships. The core system, legacy applications (e.g. pathology), and middleware are all provided (or have been provided) by different companies, and the advent of PPP has changed relationships between providers such that the Trust only has an indirect relationship with legacy providers.

Some of the impediments to integration through integrated healthcare records simply reflect the scale of the organisations and services involved. For large organisations with complex information systems, achieving even modest levels of integration can be difficult in practice (Ciborra 1994). Here, the issue of funding and ensuring accurate statistics brought this concern to the fore:

> **Project Manager**: …I did meet with XXX yesterday to discuss some of the issues … because the reports we hand into the NHS are crucial to our funding, as a Trust and obviously we have to get the reporting right and there's a huge risk to the Trust because we're going live six weeks before the end of year, and … all of our end of year reports we have to make sure are right between that six week period, … I needed to speak … and make sure he understood very clearly, what these risks were … it is an issue that we need to really look at because we do need to make sure our reports are correct that we're handing in and OurComp has to build the system to NHS requirements so we do have to sort of match up all these things…

The government and public desire for transparency, league tables, and reporting mechanisms based on 'real-time' statistical gathering places a strong reporting focus on the EPR. This accentuates the need for business-focused organisational acumen to understand how to produce figures that paint the Trust in the best light within the 'rules' of production for those figures (Bittner's 'gambit of compliance' (1965)).

The evolving nature of the services leads to difficulties in providing technical support that can evolve to match organisational change. Large organisations exhibit further complexities related to scale, numbers of distinct roles and processes, and the richness and interrelatedness of information in the organisation. Information exchange practices and systems are rooted in local work processes as well as wider patterns of coordination and communication. Attempts to change practices and redefine roles and relationships may lead to resistance, if those involved have different commitments and understandings of organisational processes and service provision. In this project, while acknowledging the problems around users and user participation, we became clearly aware of the problems that can arise when users feel under-consulted. This is particularly clear when they have been lead to believe that the design is meant to support their work but they are presented with something that seems anything but this at the late stage of testing. As stated by a PAS manager:

> We don't want to sign this off before we go through everything in the proper detail… we are not fully happy about accepting that training will sort out all of these problems … some of them seem like major problems.

It is clearly not desirable to reach a stage where users feel the only way to get their concerns heard is to threaten to withhold sign-off!

Current health and social care policy initiatives in the UK make significant claims about the desirability of integrated services for better health and social care, that is, more patient-centred health care delivery, improved resource utilisation, and management of information. Plans for implementing these initiatives appear to be largely predicated on information integration being a precondition for service integration. The EPR is an element of this strategy, yet as our research too readily documents, its implementation presents formidable challenges.

7.4 Conclusions: The Practicalities of Participation and Socio-Technical Design

The obvious lesson from this study is that designing a system (even through configuration of a COTS product) for the complex setting of an NHS Trust means being thoroughly realistic about a mass of contingencies that will push and pull at the project. Design as conceived in abstract is a structured, coherent, and systematic activity. Design in the 'wild' is subject to vagaries that place a project team under constant pressure to maintain something with a semblance of structure and coherence, something that seems pretty systematic. This is not to write off design methods. Indeed, this chapter has attempted to show how the practical business of design uses the tools of the design as a critical resource for getting things done.

The tools of design (e.g. method, structured approach, and documentation) serve as a resource for planning the project structure and approach, and they are the means for working out what has been done, what needs to be done next, who should be doing what, what needs special attention, whether things are on schedule, and so forth. In this way, the practical everyday management of the project resonates with these tools. This enables the project team to measure and compare the progress of development in different areas. It allows them to understand why, in one area, things are going as planned, while in another progress and development has deviated from what was intended. While we may seek to criticise aspects of the actual approach to design, because we believe in a different form of participation or a 'better' approach to socio-technical design, the material presented in this chapter directs us to inherent problems in implementing any approach to design. We have attempted to show a number of the ways in which design and participation are so contingent to practical concerns, to inter and intraorganisational issues, politics, and so forth. The challenge to the PD community is therefore not just the provision of methods to understand the user and their practices, to facilitate participation throughout the life-cycle in design, testing, evolution, and so forth, the challenge is how to make these methods work more consistently in settings like the Trust.

At this point we have to ask which of these contingencies can be dealt with by researchers and practitioners interested in pursuing socio-technical participant design. First, it is worth reiterating the contingencies that we described as having an impact on the design aspirations of the project. We might say that lack of resource has an impact on the ability to realise the goals of the project. This may be true to a certain extent but it is well known that simply investing extra money, people, time, and so forth into a project is no guarantee of swift success (Brooks 1995).

When we focus more specifically we can see that the contingencies fall into various groupings. Participation and socio-technical approaches to design vary because of contingencies that stem from: (1) differences in analysts and stakeholders and their relationships, (2) differences in complexity encountered in different areas (or modules) of development, (3) competition between participation and socio-technical concerns and other design concerns, and (4) organisational, interorganisational and regulatory (Government) pressures. Stakeholders and users are diverse groups, sometimes willing and other times not, with potentially competing requirements. Analysts vary in their experience of their given area, and over time, working relationships with users have been established. These differences mean that the ability of teams to follow through the intended plan differs markedly. During the course of the project it emerged that certain areas of development offered specific complexities, but it was difficult to re-allocate already scant resources.

User involvement and the desire to understand current practice and process may be set out as a cornerstone of a design but in reality it not only has to both compete with a raft of other design considerations but also has to fit in with their production and a schedule of completion. Sometimes, user considerations may simply be lost in a morass of troubles, at other times they may be shunted down the list as less salient or important at that point in time. Aspirations may simply be dropped as schedules of production mean a solution must be delivered now. This means that design reverts to what is more easy and basic rather than being socio-technical or transformative. Finally, the organisational backdrop, with multiple (sometimes competing) user groups, all other organisational business, the need to work with other organisations (e.g. in a network of providers) and within a regulatory framework also puts various pressures on the project.

No doubt, better methods, tools, different approaches to design and a sensitivity to these contingencies, and the impacts they have may help in assuring a more systematic participant socio-technical approach to design. However, we can also see that enabling design as we would like it to proceed requires an organisational will and a regulatory framework that will allow it – and these may be outside the designer's remit. In the case reported here, the Trust is only realising that the EPR has ramifications and interactions across the whole working of the organisation – and that understanding and transforming the workings of the organisation is as important (if not more) than any technical solution and must be done in concert with any development. Finally, what also needs to be acknowledged is the regulatory requirement – that funding is tied to performance measures – complicates the requirement for the EPR. The collection of statistics and the compilation of reports become a key focus for design, yet another contingency, and one which clearly may compete with user and socio-technical concerns in a project with many pressures already bearing on it.

In light of this situation, as researchers with many years of experience in employing ethnography for the purposes of design, we have been thinking seriously about how we might adapt our methods for the most fruitful impact in such a setting. It is apparent to us that conducting a 'comprehensive' ethnography of a hospital not only presents methodological and logistical difficulties for an approach most often suited to smaller-scale settings but that cost for benefit would also make this prohibitive. Instead, we have developed a notion of how small-scale ethnographies might be usefully 'timed' and 'targeted' based on our understanding of the relative distribution of 'success' and 'failure' of participation (and the project) in different areas. It is clear that certain areas of the project (e.g. PAS and A&E) were causing problems early in the design while others were progressing well (e.g. radiology). Much of their difficulties resided in the fact that, because of problems in participation, the project team had a lack of knowledge about the work of users in that area. This made it difficult for them to design for those activities or to work out whether any design requirements from elsewhere (e.g. the NHS) would have a negative impact on those practices. Consequently, problems only really came to light, too late, during testing and after the 'go-live'. It would have been relatively straightforward (and cheap) to commission small-scale ethnographies to focus on these areas and issues much earlier in the design, hopefully aiding in sorting out these issues at a less-problematic stage. We intend to test out this idea in future projects.

Acknowledgements

Thanks to all the staff at the Trust who assisted with this work. This work was funded by the EPSRC Chameleon project (Grant no. CSD 7680).

References

Berg, M. (1997). Rationalizing Medical Work: Decision Support Techniques and Medical Practices. Boston: MIT Press.

Bittner, E. (1965). The concept of organisation. *Social Research*, 23, 239–255.

Bjerknes, G. and Bratteteig, T. (1995). User participation and democracy: A discussion of Scandinavian research on system development. *Scandinavian Journal of Information Systems*, 7(1), 73–98.

Blomberg, J. and Kensing, F. (Eds.). (1998). Special issue on participatory design. *CSCW Journal*, 7(3–4).

Blythin, S., Hughes, J., Kristoffersen, S. and Mark Rouncefield (1997). Recognising 'success' and 'failure': Towards the 'illuminative' evaluation of groupware. In *Proceedings of Group'97*, Phoenix, USA pp. 167–185.

Brooks, F. (1995). The Mythical Man Month: Essays on Software Engineering, Anniversary Edn. Addison-Wesley: Boston.

Button, G and Sharrock, W. (1994). Occasioned practices in the work of software engineers. In: M. Jirotka and J. Goguen (Eds.), *Requirements Engineering Social & Technical Issues*. London: Academic Press.

Carroll, J., Chin, G., Rossen, M. and Neale, D. (2000). The development of cooperation: Five years of participatory design in the virtual school. In *Proceedings of DIS 00*. August 17–19th New York, USA.

Ciborra, C. (1994). The grassroots of IT and strategy. In: C. Ciborra and T. Jelassi (Eds.), *Strategic Information Systems*. Chichester: Wiley.

Clarke, K., Hughes, J., Martin, D., Rouncefield, M., Hartswood, M., Procter, R., Slack, R. and Voss, A. (2003). Dependable red hot action. *ECSCW '03* (September). Helsinki: Kluwer.

Crabtree, A. (1998). Ethnography in participatory design. In *Proceedings of the 1998 Participatory design Conference*, Seattle, Washington, pp. 93–105.

Ehn, P. (1988). *Work Oriented Design of Computer Artefacts*. Stockholm, Sweden: Arbetslivcentrum.

Ellingsen, G. and Monterio, E. (2000). A patchwork planet: The heterogeneity of electronic patient record systems in hospitals. In *Proc. IRIS'2000* (August). Uddevalla, Sweden.

Geertz, C. (1973). 'Thick description: Toward an interpretive theory of culture' in The Interpretation of Culture. New York: Basic Books, pp. 3–30.

Grudin, J. (1990). The computer reaches out: The historical continuity of the interface. In *Proceedings of CHI 90,* Seattle, US pp. 261–268.

Hanseth, O. and Monteiro, E. (1998). Changing irreversible networks. In *Proc. ECIS* (Aix-en-Provence, June).

Hartswood, M., Procter, R., Rouncefield, M. and Sharpe, M. (2000) Being There and doing IT in the workplace: A case study of a co-development approach in healthcare. *In Proceedings of the 2000 Participatory Design Conference* (December), New York, USA.

Hartswood, M., Procter, R., Rouncefield, M. and Slack, M. (2003). Making a case in medical work: Implications for the electronic medical record. *Journal of Computer Supported Cooperative Work*, 12(3) 241–266.

Hartswood, M., Procter, R., Slack, R., Voß, A., Buscher, M., Rouncefield, M. and Rouchy, P. (2002). Co-realisation: Towards a principled synthesis of ethnomethodology and participatory design. *Scandinavian Journal of Information Systems*, 14(2), 9–30.

Hartswood, M., Procter, R., Rouncefield, M., Slack, R., Voss, A., Büscher, M. and Rouchy, P. (2007). Co-realisation: Evolving IT artefacts by design. In: M. Ackerman, C. Halverson, T. Erickson and W. Kellogg (Eds.), *Resources, Co-Evolution and Artefacts.* Berlin: Springer, pp. 59–94.

Heath, C. and Luff, P. (1996). Documents and professional practice: 'Bad' organizational reasons for 'good' clinical records. In *Proceedings of CSCW 96,* Boston, US, pp. 354–363.

Hughes, J., Randall, D. and Shapiro, D. (1992). Faltering from ethnography to design. *Proceedings of ACM CSCW '92, Conference on Computer-Supported Cooperative Work*, Toronto, Canada pp. 115–122.

Martin, D. and Rouncefield, M. (2003). Making the organisation come alive: Talking through and about the technology in remote banking. *Human-Computer Interaction*, 18(1 & 2), 111–148.

Martin, D., Rouncefield, M., O'Neill, J., Hartswood, M. and Randall, D. (2005). Timing in the art of integration: 'That's how the bastille got stormed'. In *Proceedings of ACM Group '05* (November 6–9), Florida, USA, pp. 313–322.

Martin, D. (2006). Who and what are electronic patient records for? An ethnomethodological ethnography of system deployment in the NHS. In *Proceedings of Symposium on Current Development in Ethnographic Research in the Social and Management Sciences* (13–14th September 2006), Liverpool University, Liverpool, UK.

Williams, R., Slack, R. and Stewart, J. (2000). Social learning in multimedia, final report to european commission. *DGXII TSER*, University of Edinburgh, Edinburgh.

Williams, R., Stewart, J. and Slack, R. (2005). Social Learning in Technological Innovation: Experimenting with Information and Communication Technologies. Edward Elgar.

Chapter 8

Bottom-up, Top-down? Connecting Software Architecture Design with Use

Monika Büscher[1], Michael Christensen[2], Klaus Marius Hansen[2], Preben Mogensen[2], and Dan Shapiro[1]

[1] Department of Sociology, Lancaster University, UK, m.buscher|d.shapiro@lancaster.ac.uk
[2] Computer Science, Aarhus University, Denmark, toby|klaus.m.hansen|preben@daimi.au.dk

8.1 Introduction

Participatory design (PD) has traditionally focused on the design of end-user applications or the co-realisation of a more holistic socio-technical bricolage of new and existing technologies and practices. 'Infrastructural' design issues like software architectures, programming languages, communication, security, and resource models do not seem to be in need of, nor amenable to, PD. Yet we should expect, and research has indeed shown, that there are deeply consequential relationships between use and software architectural design. If designers hide the 'sensing', 'reasoning', and computation technologies do, for example, people can find it difficult to perceive, understand, and creatively exploit technological affordances (e.g. Belotti and Edwards, 2001). In addition, the causes of failure and breakdowns can be hard to detect and even harder to address (Belotti et al., 2002).

Moreover, the emergence of ubiquitous, ambient, and component-based computing has taken computing out of comprehensive systems into a multitude of devices, services, and resources. In some sense this makes the computer disappear or become invisible (Weiser, 1991), and it enables increased flexibility and 'bricolage' of disparate elements, but it also introduces extra difficulties, for example, when it comes to determining which computing devices, services, or resources are the most appropriate to use in specific situations. To engage ubiquitous computing technologies effectively and creatively, people need support in making computational processes, states, and potential perceivable or 'palpable' as and when they may need or wish to do so, and in ways that are appropriate for the particular situation they are in, their level of computational literacy, and their interests. New software architectures are needed to support palpability. To address this, various European research teams have come together in the 'Palpable Computing' (PalCom) project (www.ist-palcom.org). The project is creating a range of palpable, ambient computing prototypes in health care,

A. Voss et al. (eds.), *Configuring User-Designer Relations*,
DOI: 10.1007/978-1-84628-925-5_8,

emergency services, and landscape architecture. Its principal aim, however, is to create an open architecture for palpable computing. The open architecture will consist of a set of specifications as well as a reference implementation of these specifications.

This is an ambitious goal. The demand for appropriateness and the complex, multilayered translations between material computational processes and the functionality and interfaces that the users experience mean that it is a goal that is impossible to meet completely. However, some significant progress can and must be made if ubiquitous computing is to be an attractive and useful prospect. Clearly, design for palpability is not simply a matter of revealing what was previously hidden. Palpable computing is a new design initiative that envisages ubiquitous technologies whose states, processes, and affordances can be made available to the senses, or 'palpable', and that are therefore more easily understood, appropriated, and controlled. To address palpability, we take six dimensions of the vision of ambient and ubiquitous computing, and challenge them by considering their opposites. Users will often need to find a position that lies between the extremes:

ubiquitous/ambient computing	complemented with	palpable computing
invisibility		visibility
construction		deconstruction
scalability		understandability
heterogeneity		coherence
change		stability
automation		user control and deference

When a supposedly seamless and transparent set of connections breaks down, for example, users should be able to make them visible and inspect what has gone wrong. Similarly, users should be able to deconstruct an ambient assembly of devices and services, both to inspect it for repair and to use its elements for new assemblies. While ambient environments should be able to scale up to large numbers of participating elements, they should also remain understandable. Coherence must be forged from heterogeneous materials, such as disparate digital and physical devices and information, while recognising and where necessary preserving the particularities of each. Changes – for example, of location, resources, context, and activity – are normal in an ambient environment, but sometimes users need to be aware of the change and sometimes they need to experience highly stable adaptivity. Users do not want to be constantly pestered with choices and they need to be able to delegate 'routine' decisions, but it is inevitable that the system will often guess wrong, so users must always be able to retrieve control – and must have the information to help them to know when they might want to do so.

To support the situated negotiation of these core dimensions of ubiquitous computing and to allow people to make computational 'sensing', 'reasoning', potential and actual activities palpable, new forms and depths of interactivity are required. This is where the ambition and, perhaps, inescapable unattainability of the ultimate design goal of palpable computing lies. Some form of human-like social and contextual perception and skill on the part of the technologies seems to be essential; yet, research within computer-supported cooperative work (CSCW) and related fields proves that it is impossible to produce anything but very limited and flawed versions of such perception and skill (Suchman, 1987; Dreyfus, 1992). When people interact with each other, they are able to negotiate contradictions and complementarities with ease, using nuanced skill, perception, and judgment to act appropriately as the situation demands. They are able, that is, to act with social and contextual skill. For computers, however, this is extraordinarily difficult. Palpable applications and services 'must' be able to make, and support the making of, optimal choices concerning each of the dimensions outlined above in situated use, and a palpable software architecture 'must' support the construction and operation of applications and services that can do so. But we know in advance that it will not be possible to achieve this completely, and that various compromises and simplifications will have to be made.

Many designers of computer applications, spanning groupware and CSCW systems (Dourish, 2003; Bansler and Havn, 2006), ubiquitous computing (Chalmers, 2003), context-aware and ambient systems (Belotti et al., 2002), autonomic systems (Anderson et al., 2003), grid technology (Hartswood et al., 2006), and system security (de Paula et al., 2005) share similar concerns. Component-based computing potentially makes creative (de)composition possible and, more explicitly than any socio-technical step before, turns users into designers. It dissolves the privileged position of the designer who knows 'the system's range of actions in advance' (Dourish, 2003). Research and design have begun to address these challenges with flexible architectures that support tailoring (MacLean et al. 1990), with maps and models that reflect, and allow users to modify, system behaviour (Dourish, 1995), and ways of revealing system properties through 'seamful design' (Chalmers and Galani, 2004). The PalCom research builds on this work. In particular, we seek to move beyond the appreciation that it is impossible for designers to predict what kinds of translations of computational states or processes would be appropriate for different users in different situations. While developing reflective, agent, and component-based support for palpability (Rimassa et al., 2005, Ingstrup and Hansen, 2005), PalCom also supports strategies that rely less on the skill of designers to anticipate how and when someone (whose level of computational literacy and situated needs for inspection are unknown) might wish to examine computational processes, and more on support for 'reflexive' – in the sense of direct, two-way, feedback-rich – forms of human-computing interaction. Our design incorporates the evolution of standards (Belotti et al., 2002) and a variety of discovery and inspection tools.

To pursue these software architectural design goals in a way that fits design into emerging practice, an ethnographically informed, PD approach is essential. However, stretching the iterative cycles of PD (see Fig. 8.1) to involve users in the design of software architectures poses a number of difficulties. First in line is the indirectness of users' experience of computer architectures.

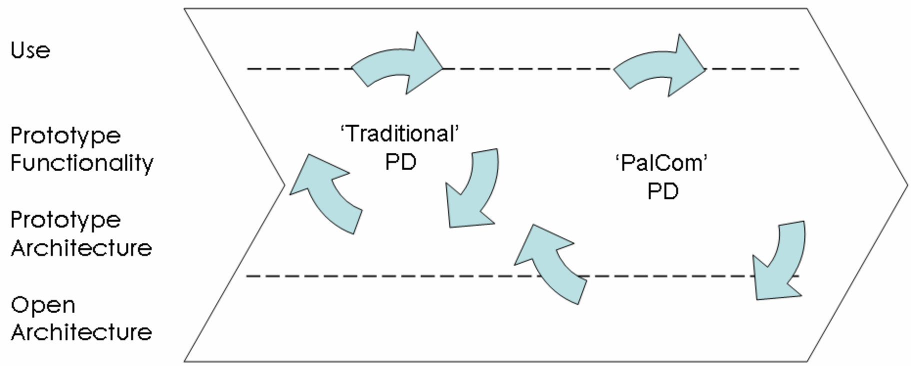

Fig. 8.1. Stretching 'traditional' participatory design methods to inform software architecture innovation

In 'traditional' PD, user participation usually informs the design of hardware and software that seek to support the users' work directly. Users are able to engage with mock-ups and prototypes of the objects they are co-designing directly, often in a hands-on manner. Where software architecture is concerned, this engagement is indirect. Even though users of palpable applications and services will rely on features of the software architecture to make computational states, processes, and affordances palpable, they will rarely interact directly with it. In their pioneering exploration of challenges for user-centred design and evaluation of infrastructure, Edwards et al. focus on the indirectness of users' experience with computer architectures and raise important questions (Edwards et al., 2003).

- Is it possible to more directly couple design of infrastructure features to the design of application features?
- How can this more direct coupling exist when the applications the infrastructure supports do not yet exist or cannot be built without the infrastructure itself?
- Could the context of either users or the use of these unknown applications have an important impact on the features we choose?
- How can we avoid building a bloated system incorporating every conceivable feature, while ensuring we have a system that will not be constantly updated (and so repeatedly broken) throughout its lifespan? (Edwards et al., 2003)

Our experience with PD shows that in-depth, long-term engagement with the context of users and use is essential for good design. We involve users deeply and equally as co-designers in long-term processes of socio-technical co-innovation. This is motivated by the fact that long-term use (and design-in-use) of prototypes that are as realistic as possible, in settings that are as realistic as possible, allows users to bring hands-on practical creativity to the use of new technologies. This is a condition for the emergence of viable future practices which, in turn, should inform the design of the technologies under development. Thus, to bring PD to the design of software architectures, we must also ask:

- How can we make use experience of software architectures more direct?

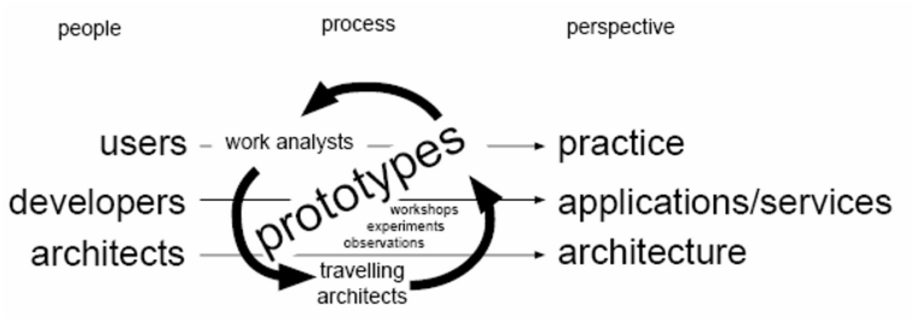

Fig. 8.2. Schema of the participatory design process

In this chapter we describe how we bring PD to the design of the PalCom open architecture. The schema in Fig. 8.2 gives an overview. Four sets of people with different primary interests and skills (users, work analysts, application developers, and software architects) connect through observations, participatory workshops, and experiments. Collaboration is often face to face and hands on, as users, work analysts, developers, and architects travel to each others' sites of work, to bring prototypes into real-world use. In the context of participatory and ethnographically informed design, there is nothing new in users, work analysts, and application developers working closely together to inform and challenge the evolving design. Bringing software architects into this process is less usual, however, and a technique we have created to support this is the formation of a group of 'travelling architects' (Corry et al., 2006). Prototypes embody, and serve as the focus for, innovation in terms of practice, applications and services, and architecture.

This method introduces at least four participatory elements to the design of the open architecture. First, the analysis of work practice and of corresponding possibilities for technical support suggests requirements for an underlying software architecture. Second, practitioners' experiences of using evolving application prototypes expose strengths and weaknesses in the software architecture design, and suggest further requirements. Third, the presence of travelling architects – gaining first-hand experience of users' work settings and of their encounters with prototypes – opens new direct pathways to the software architecture and empowers software architects to participate in a wider range of discussions and negotiations around the design. Fourth, the application developers within the project are themselves users of the evolving open architecture, opening up an opportunity for a participatory design relation amongst the computer scientists in the project.

Sections 8.2 and 8.3 below explore the intersections between these four elements by focussing on a central example, tracking the evolution of the concept of 'assemblies' through a series of reflections from different perspectives, revealing how perspectives and experiences from use, application prototype design, and software architecture design intertwine in the participatory design of the PalCom open architecture. Section 8.2 explores how the concept of 'assemblies' arose in the course of close collaboration with one set of prospective users of an application prototype. It formulates some core technical challenges, describes scenarios derived from work practice of the prototypes in use, and considers some implications for the open architecture. Section 8.3 explores the ways in which the concept of 'assembly' was taken up in the open architecture itself. In Section 8.4, we draw out some key insights from this reflective process.

8.2 Challenges to Assembling Infrastructure, Applications, and Services

Ubiquitous computing has always posed technical challenges for software architectures (Weiser, 1993). This stems, in part, from a complex interplay of requirements from particular applications and particular use and, in part, from general properties of these kinds of computing systems such as resource constraints, use of wireless connectivity, and mobility of devices and users. Considered from a technical point of view, many of the six dimensions of the challenge for palpable computing (invisibility/visibility, scalability/understandability, construction/deconstruction, heterogeneity/coherence, change/stability, automation/user control and deference) are amenable to established object-oriented software engineering practices. In this chapter, we will focus on the negotiation of visibility/invisibility and construction/deconstruction in an effort to achieve a creative understanding of computational affordances on a small and large scale, although we also touch on the other dimensions. *Invisibility* of the internals of objects, for example, is usually supported by information hiding and considered a major technique in managing dependencies in software systems (Parnas, 1972). *Construction* (or *composition*) is the *raison d'être* of component-based architectures in which applications ideally may be composed from available software components (Szyperski, 1998). *Understandability* may be said to be coupled to (static) typing in programming languages where program elements are statically assigned a set of permissible data values.

On the other hand, some of the complementary concepts in the challenge pairs give rise to interesting issues in languages, middleware, and software architecture. *Visibility*, for example, may be in conflict with information hiding (Ørbæk, 2005), in that controlled ways of 'opening up' software systems are needed. In particular, if exceptions arise in the use of palpable computing systems, visibility of what has gone wrong and possibly why becomes important. Actually, in a dynamic pervasive computing world, failure cannot really be seen as exceptional and thus we instead try to design for contingency handling (a concept covering more than just failure handling) rather than exception handling. *Change* of, for example, location is also a challenge in that references to resources from software components need to be re-established. *Deconstruction/decomposition*, in particular when the deconstruction is not an exact inverse of a previous construction, emerges as a major and radical new issue. In general, it may be said that much effort has been expended in middleware development in order to make application programming as transparent as possible to location/distribution, time, failures, etc., whereas it was realised at the outset that palpable computing would have to go beyond this in addressing the challenge pairs, for example, in terms of having to support visibility of components (and their locations) in order to support decomposition. Indeed, it has quite often been remarked that 'transparent' in computer science – meaning concealed and invisible – is quite contrary to its everyday use where it means open and accountable. One example is that of distributed systems where 'distribution transparency' means exactly that it is not known to components of the system that they reside on different hosts (cf., e.g. Stroud, 1992).

8.2.1 Gaining a Sense of How Assembly Could Be Achieved in a World Where Applications/Services Do not yet Exist

These technical issues have given input to ethnographic work as well as PD in PalCom. It should be noted that although the sequence here places technical constraints and opportunities first, it does not imply a cause/effect relationship from technical issues to fieldwork or design. The analysis of technical issues is deeply and continuously inspired by observations of existing practice and develops opportunities and problems for ubiquitous computing systems in the light of such observations.

One suite of application prototypes we are developing as drivers for software architectural design seeks to support landscape architects in landscape and visual impact assessment (LVIA), for example, of wind farms. The major difficulties in this work lie in evaluating the impact of planned but not-yet-existing developments on views and experiences of landscape (Büscher, 2006). This involves identifying and finding key viewpoints, and carrying out and documenting a complex and rigorous process of evaluation (Fig. 8.3).

In large study areas of undulating hills (60–80 km^2), it can be extremely difficult to keep track of the location of a proposed (but not yet physically present) wind farm and envisage its visual and experiential impact on people's experience of the landscape. The 'sitepack' prototype is designed to support landscape and visual assessment. It allows users to assemble photo and video camera(s), location devices, displays, the car, and other components, including computation services that convert location signals or track specific locations.

To illustrate some key ideas and challenges around which our PD process revolves, and to give readers a sense of some concrete prototypes, we present a set of brief envisioning scenarios (Fig. 8.4–8.10). They take activities observed in real-work practice within a typical working day for a landscape architect, imagined in the context of new support tools. The scenarios, although quite challenging, simplify the reality of work practice, in that they assume only one landscape architect is on site. In reality, there will often be two in the car and sometimes there will be more than one car.

Fig. 8.3. Landscape architect Lynda trying to see whether a proposed wind farm would be visible from touristically or otherwise significant viewpoints or when passing: While driving, with maps, computer models, and GPS

Fig. 8.4. Sketches and mock-ups of the SiteTracker from participatory design workshops with the landscape architects

The scenarios describe how a 'SiteTracker' service is set running on a display device and how Lynda brings the physical assembly to the car. She mounts a GPS in the front window, a display on the dashboard, and a video camera and compass on top of the motor that, in turn, is mounted on the roof of the car inside a protective casing (Fig. 8. 5).

Using the SiteTracker When Driving

While driving, the GPS constantly provides location information, and the digital compass provides directional information of where the video camera is pointing (with faster updates than the GPS). On this basis, the SiteTracker service turns the motor, and thereby the camera, to point towards the proposed wind farm, and the resulting video footage from the camera is shown on the display with an overlay showing exactly where the centre of the wind farm would be, seen from the position of the camera on the roof (Fig. 8.6).

While driving, Lynda passes a number of places that will need to be documented later on when the weather improves (documenting a viewpoint usually requires clear, sunny weather to ensure satisfactory visibility). To help her remember viewpoints where pictures should eventually be taken, Lynda frequently stops the car, gets out a still camera, unclips the GPS from the dashboard (disassembling the SiteTracker) and reassembles the GPS with the still camera to form a 'GeoTagger' (Fig. 8.7), providing a light-weight solution for bringing out into the field. When taking pictures, the GPS coordinates and a rough direction from the GPS is stored alongside the pictures on the camera.

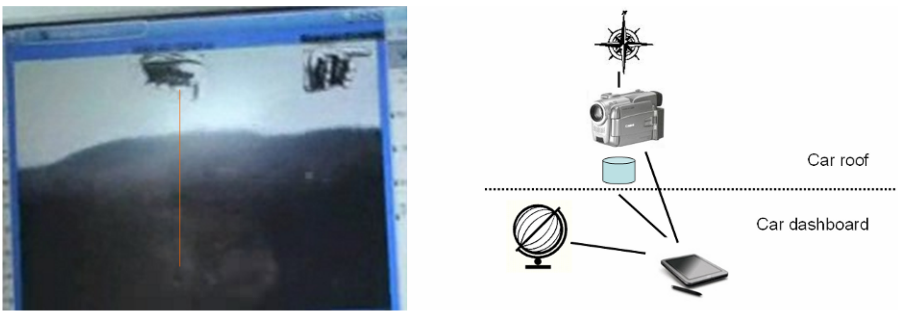

Fig. 8.5. First SiteTracker prototype and its components. The hands and the line track the location of the centre of the wind farm and other important landmarks

Fig. 8.6. Current SiteTracker prototype

Documenting the Site

The sky clears and Lynda passes an important viewpoint. She decides to document the view by taking high-quality panoramic pictures (at least 180°) for photomontages for the official report. She unclips the SiteTracker assembly from the car roof, replaces the video camera with a high-resolution still camera and mounts the new GeoTagger assembly on a tripod. The tripod provides tilt information. Using the GeoTagger service she can now take panoramic pictures, where locations, accurate directions, and tilt of the camera are stored along with the pictures (Fig. 8.8). On return to the car, she disassembles the GeoTagger and reassembles the SiteTracker to continue the survey.

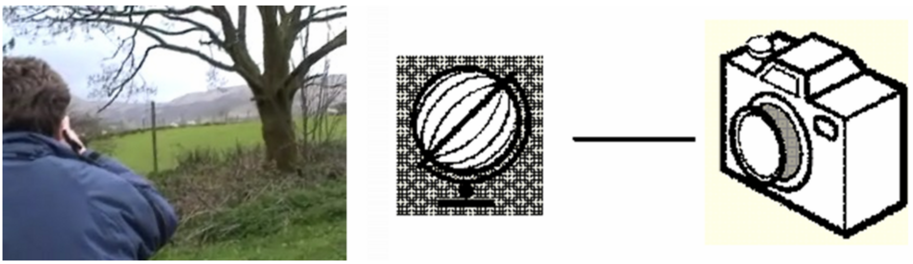

Fig. 8.7. The ‘GeoTagger’ indexes photographs with location and direction information

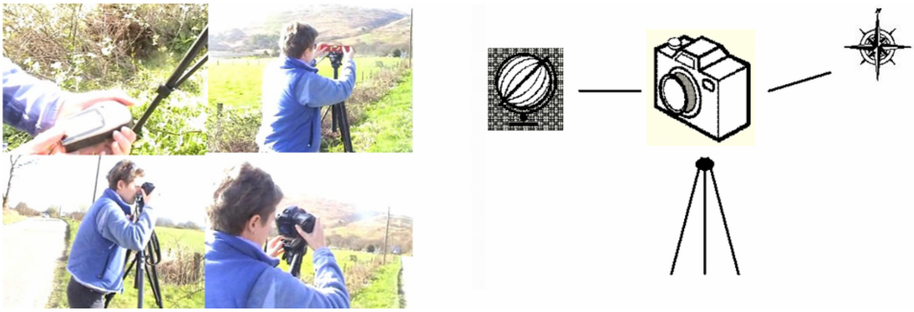

Fig. 8.8. The GeoTagger expanded with tripod

Visiting a Landowner

Later the same day Lynda visits one of the landowners possessing knowledge about the local usages of the terrain, wildlife, biodiversity, etc. She dismounts the display from the dashboard, stores some pictures, maps, and video footage on the display's storage media and walks into the landowner's offices (Fig. 8.9).

In order to present draft layouts and findings, the display is now made part of new assemblies (via the landowner's network), utilising local devices: for example, by accessing material via the small display device, but showing and navigating through it using a large screen available in the office. Changes, annotations, etc. are stored on the display device.

Fig. 8.9. Assembly at the landowner's office

When Lynda leaves the premises, all material that was shown on external devices (unless explicitly agreed otherwise) is taken back with the help of a 'take back service', so that no potentially confidential material is left on external devices.

On a second survey, it turns out that changes have happened since the last visit. Firstly, parts of industrial forestry have been felled; so much more of the wind farm will be visible from an important viewpoint. Secondly, now trees and hedges have leaves (the first visit was during winter), meaning that the hedge along the roadside can no longer be seen through.

As a consequence, Lynda has to extend the tripod so that the GeoTagger is situated some 2.5 m above ground, making it impossible to look through the camera and operate it. Therefore, Lynda extends the GeoTagger assembly with a mobile phone (Fig. 8.10). The image recorded by the still camera is now shown on the phone's display, and the phone's keypad is used as a remote controller to turn on the camera and take pictures.

Remote Collaboration

The visibility of the proposed wind farm is now more problematic than previously envisaged, and Lynda starts to wonder whether this may have an influence on the overall layout of the turbines and viewpoints. With the GeoTagger still mounted on the tripod, Lynda now uses the mobile phone to create a data connection back to her home office in order to transfer pictures of the new visibility to show and discuss with colleagues. As the discussion unfolds, the colleague back home is able to remotely control the devices on the tripod (e.g. turning the camera and seeing the result).

Below, we summarise how the prototypes described here probe the PalCom open architecture. This corresponds to the first of the four participatory elements of the design of the open architecture introduced earlier, how the analysis of work practice and of corresponding possibilities for technical support suggest requirements for an underlying software architecture.

Fig. 8.10. Site Tracker prototype and Site Tracker assembly expanded with remote control (mobile phone). Snapshot from a participatory field experiment

8.2.2 Challenges

Continuous (Re)assembling

All scenarios involve a continuous (re)assembly and *(re)construction* of services and devices. This is richly supported back in the office with appropriate prototype interfaces to make and show the device and service assembly. However, the work also calls for various disassemblies and reassemblies in the field with more impoverished resources, which must nonetheless optimise both making, and representing to the user (making *visible*) the assemblies that are in play. It needs always to be clear to what assembly (if any) a particular device currently belongs, what assemblies are in play, on what device a particular service is running, etc.

On-the-Move

In all the scenarios, the assemblies in question will be in motion. This means that even if an assembly remains constant itself, its context changes frequently. An assembly must react appropriately to resulting *changes* – by, for example, notifying users if potentially useful additional resources become available, such as the processing power in devices in a car that has come in radio range or by switching communication channels when one drops out. This calls for appropriate choices and behaviour, and appropriate documentation of such choices and behaviours, on the part of the assembly.

Shifting Modes of Cooperation

The scenarios entail shifts in the actors in collaboration as well as in the modes of collaboration. This may require a change in the behaviour of an assembly, even though neither its constituents nor its physical environment has changed. It may, for example, raise challenges regarding privacy and confidentiality of actions and materials as well as challenges in relation to who operates what assemblies, support for collaborative work, the question of whether users are part of the assemblies, and how to make those relations *visible* and *understandable*. The 'character' of an assembly depends on such intangibles as the people involved and their purposes.

Quality of Service

Different assemblies may be able to do the same things, but with different capacities, for example, with different levels of accuracy. A high degree of accuracy is not required in all the scenarios, but it is very important in all of them that the user knows and is made aware of the given accuracy. If a landscape architect is taking photographs in poor weather, for general work planning purposes rather than as photographs for official records, she may decide that relatively inaccurate direction information, derived from GPS alone, is adequate. But she should not be misled either in the present or when reviewing materials at a later date that just because a compass direction is given it has the accuracy of a digital compass reading.

It may be appropriate to operate with an implicit assembly with regard to accuracy, where the assemblies 'choose' among several potential services offering location information, depending on which one is most accurate at the moment (this changes as one moves), but paying attention always to represent the accuracy available in the current state.

Unanticipated Use

In the scenarios above, we have anticipated a number of assemblies coming into effect during a rather short period of time. What is also expected from this family of situations is that it will produce a set of unanticipated and unpredictable usages of the existing services and devices, thereby providing for unanticipated or emergent (serendipitous) assemblies and contexts. This in turn informs architectural discussions about whether 'types' may emerge during runtime or will be known at design time, whether it is just a matter of naming a particular assembly for one's own later reuse, or whether it is a matter of sharing a new type of assembly among colleagues, etc.

8.3 Assemblies

In Section 8.2, 'assemblies' were discussed as a concept arising in and from practice and prototype design, and some consequent challenges for the open architecture were considered. But is the concept of assembly itself also relevant for the open architecture and, if so, how? In this section we use the development of the concept of palpable assemblies as a representative illustration of the ways in which the competencies of ethnographers, users, software developers, and software architects interact as part of co-design. In doing so, we analyse four instances of how the assembly concept has evolved, each explored from three different perspectives: software architecture, application development, and use. In Section 8.3.1, we describe how a basic notion of assemblies was introduced to the open architecture. This prompted reflections on the relationship of assemblies to the more conventional software architecture concept of service composition, discussed in Section 8.3.2. The challenges thrown up by this highlighted the issue of assemblies as resource composition, discussed in Section 8.3.3. Lastly, the developing centrality of assemblies foregrounded the need for means to browse and inspect them – to make them palpable – discussed in Section 8.3.4. At all of these stages, there was a consequential interplay between the perspectives of end-users and work analysts, application developers, and software architects.

8.3.1 Basic Assemblies

Landscape architects – like many professionals – routinely put together 'assemblies' of devices (the car, cameras, tripods, GPS, compass, maps, etc.) for particular jobs. To leverage some of the potential of computing technologies into this practice and to drive architectural design, users, and work analysts (in collaboration with application prototype developers and software architects) began to talk about engagement with assemblies, components, and devices.

Software Architecture Perspective

The concept of an assembly was embraced by the architects, and attempts were made at translating this concept directly into software architecture. A decision was taken to make the assembly a first-class object of the software architecture. A ‘first-class object’ in this context is a construction that users of the software architecture (e.g. application developers) may directly use to construct their programmes, for example, through a set of specific classes in an object-oriented framework. The assembly was designed as a ‘service’ that had the responsibility of coordinating other services. In the context of the architecture, a ‘service’ is functionality (running in a process) that announces itself on the network and that can be accessed through messages to another process. In the scenarios above, examples of services by this definition are the GeoTagger, the displays, the ‘take back service’, etc. Furthermore, the assembly had responsibility for monitoring the state of the assembly in terms of the availability of the constituent services.

In summary, Fig. 8.11 shows the central concepts of the first basic palpable computing architecture. An assembly is here seen as a set of cooperating services which are each runtime components that in addition to being able to run on a device also provide the service capabilities outlined above. This design may be seen as a rather direct translation of the use-oriented concept into architectural concepts where the assembled parts are considered to be units of communication and functionality, or services, in a distributed system.

Application Developers’ Perspective

Landscape architects’ work on site is only one of several application domains explored with the aim of informing software architectural design in the PalCom project. The challenge for the prototype work is not to design ‘perfect’ special purpose prototypes in support of work in each application domain, but rather, to support the dynamic configuration and reconfiguration of a set of interacting devices into assemblies supporting a wide range of different usages, and thereby to challenge and inform the design of the software architecture. This means that the participatory design of the application prototypes themselves and concerns with their usability are a second-order priority. A delicate balance has to be struck to develop realistic and functional-enough application prototypes that allow users to appropriate and shape a socio-technical future where palpable computing is available, but that do not ‘waste’ valuable resources needed for the exploration of architectural design requirements. Prototypes may remain ‘sketchy’, complex, and fragile for longer than one would otherwise accept. They maybe ‘wrapped’, that is, run on a laptop simulating, for example, a mobile phone, rather than instantiated inside an actual mobile phone, and consist of more parts and actions than is obvious to the user.

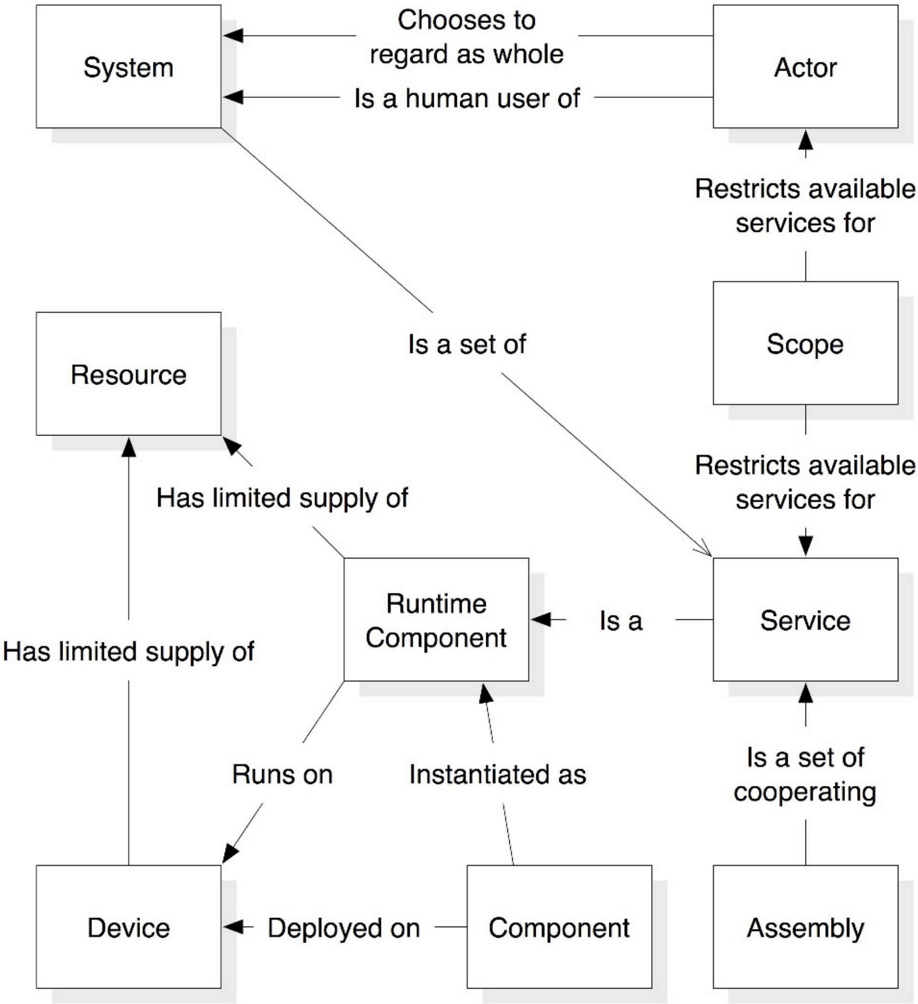

Fig. 8.11. The central architectural concepts and their relations taken from the first complete version of the open architecture for palpable computing (from first project internal architecture overview deliverable in 2004). The boxes illustrate concepts and the arrows define qualified relationships between concepts. The concepts related to assemblies are highlighted

In the first iteration of the GeoTagger the assembly consists of a digital still camera, a GPS, a display device (e.g. laptop or PDA) and a mobile phone. When the camera takes a picture, it automatically notifies its surroundings of this. At the same time the GPS is constantly emitting world coordinates for its current location. A software component assembled with the camera and the GPS writes the current location information into the (metadata part of the) image received from the camera. The updated image is then displayed on, for example, the PDA and simultaneously sent to a web server (typically located back at the office), utilising the Internet capabilities offered by the mobile phone.

The SiteTracker, similarly, consists of four devices: a GPS, a display, a video camera, and a digital compass. The GPS constantly provides location information, and the digital compass directional information of where the video camera is pointing. The resulting video footage from the camera is shown on a display with an overlay showing exactly where the point(s) of interest would be. The GPS that takes part in this assembly may be the same GPS as the one that is part of the GeoTagger assembly – it is acting as a service in different contexts.

From a use perspective, going from GeoTagger to SiteTracker or vice versa is a matter of disassembling and reassembling a number of devices.

Development of the application prototypes takes place in parallel to the development of the open architecture, and for this reason the open architecture described above was in fact not the first architecture developed for these prototypes. For the first iteration of the GeoTagger and the SiteTracker, a prototype software architecture implementation, called Corundum (Ørbæk, 2005), was developed by the application developers themselves. Inspired by the understanding of, and vision for, use developed through fieldwork and participatory engagement with users, the prototype software architecture implementation behind these first application prototypes focused on supporting five main concepts:

- assembly – a set of communicating services,
- service – announces itself to its surroundings and communicates asynchronously with other services,
- process – contains services and components and holds a hierarchical map,
- component – a module residing on disk, can be loaded into a process, and
- hierarchical map – a tree-structured name space used to hold (most of) the non-transient data of a single process.

All components and services ran on Corundum (Bardram et al., 2004) which 'encourages an *extrovert programming style*, where components and services expose what they can do (potential uses, events accepted and sent), what they are doing (e.g. logging), and what they have been doing (history). This is all done via the h-map (see Fig. 8.12), which is globally visible and accessible from outside the process over a network' (Ørbæk, 2005). The Corundum framework differs from the first version of the open architecture described above in several respects. The devices that take part in the assembly are seen as a set of communicating services contained in processes on a network. Each of the services can be externally configured through manipulation of an externally visible hierarchical map. An assembler service also uses this hierarchical map when dynamically (re)configuring the paths of communication necessary for a specific assembly configuration. Each process potentially consists of a number of services and components.

One of the main points here is that the technical infrastructure of these early prototypes is deeply influenced by a use perspective – devices have a number of external interfaces that users configure in order to assemble the devices and to make them communicate. However, the story is, of course, more complicated than that – in order to make prototypes like the above do anything beyond the most trivial, the need arises for more pure software components and services. In the case of the GeoTagger, there is a need for a piece of logic, for example, that combines the image and the

```
communities/    ←list of communities that the service is a member of
  poecomm: [int: 1]
community: [str: 'poecomm']    ←current default community
bearers/    ←bearer protocols supported
  udp: [msghandler]    ←UDP/IP unicast
  udpmcast: [msghandler]    ←UDP multicast
  tcp: [msghandler]    ←TCP/IP point-to-point
discovery/    ←things related to discovery are below here
  announcer: [msghandler]    ←handles outgoing announcements
  interval: [int: 5]    ←announcement interval in seconds
  announcement: [msghandler]    ←handler for incoming announcements
  outgoing/    ←directory containing announcements that are periodically sent from here
      logging: [message: [msg:    ←this process announces a logging service
          sender: udp:poecomm:0.0.0.0:23457:srv;logging;entry;
          recipient: udpmcast:poecomm:239.3.3.4:23456:discovery;announcement;logging()]
      du1: [message: [msg:    ←the service also announces a du1 service
          sender: tcp:poecomm:0.0.0.0:23456:srv;du1;entry;
          recipient: udpmcast:poecomm:239.3.3.4:23456:discovery;announcement;du1()]
  listeners/    ←one may install listeners here if they too need to hear incoming announcements
      diruser: [msghandler]
```

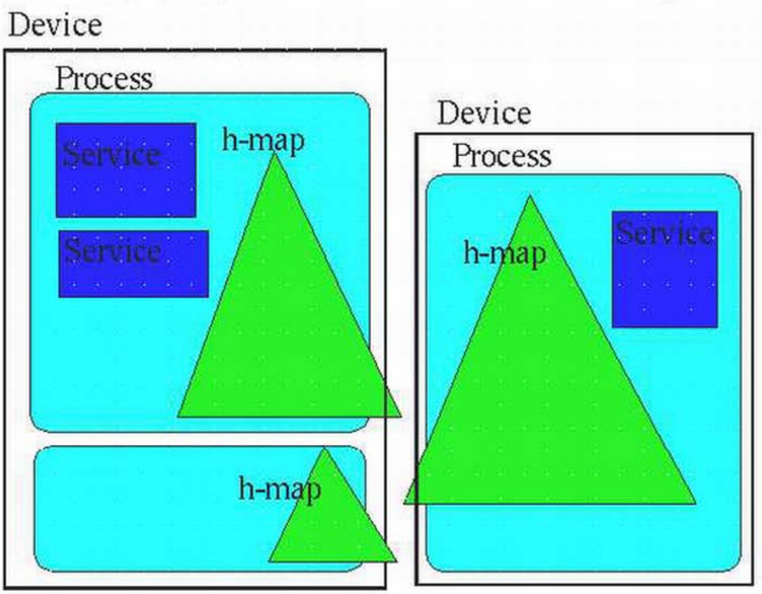

Fig. 8.12. Hierarchical maps: 'Two devices, one hosting two processes each with their own h-map. The h-maps extend outside the devices to illustrate that they are accessible from the outside'. The listing on the right is a commented dump of the h-map of an isolated instance of a simple service (du1), in a situation where it cannot see other services. It is one of the simplest real-world examples (Ørbæk, 2005)

coordinates. Since services are distributed and able to dynamically discover and use each other, this service can in principle reside on any of the participating devices. However, making an informed decision – by the user or (semi-) automatically by a run-time system – about which device to run such a service requires some degree of software architectural support for visibility and inspectability of available resources (processing power, available memory, network bandwidth, etc.) – all matters that were to become central to the open architecture.

Fig. 8.13. SiteTracker

Use Perspective

Turning back to the fieldwork, the prototypes were put to use with two landscape architects to carry out some initial experiments (Fig. 8.13). The SiteTracker, for example, produces useful, dynamic, composite pictures that accurately track specified points in the landscape. This is first achieved in a static context. Subsequently, when the experiment is repeated in a moving car, the prototype continues to work accurately. Unfortunately, the soldering on the connection to the digital compass breaks after just a few minutes of driving. We experiment with the compass internal to the GPS, but it does not provide updates fast enough and the experiment has to be abandoned.

In the course of the experiment, a number of difficulties arise that are inspiring for redesign. We only outline difficulties and design implications for the software architecture, as our focus in this chapter is on the participatory process, not the detailed design of the software architecture or the application prototypes (for more detail on the architecture design, see Andersen et al., 2005)).

Difficulty	Implications for design or design process
Calibrating the compass and the GPS is awkward. Calibration seems to be fragile and requires frequent repetition of the calibration process.	Ways of detecting trouble caused by faulty calibrations and practices of testing the accuracy of calibrations should be supported.
The wrapped setup – with cables and laptop – is clumsy, and it is difficult to see anything on the screen in the sunlight.	To enable real users to experiment with the prototype in as realistic as possible use situations, a less complex design is required
The translation from GPS to ordnance survey (OS) coordinates is faulty. The cause is unknown. The problem is fixed by driving to a known point of interest and recording the position in OS coordinates.	The detection of faults in the computation, and ideally their causes, should be supported.
When trying to reassemble the SiteTracker after a break, it turns out that a LAN/Wi-Fi type network has to literally be put in place before an assembly can be made. This is be-	Software architecture should not require connection to LAN/Wi-FI infrastructure in its physical surroundings as such infrastructures will typically not be present when on the move. Generally, the

cause, in order to exchange messages, services and assemblies at this stage require the presence of a network connection that supports UDP. On the Windows laptop this is only present if the laptop is connected to such a network infrastructure in the physical surroundings. Therefore, it is not possible to assemble using just the single laptop, the camera, the GPS, and the compass.	software architecture should be able to scale from working in infrastructure-rich environments to the infrastructurally simpler environments.

A second round of experiments with a modified SiteTracker prototype takes place a few weeks later. This time, however, the developers run into a whole series of problems right from the start. These, too, reveal pertinent design issues:

Difficulty	Implications for design or design process
When connecting a device it is sometimes necessary to find the virtual com port to which it connects in order to make the service communicate with the device via this port. The virtual com port number is dynamically assigned whenever such a device tries to connect – the com port may change depending on how many devices are currently connected.	When devices connect they should automatically acquire the necessary resources for establishing the connection. On the other hand, in case of a failure, there should be support for making such connection resources visible.
Currently, an assembly is invoked via an XML specification in a file, that can be located on any one of the devices involved in the assembly, and the meta-assembler – the service responsible for setting up and maintaining the assembly – then looks around and sets up communication between services, it does not start them.	There is a need for an overview of the assemblies available for launch and a mechanism to invoke an assembly in such a way that it automatically attempts to start the required services.
Connectivity is still required before an assembly can be made, even if only one computer is involved. Is this a design flaw in the Corundum framework? A constraint from Windows? An IP problem? If the computer on which a service is running does not have network connectivity, it is impossible to transmit messages.	The software architecture should support tools for monitoring communication paths and network traffic.

The SiteTracker loads points of interest from a configuration file on start-up. The easiest way to add or change points of interest is currently to manually change the configuration file and then to restart the SiteTracker service. Corundum actually supports on-the-fly changes, but there are no tools to support actually doing it.	Better tools for inspection and change of the state of a running service. Use exposes a missing link between prototype and architecture.
There is no mechanism or user interface to see or select what configuration file the SiteTracker actually reads from when started. In this instance there are two different files, one with UK OS and one with Danish position information.	The options and selections should be inspectable. There could be a need for detecting and visualising the physical context in which services and assemblies exist. This may also be subject to (semi-) automation, for example, on the basis of location.
The translation between GPS and OS is wrong, but we do not know where it goes wrong.	It would be nice to be able to take a service out of its current assembly and network context and simply test it by 'poking' it with some input seeing if it comes back with a proper output.
It would sometimes be useful for the SiteTracker user interface to visualise the coordinates sent to it from the GPS service.	The basic state of any service should by default be able to be shown in a graphical user interface and it should be possible to dynamically combine and change user interfaces while services are running.
In the experiment the tracker 'hands' (Fig. 8.5) jumped from one side of the display to the other. This could have several causes – the field of view could be too big, the point of interest could be behind, the coordinates could be wrong. In a later trial it turns out that this issue was caused by the assembly not being properly assembled – that is, communication paths were not properly set-up and the SiteTracker service was using outdated and flawed position and orientation data for its calculations. A further test in Aarhus reveals similar problems, but here the tracking is correct. This suggests that there are also conversion failures.	Again this calls for tools and architecture support for getting an overview of running services, their paths of communication and whether or not they are participating in a running assembly.

In general, the difficulties encountered in the use experiments show that there are more activities taking place, with more potential for things to go wrong than were anticipated, which impact at the level of the architecture as well as the level of the prototypes.

8.3.2 Assemblies as Service Composition

As we saw in architecture description the beginning of Section 8.3.1, an assembly is defined as a set of cooperating services. In this section we consider the practical implications of this and how it should be realised, again from software architecture, application prototype, and use perspectives.

Software Architecture Perspective

A main conclusion from the use perspective regarding basic assemblies was that the open architecture should support introspection and visibility in various ways (Section 8.3.1). Though this was always known in principle, experience from use enabled it to be given specific content. Section 8.3.3 explains how the concept of 'resources' partly helps meet the challenges. The software architecture was evolved to support this through a refinement of the idea of assemblies as sets of services, eventually leading to the realisation that the software architecture also needs to support a more complete concept of assemblies.

As part of this refinement process, the investigation of the basic concept of 'assemblies as services', led to exploration and refinements of the assembly concept based, among other things, on what services are traditionally thought to be in software architecture (Szyperski, 1998). One example of this would be the classification of services as 'stateless' or 'stateful'. A stateless service does not retain a client-specific state (such as the latest GPS coordinate of a specific client) between uses of the service whereas stateful services may do so.

Such a distinction is important for (among other things) reasons of scalability (and understandability) of service composition and use in software architecture: if a service is stateless it may be replicated so that different clients access different instances of run-time components and conversely many clients may use a resource-intensive service concurrently. 'Scalability' is an example of an 'architectural quality' (Bass et al., 2003) that exemplifies architectural concepts and practice that are important in designing software architectures. Most architectural qualities correspond to architecturally significant 'external qualities' of ISO 9126 (ISO/IEC 2001). In contrast, the qualities that participatory architectural design is also concerned with are qualities in use (effectiveness, productivity, safety, and satisfaction as seen from the point of view of ISO 9126). As a result of field studies and workshops, the assembly concept as outlined above was thought to support desirable qualities in use. On the other hand, little stress was put on external qualities such as performance or scalability in the Corundum prototype and in the h-map implementation. Thus, these qualities remained to be explored in the context of software architecture. The participation of 'travelling architects' in some of these use experiments (the third of the four participatory elements of the design of the open architecture introduced earlier) helped to communicate the importance of the approaches adopted in Corundum, and to effect their transfer to the open architecture.

A further refinement of the assembly concept was the realisation that assemblies (at this state of the project) could be thought of as primarily and mainly *service compositions.* As a consequence, it was considered to remove assemblies as a first-class concept in the architecture: if assemblies were only (dynamic) service compositions, their realisation could have been expressed in terms of reference compositions of components/run-time components. However, for reasons discussed in more detail in the section below, it was decided that it was necessary and beneficial to leave the assembly concepts as a central and first-class part of the open architecture. The architectural refinement of the concepts of assemblies and services was then used in application prototyping as discussed below.

Application Developers' Perspective

The second major iteration of the prototypes involved a move towards a more fine-grained, service-oriented architecture as defined by the open architecture. Figure 8.14 shows a schematic of the first version of the SiteTracker assembly, combining a GPS service, emitting basic GPS location information, a compass service, emitting compass direction, and a SiteTracker service combining video images with location and directional information.

Also integrated into the SiteTracker service was a so-called 'GeoParser' component. This component took raw GPS protocol strings (nmea-0183), parsed, and converted them into a coordinate system that was appropriate to do the mathematics involved in locating the points of interest in the video image. Experience in use and other considerations (outlined below) suggested that this structure needed to be

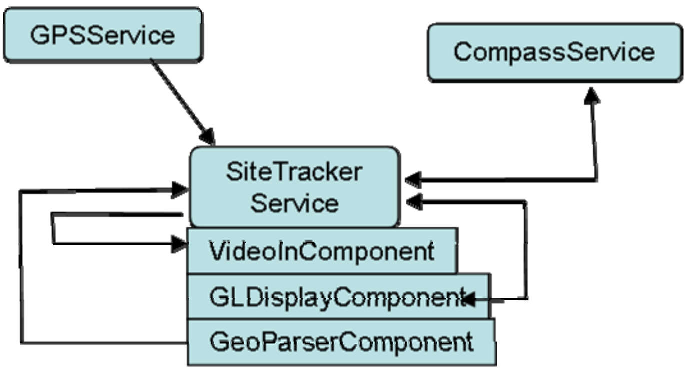

Fig. 8.14. Simplified view of the original SiteTracker services, components, and communication paths, showing that there are three services involved: GPS, Compass, and SiteTracker; and that the last loads and uses three components inside it (arrows depict paths of communication)

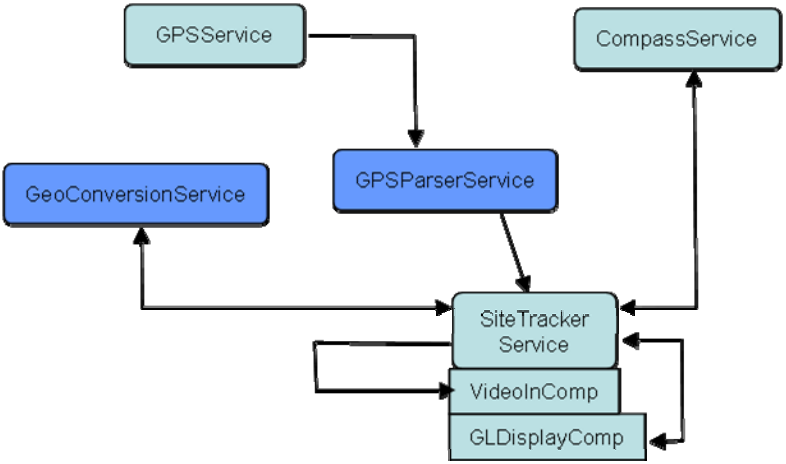

Fig. 8.15. Simplified view of the second version of SiteTracker services, components, and communication paths

changed. The second version of the SiteTracker (Fig. 8.15), for example, breaks the GeoParser functionality into two: a basic GPSParserService for parsing the GPS protocol strings (emitting coordinates in latitude and longitude) and a GeoConversionService for converting between different geographically related coordinate systems. Furthermore, these functionalities were no longer loaded directly into the main SiteTracker service but instead acted as separate services in their own right.

There are several reasons for this small but significant change in the software architecture of the prototype. Firstly, in a general architectural context we wanted to further explore the scalability and service composition qualities of the prototypes. For example, if the main SiteTracker service is running on a resource-constrained device, the conversion and parsing services can be deployed on separate devices in the network in order to achieve better load balance. Also, since the parsing and conversion services are more or less stateless, other services can dynamically attach to them and make simultaneous use of their functionality – saving the need to load the component in more than one place and making efficient use of available computing power on the network. In the latest version of the SiteTracker this is put to practical use when a landscape architect wishes to supplement the augmented video image of the SiteTracker service with a digital map showing their current position and the positions of points of interest. This map service also needs to parse and convert coordinates and therefore looks up running versions of these services on the network and assembles itself with them in order to show the updated information.

As users of the SiteTracker, the landscape architects will not see any changes in functionality through this underlying change of architecture. However, as users of the software architecture, they (and the software developers) experience a significant improvement in relation to how flexibly the parts of the system can be composed, decomposed, and deployed. This enhances the end-users' experience, in that the assemblies and constituent services lend themselves to a richer set of options in relation to end-users composing their own assemblies.

Furthermore, the use experiences gained from the first experiments, as explained in Section 8.3.1, strongly indicated the need for better tools for inspection and awareness in relation to services and assemblies and their context. At this point in time we therefore, firstly, initiated development of a basic tool for the browsing and composition of running services and assemblies in the network – an 'inspector'. Secondly, the need for inspection of any single service on the network – potentially from a remote location – inspired initial work on the design of a framework for remote inspection and control of services.

Use Perspective (Application Developers as Users)

The developers of the application prototypes are closely engaged in the design of the open architecture. In fact, as we saw, they themselves designed a first prototype implementation of a PalCom open architecture, *Corundum*, in parallel to more comprehensive and conceptual efforts on the part of the software architects. They also use this and subsequent iterations of the PalCom open architecture as part of their development and implementation work, and they are, therefore, an invaluable resource in the participatory design process. The goal of the open architecture is to support people from different walks of life, with different levels of 'computer literacy', and engaged in different situated activities in making computational states and processes palpable. The challenge is to enable the production of appropriate reflections of computational states and processes (Dourish, 1995) or otherwise 'sensible' data. Software application developers are highly IT-literate users. By examining their current practices of making computational states and processes palpable and by engaging them in a participatory design process important insights for the design of the PalCom open architecture can be gained. This corresponds to the last of the four participatory elements of the design of the PalCom open architecture.

On the right hand side of Fig. 8.16 we see the SiteTracker and other prototypes working at a 'Future Laboratory' with users from another application domain – emergency response services (police, fire brigade, medical teams) – at the emergency services training ground in Aarhus, Denmark. Future Laboratories enable users to

Fig. 8.16. Developing applications and services on prototypes of the PalCom open architecture

'colonise' and shape a socio-technical future by asking and allowing them to accomplish realistic work with functional prototypes in 'as realistic as possible' work settings (Büscher et al., 2004). The commitment to serious hands-on simulation and exploration of real-world work enables embodied, practical creativity, and reflection as well as participatory evaluation. Here, we have staged a car pile-up, paramedics are putting biosensors and locators that will be part of the SiteTracker assembly on victims, and someone is taking pictures of the victims. That data is sent to the trauma doctor in a prototype acute medical coordination centre. The trauma doctor needs to decide the hospital to which the victims should go, taking into account the nature of their injuries and the special skills at the different hospitals. Amongst other applications and services, the SiteTracker and GeoTagger are used to take pictures of the individual patients at the scene of the accident and display them on one of the screens in the acute medical coordination centre.

Future Laboratories foster the emergence and evaluation of future practices, which is particularly important when it comes to involving users in the design of software architectures envisaged to support the use of ubiquitous computing as it emerges over the next decades. Our series of Future Laboratories is still in progress and will be the subject of future publications. But the fact that Future Laboratories with end-users require functional prototypes means that developers have to create, assemble, and test them extensively, in effect inventing and evaluating emerging future practices of developing software applications – carrying out 'Future Laboratories' of development work.

Below we present an analysis of events on the day before the Major Incidents Future Laboratory, when developers were making the prototypes work, coding, assembling, and addressing difficulties by making their causes 'palpable' wherever this is possible with the support of the prototype PalCom open architecture.

Jesper is assembling the SiteTracker. He looks, waits, then exclaims: 'What?' and reads out loud: 'no cameras are currently connected', reaching for the network cable as he speaks (Fig. 8.17). Michael saw that the camera had stopped responding, and turned the switch to wake it up, but there are also messages about failed 'decryption'. They speculate about these errors until they hear Esben laughing behind them. To debug, Esben changed the Java version of the Corundum architecture prototype so it does not encrypt anymore. Because Jesper is receiving messages from services on the Java architecture and his C++ version of the architecture tries to decrypt them, they are getting errors, but this is not what is causing the lack of connection between the camera service and the display service.

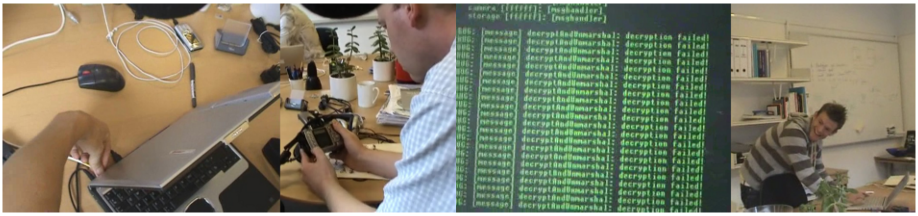

Fig. 8.17. Assembling the SiteTracker

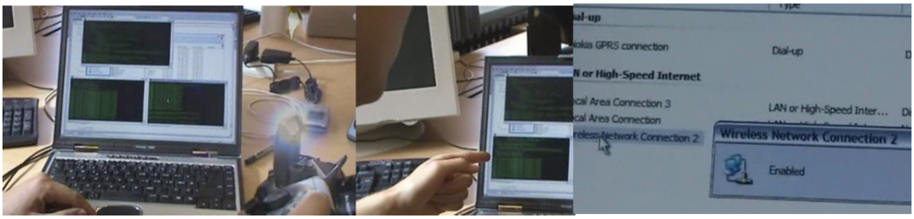

Fig. 8.18. Trying to figure out what is wrong

Jesper's hand reaches for the network cable again. Similar trouble was caused earlier by network problems. He leaves the cable plugged in, though, and does a number of things:

He picks up the camera and takes a picture (Fig. 8.18), and notes that where it should say 'get file', nothing happens, while the assembler says 'assembly is possible' and is, indeed, assembling. The meta-assembler is adding subscriptions to services. Only the day before Jesper took five pictures and it went 'tick tick tick, they arrived with "get file messages" …'. The network is still the prime suspect. Jesper unplugs the cable and switches to the wireless network, but to no avail. While Esben's sensor services are working nicely, Michael and Jesper are frustrated. They download and install a loopback adaptor to create a 'one machine guaranteed functioning network' to check conclusively whether the problem is network related. It is not. This takes about 30 minutes.

Michael suggests checking each individual subscription. They start the 'inspector' and examine what is going on. It does not help, and desperation sets in. They restart the machine. They consider dropping the prototype from the experiments at the emergency exercise.

They wonder if there are too many images on the camera (which has also been a problem in the past), but again, no. Jesper explains their current understanding of the problem:

> ... for some reason the assembler doesn't finish the job. It doesn't set up the subscription between the two services. It can see both, it attaches to both and the next step is actually to set their subscriptions up and for some reason it fails that. So when I press the button, due to the fact that the camera does not have any subscribers it does not send a picture out on the network and then the service that is supposed to display it never receives it.

The machine is back up and running and it works … once. The second time nothing happens. Jesper suspects that he tried to take another picture too soon and waits a moment before he takes another one, and it works again.

When the GeoTagger and the SiteTracker work during the Future Laboratory with the emergency personnel, the developers notice a strangely long delay between taking the picture and it showing up on the display. A week later, at another demo, they figure out some of what is wrong. For example, when the camera takes the picture, it is so busy it stops sending even a heartbeat – a simple message saying 'I'm alive' to the other services. This breaks the assembly. As soon as the heartbeat is back, the assembly is re-established – but this takes time – and only when it is done can the overview service display the picture. The solution is to run the heartbeat in a separate thread or as part of the communication layer instead of sharing a thread with the camera data and processes.

This story informed a day-long 'fieldstorm': a data session with application developers and software architects where the aim is to generate ideas for technologies that could support the work of developers in making the causes of failures (and possibilities for creative assembly) palpable.

The discussion brought out a list of methods of finding out what is going on:

- The developers insert print commands into the code to produce messages (like 'assembly is possible', 'assembling', 'no cameras connected').
- People make amplifiers/translators for themselves (like the inspector).
- There is something like 'pattern recognition'. Flows of messages 'look right' when things are working and 'wrong' when something is wrong.
- There are other sensory clues (e.g. the sound of Mac storage in infinite loop).
- There is categorisation: specific message types 'belong' to specific processes.
- There is a strong sense of sequence and timing, which helps sense whether things are going well or badly.
- There is a temptation to re-create 'good' (i.e., well-known) environments where things worked even when that is not necessary.
- People pose hypotheses of what might be wrong and falsify.
- There is a temptation to test things one can easily test, especially under time pressure, and to ignore potential causes that are outside one's scope.
- A lot of the process of encountering and dealing with trouble is made public by 'talking out loud to the machine' ('no cameras currently connected').

Finding out is a mixture of 'intuition, detective work, collaboration, and trial and error'. The skills that some developers bring to the matter of computational potential are remarkable. However, they are not just special talents, but also the acquired and honed result of everyday practical engagement with computational technologies. Perceptual acuity and analytic proficiency can be trained. They rely on the reflexivity of interaction with computational matter.

The term 'reflexivity' as it is used here is inspired by notions of the spontaneous, 'knee-jerk' reflexive reaction to physical stimuli, and the mutually defining, reflexive character of moves in human–human interaction highlighted by ethnomethodology (Garfinkel, 1967; Lynch, 2000). It does not imply deliberate reflection. In interactions with each other, but also with technologies and the material world, people treat appearances 'as "the document of," as "pointing to," as "standing on behalf of" a presupposed underlying pattern' (Mannheim, quoted in Garfinkel, 1967). In human–human interaction, this 'documentary method of interpretation' is sequentially organised and reflexive, that is, each move – each utterance, silence, gaze, or embodied behaviour – is shaped by preceding and subsequent events. Each move prospectively informs the next and retrospectively shapes what has happened before. Each move documents a particular understanding of what is going on, and, as such, shapes the interaction as a whole – for example, as an informal conversation, a meeting, or a medical consultation.

Although in human–matter interaction only one partner is sentient, engagement relies on similarly reflexive, sequentially organised moves and documentary methods of interpretation. Materials 'document' their processes or states and their 'understandings' of moves that their human or non-human counterparts make in the interaction. In everyday encounters with materials, much of our human response to material moves becomes reflexive in the sense of automatic. The acts of perception, interpretation, and response are unnoticed, what is perceived is a 'flow' of activities. However, human–matter interaction in science, medicine, sport, craft, engineering, and many other activities amply documents that perception can be trained, and that ways can be found to make materials whose moves are outside of the human 'naked' sensorium speak in a way that people can sense. The developers' methods of making computational states and processes 'speak' by translating, amplifying, and eliciting documentary evidence are instantiations of such practices.

It is a major aim and a major challenge for our participatory design and research efforts to support advanced as well as 'ordinary' users' practices of making computational states and processes palpable. Paying close attention to developers' practices is one strongly informing strategy for palpable design. Participatory design with developers and end-users, based around hands-on engagement with prototype architectures and prototype applications and services, suggests that support for 'reflexivity' (as well as reflection) is a productive avenue for design.

8.3.3 Inspection and Awareness of Resources

The troubles occasioned in using the prototypes repeatedly showed the need for computational states and processes to be made palpable and 'brought into the light' so that their operation could be understood and engaged with. In this section we indicate, in very brief outline, some of the ways in which this is being achieved.

Assemblies as Resource Composition

Some of the challenges outlined above may be handled by supporting a more detailed, fine-grained, and dynamic modelling and use of resources (e.g. the load level of a CPU such as in the camera example above) in palpable computing. The need for handling resources in a detailed way in palpable systems led to the inclusion of the concept of first- and second-order resources in the open architecture (Fig. 8.19). Second-order resources encompass a diverse set of concepts (among others services, actors, and communication channels). Second-order resources, in turn, contain first-order resources which are resources found in hardware and software layers below the palpable computing open architecture. Examples include memory, storage, and battery power. An assembly consists of collections of first- and second-order resources, and communication. It has a description (of how it is assembled and how it behaves when running) and is run on a computational node running the PalCom open architecture.

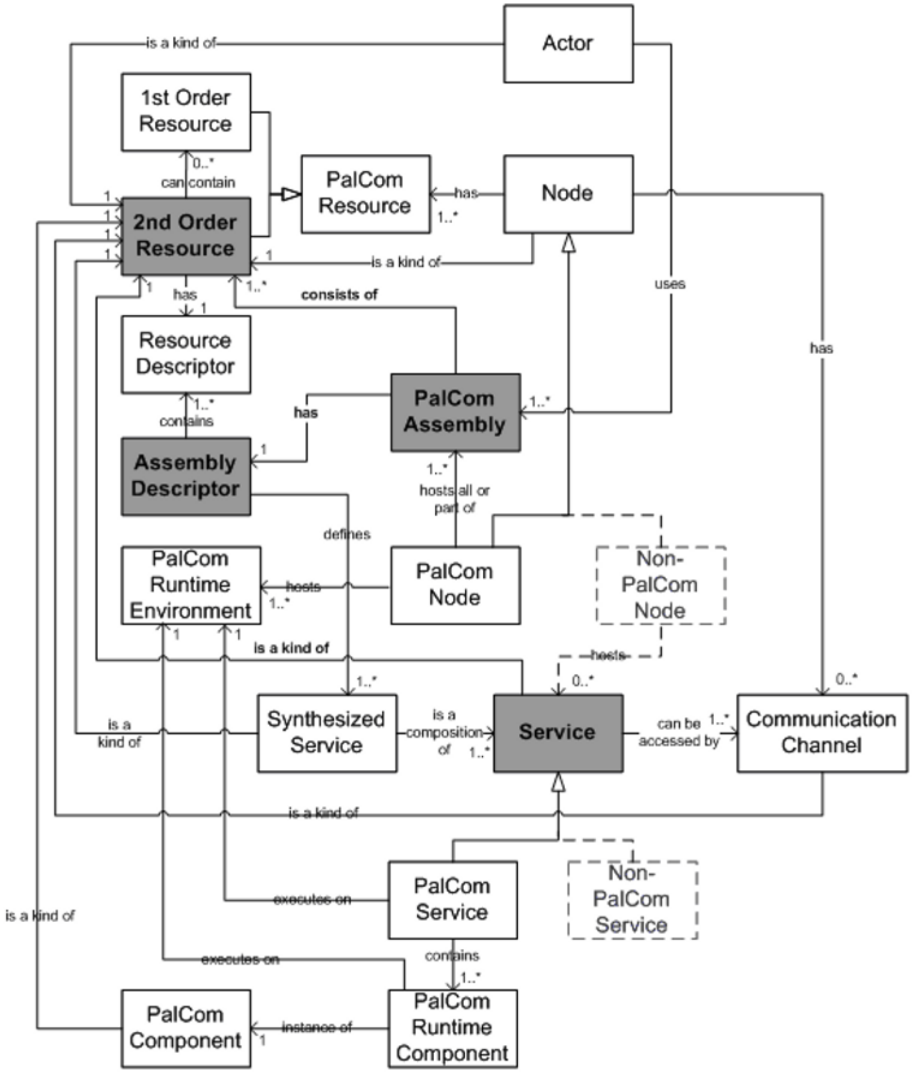

Fig. 8.19. Central concepts and relationships from the second complete version of the palpable computing open architecture (from 2006 Deliverable). The concepts are further developed and refined from the concepts shown in Fig. 8.11.

Browsing Services and Assemblies

The experiences gained from developing and debugging the prototypes as well as the lessons learned and changes made in relation to the architecture have together led to the development of a tool for browsing, combining, and inspecting services and assemblies (Fig. 8.20). This latest version of the tool is a reimplementation of the first 'Inspector' prototype of such a tool mentioned in Section 8.3.2. The new version

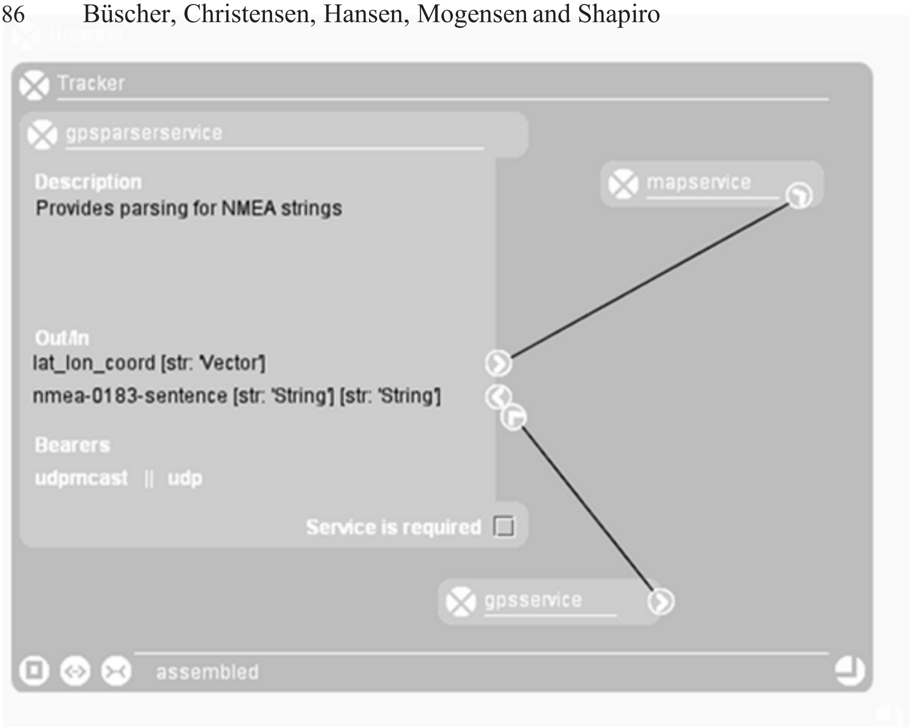

Fig. 8.20. A screen shot of the current prototype implementation of a service and assembly browse and inspection tool

combines browsing and composition functionalities with capabilities for inspection of single services and assemblies. To do this, the tool builds on top of the framework, also mentioned above, for remote control and display of services.

Through its graphical user interface, the tool shows all services running on devices in the networking context and lets the user inspect possible ingoing and outgoing connections of each service. Furthermore, the outgoing and incoming interfaces of services can be combined into assemblies and all currently running assemblies can be browsed and inspected.

This is the functionality supported by the current prototype implementation of the tool, and plans are in the near future to extend the tool with the following abilities to:

- further inspect and change the state of single services – possibly with the option to isolate the service and test it in its own 'sandbox',
- visualise required and used resources for services and possible reconfigurations of resources in relation to instantiation of assemblies,
- monitor and filter data sent between services collaborating in assemblies, and
- further inspect the properties of the context in which services and assemblies exist.

On the one hand, the construction of the tool and the functionalities added to it follow directly out of a simple set of demands stemming from the development and debugging of the prototypes described in the previous sections. However, the point is that in order to make the tool truly workable for everyday users from different walks of life, with different skills, and engaged in different use situations, the underlying software architecture has to support such functionalities. By design, any service, for example, has to support inspection and allow for the change of its state at run-time. Different forms of monitoring, browsing, and changing the behaviour of assemblies in context can be supported through the assembly concept with first- and second-order resources, encompassing, for example, other services and communication channels.

In a broader context, the development of the tools and the architecture supporting them is a way of attempting not only to reveal the materiality of digital entities, such as services and assemblies in a network, but also to provide a way of supporting the interplay and dialogue with such materials. Such support for reflexivity has a number of software architectural implications. In addition to the support for introspection of dynamic resources, that is, the *present*, the dialogue with computational material can also be based on assemblies that have been used previously, that is, the *past*, and with possibilities for assemblies in a given computational context, that is, the *future*.

Supporting users in reusing past (templates for) assemblies points to the need for distributed storage. Given the inherently ad hoc network of devices and services in palpable computing, this requirement may lead to significant changes in the communication layer of the open architecture. If users of a set of services and devices in a given context should be supported in making informed choices about possible futures of assemblies, distributed storage should be augmented with more powerful semantic models of the capabilities of available resources/services. For example, given a set of services (such as a camera and a compass service), it should be possible for the open architecture to, for example, support an application that suggests looking for other services (such as a GPS service) to create a SiteTracker assembly.

8.4 Pulling Things Together

We started out with the puzzle of what the relationship could and should be between software architecture design, application prototype design, and experience of use, and of whether these are amenable to an integrated participatory design approach. The material presented in this chapter shows that these ‘distant’ elements of a large project are indeed mutually informing, and can be made very productively so with some conscious focus and targeted methods.

Figure 8.21 summarises our participatory open architecture design iterations up to this point. The cross-connections that emerged took various forms, some more general and some direct. We began with a set of scenarios, formulated through ethnographic fieldwork, participatory analysis and design workshops, prototype design and experiments. The scenarios envisage how landscape architects would assemble and use sets of devices and services whose processes and affordances can be made palpable, supported by the open architecture under design. Focusing on the real needs of skilled practitioners produced initial requirements for palpable computing that were more specific and problem oriented than could be expected from an attempt to consider ‘ambience’ in the abstract.

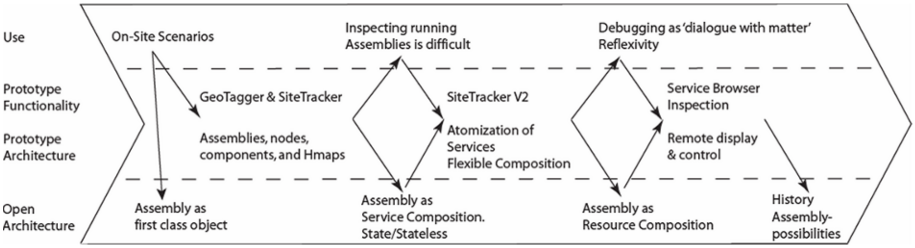

Fig. 8.21. Map of the start of the participatory design of the PalCom open architecture

The application prototype designers could not wait for a considered open architecture to be ready, and so programmed an 'extreme' version of their own. Because this was available for early experiments with the prototypes in use, further problems, limitations, and needs were exposed. Due in part to the cross-participation of personnel, going beyond conventional communication between software design subdisciplines, these lessons were incorporated into the design for the open architecture, where they were generalised to the demands of other settings, and integrated with other practical and theoretical imperatives. Some of the lessons had a relatively specific focus, such as the atomisation of various services for the SiteTracker. Some were more far reaching, such as the need for inspection and browsing services and the forms these could usefully take. At least one was structural: the adoption of the concept of 'assembly', originating in use, as a first-class object in the open architecture.

Architectural and application prototypes are now being taken to the test in more demanding realistic use situations. This requires as full as possible functionality and the application prototype developers engage in realistic testing and experimenting. This gave the opportunity to introduce a participatory design element among the computer scientists in the project themselves, by studying the ways in which application prototype designers made use of the evolving open architecture. As well as exposing further problems and needs, this demonstrated how in practice designers make software palpable, interrogating intangible materials in ways that bring them to sight and to voice. Observations of their practices of debugging reveal a reflexive 'dialogue with computational matter' that relies on rich sensory feedback. Where such sensory data is unavailable, the developers devise means of translating, amplifying, and manifesting computational processes. This provokes the need, and shows some of the possibilities, for 'reflexive design', and this is being addressed in the third iteration of the application prototypes. These allow users – in the first instance the developers themselves – to inspect, monitor, and perceive computational processes and affordances.

Our experience shows that the lessons learned in 'traditional' PD, namely that by involving users, more innovative and more viable socio-technical change can be brought about are equally true when it comes to architectural design. The point we are at the moment (for a report from later experiments see Büscher et al., 2008) is a gateway to more direct end-user experience of how the open architecture does (or does not) support people in making computational affordances and processes palpable at the time of writing. By observing and by engaging application developers as users in a participatory architecture design process, we have chosen a perspicuous setting where we can study and experiment with current practices of making computational processes and affordances palpable. In doing so, we gain concrete insight into the constraints and possibilities for software architecture design for palpability. If developers cannot make things palpable with the support the prototype architecture provides, then end-users would also fail.

At the heart of our approach is the observation that engagement with matter is reflexive. What this means is that we go beyond reflection. Reflection assumes that some designer somewhere can anticipate the situation and the computational literacy of the person needing a representation of computational processes. Whether the user's 'status' is chosen by the user or 'detected' through context sensors, reflection assumes that designers can pre-prepare appropriate representations. While we ourselves engage in reflective design, we are certain that it is ultimately impossible to achieve appropriateness in this way. In parallel, we therefore also choose a radically different approach: by documenting material processes as 'objectively' and at as 'atomic' a level as possible, we provide 'sensible' data. We believe that there are already standards of producing such documentary evidence emerging, and not just in our own work. People may not be able to perceive such documentary evidence with their 'naked' senses and not without training and acculturation. We build tools that can amplify, translate, and manifest such documentary evidence. This, in turn, will enable training and acculturation.

References

Andersen P., Bardram J.E., Christensen, H.B., Corry, A.V., Greenwood, D., Hansen K.M. and Schmid, R. (2005). Open Architecture for Palpable Computing Some Thoughts on Object Technology*, Palpable Computing, and Architectures for Ambient Computing. Ob-ject Technology for Ambient Intelligence Workshop*, Glasgow, U.K. *Proceedings of ECOOP 2005*.

Anderson, S., Hartswood, M., Procter, R., Rouncefield, M., Slack, R., Soutter, J. and Voss, A. (2003). Making Autonomic Computing Systems Accountable: The Problem of Human–Computer Interaction. *In Proceedings of the 1st International Workshop on Autonomic Computing Systems, 14th International Conference on Database and Expert Systems Applications*, Prague, September.

Bansler, J.P. and Havn, E. (2006). Sensemaking in technology-use mediation:Adapting groupware technology in organizations. *Journal of Computer Supported Cooperative Work*, v.15, n.1, pp. 15–55.

Bardram, J., Christensen, H.B. and Hansen, K.M. (2004). Architectural Prototyping: An Approach for Grounding Architectural Design and Learning. *In Proceedings of the 4th Working IEEE/IFIP Conference on Software Architecture (WICSA 2004)*, pp. 15–24, Oslo, Norway.

Bass, L., Clements, P. and Kazman, R. (2003). *Software Architecture in Practice*. Addison-Wesley, 2nd edition

Bellotti, V., Back, M., Edwards, W., Grinter, R.E., Henderson, A. and Lopes, C. (2002). Making sense of sensing systems: five questions for designers and researchers. In: Terveen, Loren (ed.): *Proceedings of the ACM CHI 2002 Conference on Human Factors in Computing Systems Conference*. April 20–25, 2002, Minneapolis, Minnesota. pp. 415–422.

Belotti, V. and Edwards, K. (2001). Intelligibility and Accountability: Human Considerations in Context aware systems. *Human-Computer Interaction*, v. 16, pp. 193–212.

Büscher, M., Kristensen, M. and Mogensen, P. (2008). When and how (not) to trust IT? Supporting virtual emergency teamwork. In: Fiedrich, F. and Van der Walle, B. (Eds.): *Proceedings of the 5th International ISCRAM Conference*, Washington, DC, USA, May 2008.

Büscher, M. (2006). Vision in motion. *Environment and Planning,* v. 38, n. 2. February, pp. 281–299.

Büscher M., Mogensen P., Agger Eriksen M. and Friis Kristensen, J. (2004). Ways of grounding imagination. *Proceedings of the Participatory Design Conference (PDC)*, Toronto, Canada, 27–31 July 2004 pp. 193–203.

Chalmers, M. (2003). Seamful Design and Ubicomp Infrastructure Proceedings of the Ubicomp 2003 Workshop 'At the Crossroads: The Interaction of HCI and Systems Issues in UbiComp'.

Chalmers, M. and Galani, A. (2004). Seamful Interweaving: Heterogeneity in the theory and design of interactive systems. *Proceedings of DIS 2004*: pp. 243–252.

Corry, A.V., Hansen, K.M. and Svensson, D. (2006). Travelling architects – A new way of herding cats. *Quality of Software Architectures (Lecture Notes in Computer Science 4214)* Berlin: Springer, pp. 111–126.

de Paula, R., Ding, X., Dourish, P., Nies, K., Pillet, B., Redmiles, D., Ren, J., Rode, J. and Silva Filho, R. (2005). In the Eye of the Beholder: A Visualization-based Approach to System Security. *International Journal of Human-Computer Studies*, v. 63, n. 1–2, pp. 5–24.

Dourish, P. (1995). Developing a reflective model of collaborative systems. *ACM Transactions on Computer–Human Interaction*, v. 2, n. 1, pp. 40–63.

Dourish, P. (2003). The appropriation of interactive technologies: Some lessons from Placeless Documents. *Journal of Computer Supported Cooperative Work*, v. 12, pp. 465–490.

Dreyfus, H.L. (1992). What Computers Still Can't Do: A critique of artificial reason. Cambridge, MA: MIT Press.

Edwards, K., Belotti, V., Dey, A.K. and Newman, M.W. (2003). Stuck in the middle: The challenges of user-centred design and evaluation for infrastructure. *Proceedings of the ACM Conference Conference on Human Factors in Computing Systems*, Florida.

Garfinkel, H. (1967). *Studies in Ethnomethodology*. Polity.

Hartswood, M., Procter, R., Schopf, J. M., Slack, R., Ure, J. and Voss, A. (2006). Abstractions, Accountability and Grid Usability. *Second International Conference on e-Social Science 28–30 June 2006, Manchester, UK.*

Ingstrup, M. and Hansen, K,M. (2005). A Declarative approach to architectural reflection*. 5th IEEE/IFIP Working Conference on Software Architecture WICSA*.

ISO/IEC (2001). Software Engineering – Product Quality. Part 1: Quality Model. ISO/IEC 9126-1.

Lynch, M. (2000). Against reflexivity as a academic virtue and source of privileged knowledge. *Theory, Culture and Society*, v. 17, n. 3, pp. 26–54.

MacLean, A., Carter, K., Lövstrand, L. and Moran, T. (1990). User-tailorable systems: Pressing the issue with buttons. *ACM Conference Conference on Human Factors in Computing Systems*: pp. 175–182, Seattle.

Ørbæk, P. (2005). Programming with hierarchical maps. *Technical Report DAIMI PB-575*, DAIMI. Available at: http://www.daimi.au.dk/publications/PB/575/PB-575.pdf

PalCom External Report 50: Deliverable 39 (2.2.2) Open architecture (2006). *Technical report, PalCom Project IST-002057, December*. Available at: http://www.ist-palcom.org/publications/review3/deliverables/Deliverable-39-[2.2.2]-open-architecture.pdf

Parnas, D.L. (1972). On the Criteria To Be Used in Decomposing Systems Into Modules. *Communications of the ACM*, v. 15, n, 12, pp. 1053–1058, December.

Rimassa. G., Greenwood, D. and Calisti, M. (2005). Palpable computing and the role of agent technology. Proceedings of Multi-Agent Systems and Applications IV, 4th International Central and Eastern European Conference on Multi-Agent Systems, CEEMAS.

Stroud, R. (1992). Transparency and reflection in distributed systems, In Proceedings of the 5th workshop on ACM SIGOPS European workshop: Models and paradigms for distributed systems structuring.

Suchman, L.A. (1987). Plans and Situated Actions: The Problem of Human-Machine Communications. Cambridge, UK: Cambridge University Press.

Szyperski, C. (1998). Component Software – Beyond Object-Oriented Programming. New York: Addison-Wesley.

The PalCom project (2006). Available at: http://www.ist-palcom.org/ February.

Weiser, M. (1991). The Computer for the Twenty-First Century. *Scientific American*, September, pp. 94–10.

Weiser, M. (1993) Some Computer Science Problems in Ubiquitous Computing, *Communications of the ACM*, July (reprinted as "Ubiquitous Computing". Nikkei Electronics; December 6, 1993; pp. 137–143.)

Chapter 9

Global Software and its Provenance: Generification Work in the Production of Organisational Software Packages

Neil Pollock[1] and Robin Williams[2]

[1] University of Edinburgh, School of Management, Neil.Pollock@ed.ac.uk
[2] University of Edinburgh, Research Centre for Social Sciences, R.Williams@ed.ac.uk

9.1 Introduction

This chapter addresses the seemingly implausible project of establishing a 'generic' organisational information system. There is an apparent contradiction: on the one hand, we are told of the diversity of specific organisational contexts, and on the other, we often find *the same* standardised software solutions being applied across those settings. How do generic software packages work in so many different contexts? Science and technology studies (STS) provides contrasting accounts of how this contradiction is resolved: either stressing the unwanted organisational change that standardised systems may bring; or, alternatively insisting these technologies can only be made to work through processes of 'localisation'. We argue that the focus on specificity versus localisation of application contexts draws attention away from enquiring into the origins and characteristics of generic solutions. Through comparing the design and evolution of two software packages, we shift the debate from understanding how technologies are made to work within particular settings to how they are built to work *across* a diverse range of organisational contexts. Our question is 'How do software packages achieve the mobility that allows them to bridge the heterogeneity within organisations *and* between organisations in different sectors and cultures?' We describe a set of revealed strategies through which suppliers produce software that embodies characteristics common across many users; what we term *generification work*. One aspect of this process of generification is not only the configuring of users within 'managed communities', but it also includes 'smoothing' the contents of the package and, at times, reverting to 'social authority'. Our argument is that generic systems *do* exist but that they are brought into being through an intricately managed process, involving the broader extension of a particularised software application, and, at the same time, the management of the user community attached to that solution.

A. Voss et al. (eds.), *Configuring User-Designer Relations*,
DOI: 10.1007/978-1-84628-925-5_9, © Springer-Verlag London Limited 2009

Complex organisational information systems do not travel. Berg (1997) suggests that the difficulties in transporting such systems from one place to another arise because they become fixed in 'time' and 'space'. His argument is that software becomes so thoroughly imbued with the local idiosyncrasies of its place(s) of production that it only works at the site(s) for which it was designed and built. Scholars in (STS) and other fields have spent much time describing how building anything other than the simplest artefact produces this kind of *particularisation*. There are dozens of such examples in STS of how manufacturing planning systems, finance sector administrative systems, hospital information systems, and, to use Berg's example, expert systems resist transfer to other settings.[1]

Yet, there is a curious contradiction. Despite familiar-sounding stories of failed or problematic technology transfer, there are, of course, many types of software that do appear to be highly mobile. For instance, enterprise resource planning (ERP) systems, the name given to one of the most popular types of integrated organisational information system, are used in diverse places and appear oblivious to the form, function, culture, or even geography of organisations.[2] Such has been their ability to transcend their place of production that they are now described as 'generic' or even 'global' solutions.

How are we to understand travelling software through the lens of STS?[3] Some have sought to question their existence, disputing whether there is such a thing as a generic system. According to this argument, a truly global system is a modernist dream: there are no 'genuine universals' in large-scale information technologies (Star and Ruhleder, 1996, p. 112); and their creation is akin to 'hunting for treasure at the end of a rainbow' (Hanseth and Braa, 2001, p. 261). An alternative STS approach has been to highlight the effort of local actors in making these systems work in specific local settings (McLaughlin et al., 1999). From this perspective, we might focus on the all too apparent gulf between the software presumptions and actual

[1] Webster and Williams report on the difficulties and frequent failures encountered when computer-aided production management (CAPM) systems, designed for large hierarchical American manufacturers, were implemented within the more informal, ad hoc managerial culture and practices of smaller British manufacturers (Webster and Williams, 1993). Fincham et al. (1994) identify similar problems in the transfer of packaged finance service sector administration systems from the USA to the UK where a lower and less formal division of labour prevailed. McLaughlin et al. discuss the transfer of a hospital management system from one national context to another and suggest that because the system was particular to its geographical birthplace it did not easily translate to new contexts (McLaughlin et al., 1999).

[2] It has been argued that by the late 1990s most large companies had adopted the same or a similar ERP system (Muscatello et al., 2003). Moreover, these systems are now jumping the boundary from the private to the public sector and are moving into local authorities, hospitals, and universities, a move portrayed by many as also highly unlikely.

[3] Science and technology studies (STS) is the subdiscipline which grew out of the sociology of science. It has arguably been one of the most productive bodies of work for the study of the development, uptake, and use of technology and it has been extensively deployed to study information systems and software (see Walsham, 2001).

working practices at the settings where the solution is adopted (as well as the active processes whereby humans repair these deficiencies). Despite these objections from within STS, the notion of a generic technology continues to be a powerful and an attractive idea. There are many software suppliers, for instance, who act as if it were possible to build such an object. It is not our intention to refute the rhetorics of technology suppliers who claim to create universal solutions to organisational activities; instead, we intend to take seriously their ambitions and strategies to create such solutions. Rather than focus on the effort of 'localisation', and thus highlight the already well-researched 'collision' of system and setting, we seek to examine the much less investigated and poorly understood process through which systems are designed to work across many contexts. Indeed, we think it odd that STS has little to say about generic software, given that, as we discuss below, from its earliest days it has concerned itself with how knowledge is made to transcend its place of production.

Why might this be so? Perhaps this relates to a more fundamental problem where contemporary social scientific analyses are not good at thinking about movement (Cooper, 1998). STS is interested in how technologies are translated for new contexts: and, of course, a kind of movement is examined in these studies, but the primary interest is in the process of translation as a matter of *localisation*: of how software is both made to work *within* a specific setting and how it transforms that setting. There are limitations with movement as 'simple location' applied to generic software, to use Alfred North Whitehead's term (1967; cited in Cooper, 1998, p. 108). For instance, there is little concern for any transformations in the thing that is moved, such as with the ways in which the software package is explicitly designed (and redesigned) to work *across* settings. We find it odd that there is such a wide-ranging set of terms in STS to describe the way standardised technologies are 'imported' (…'domesticated', 'appropriated', or 'worked-around') into user settings, while there is a comparative lack of emphasis on the reverse process through which an artefact is 'exported' from the setting(s) in which it was produced. This is striking since the bulk of organisational software in use today is produced in this way – the same systems are recycled from one context to another. By attempting to develop the beginnings of a vocabulary to capture this exporting, we describe the practice of making software generic (generification work), including its various explicit and revealed generification strategies, as the process of *generification*.[4] Through discussing a number of generification strategies, we hope to offer novel or fresh insight into the design and use of software packages.

The design of generic packages differs from earlier software development traditions. Suppliers traditionally developed close ties with customers, the conventional

---

[4] While we do not know of any studies of technology that use this terminology (generification work, the process and attendant strategies of generification), Errington and Gewertz (2001) provide an interesting discussion of generification in terms of the local culture of indigenous peoples and how it is affected by other, more dominant forms of knowledge. We work up the notion of generification because we think it indicates a way of making sense of how software packages are developed and recycled, and also provides a counter to biases towards localization arguments within current STS.

wisdom being that increased knowledge of users would lead to better design. In contrast, generic solution suppliers are said to actively keep users at a distance, fearing that their software will become identified with and tied to specific user organisations and thus not widely marketable (Bansler and Havn, 1996; Williams et al., 2005). Consequently, in the information systems literature, software package construction is conceived of as *design for markets* (Salzman and Rosenthal, 1994; Sawyer, 2000, 2001). Accordingly, it is said that programmers work without concrete notions of users in mind, a process Suchman describes as being akin to ‘design from nowhere’ (Suchman, 1994).[5] However, we are sceptical that complex organisational systems can be designed for abstract markets in an asocial manner. To explore this, we present material on the design and evolution of two software packages, and describe how suppliers actively manage users through configuring them within ‘communities’. In these groups, suppliers control which functionality and whose particularity will be accommodated through various forms of generification work. Before turning to the empirical material, we review how the literature on information systems has dealt with generic systems, and then we turn to relevant work within STS.

9.2 Narrative Biases in STS: Localisation

The nature of software development has changed in the last 30 years (Friedman and Cornford, 1989). Whereas user organisations once built or commissioned their own software, they now prefer to buy ‘commodified solutions’. Initially these were ‘low-level’ software systems (such as operating systems, utilities, and application tools), but increasingly they are also the ‘higher-level’ organisational information systems (such as payroll, procurement, and HR) and industry-specific systems such as those we are discussing (Brady et al., 1992; Quintas, 1994; Pollock et al., 2003). From the point of view of scholars sensitive to organisational diversity, this move is highly implausible, since software packages like ERP encompass a wide range of organisational activities which, because of their intricacy, are likely to vary from one organisation to another (Fincham et al., 1994, p. 283). In contrast, and buoyed up by the seeming success of these systems, proponents argue that they can be adapted to work in most organisations within the same class and, in principle, across different classes of organisations. In explicating these arguments, scholars point to the similarities that exist between organisations, as well as to the ‘flexibility’ of generic systems that allows them to be custom fitted to even the most idiosyncratic of settings (Davenport, 2000). As a rejoinder to these ‘universalistic’ presumptions, a large body of fine-grained empirical research has pointed to the difficulties adopters have with implementing them, as well as the large levels of unwanted organisational change they require – standardised systems may thus bring risks and unanticipated costs. The aim of much of this research has been to demonstrate that getting these systems to work is an ‘accomplishment’; an active process whereby users reconcile the gulf

[5] See also Hales (1994) for this view.

between system and actual work practices (McLaughlin et al., 1999).[6] If they can transfer between settings, it is only as a result of this major localised effort; they work because they have been redesigned around the cultures and practices of user organisations.[7]

In our view, the STS literature tends to overemphasise the collision between specific organisational practices and generic system presumptions at the point of implementation within specific user organisations (see, e.g. Walsham, 2001; Avgerou, 2002). This, we would argue, reflects the various narrative biases within current STS and sociology: that contexts of use are always individually different, unique, and typified by highly idiosyncratic practices; whereas technologies are 'singular' and 'monolithic'; and localisation is the means by which the standard and the unique are somehow brought together.[8] A further concern is that localisation studies do not adequately address the longer-term coevolution of artefacts and their social settings of use. This is not to say that we should view generic solutions as embodying features that can and should be applied in all contexts. We must also resist universalistic accounts and develop a language and set of concepts to describe how generic solutions are designed to pass over organisational, sectoral, and national boundaries, while embracing aspects of the specific features within these settings. In this respect, we argue that the notion of localisation, together with the concept of generification, can be taken further to explain this circulation. Our argument is not that the organisations in which the software circulates are the same; rather, it is that, through various generification strategies, these local sites *can be treated as the same*. How, then, are we to account for those times when the generic systems do actually travel across many contexts (Rolland and Monteiro, 2002)?

9.3 From Importing to Exporting

Ophir and Shapin (1991) asked a similar question some years ago in relation to scientific knowledge. This was a reaction to the 'localist turn' in the Sociology of Scientific Knowledge (SSK): scholars, sceptical of the claim that knowledge diffuses because it is 'true', sought to show how the universality of science was both an 'acquired quality' and 'local affair'. They did this by emphasising how facts were produced with reference to specific places and times, and that they were the product of particular communities, and that there were tacit practices involved in their produc-

[6] In their comparative study of IT systems, to give just one compelling example, McLaughlin et al. (1999) deploy a commonplace vocabulary to highlight how users actively 'appropriate' (MacKay and Gillespie, 1992), 'domesticate' (Sorensen, 1996), or 'work-around' (Gasser, 1986) the shortcomings of newly arrived technologies.

[7] An exemplary instance of this kind of writing is Avgerou's (2002) recent book.

[8] The concept of narrative bias invites us to reflect upon the repertoires of classic stories that particular schools of analysis often develop with characteristic contexts, problem diagnosis, dangers, and solutions (Williams et al., 2005). See also Woolgar and Cooper (1999) for a similar discussion of 'iconic exemplars' in STS.

tion (Knorr Cetina, 1981; Turnbull, 2000; Hanseth and Braa, 2001). Ophir and Shapin's (1991: p.15) question was 'If knowledge is such a "local product", then how does it manage to travel with such "unique efficiency"?' Others voiced similar questions at the time and this led to a growth in 'laboratory ethnographies' and an interest in demonstrating just how knowledge *escaped its locality*: this was the claim that knowledge only became universal after contextual features of locality or 'particularity' were *deleted*. Moreover, to 'solve' this problem of how knowledge moved from one laboratory to another, Latour (1987, 1999) introduced various terms such as 'immutable mobile' and, more recently, 'circulating reference'.

While these terms have become commonplace within the STS vocabulary, they also have been criticised. Firstly, much of the criticism objects to the overly imperialistic language used by Latour and other proponents of actor-network theory: 'immutability' seems to suggest that devices remain standardised at the centres at which they are produced, the locales at which they are used, and as they pass through the channels between these places. In particular, the notion of immutable mobile directs attention away from the localised work of adapting an inscription or innovation to a local context of use and setting up the conditions for its effective 'travel' (Knorr-Cetina and Amann, 1990).[9] Secondly, the terms are also criticised for implying that marks of locality are simply deleted. On the first point, and writing some years earlier, Ravetz (1972) had attempted to give a more sensitive treatment of the spread of knowledge by arguing, not for the immutability of scientific knowledge, but for its 'malleability'. Knowledge, tools, and instruments, he argued, were widely adopted through processes of 'smoothing'. That is, scientists importing methods or techniques from outside their normal domain would ignore any obscurities or unresolved conceptual difficulties surrounding that object.[10] In terms of the second point, Turnbull sought to build on Latour's work by showing how the local, rather than simply being erased, was often 'aggregated'. He illustrates this through a discussion of the way in which indigenous knowledges spread though a process of bridging:

> I argue that the common element in all knowledge systems is their localness, and their differences lie in the way that local knowledge is assembled through social strategies and technical devices for establishing equivalences and connections between otherwise heterogeneous and incompatible components. (Turnbull, 2000, p. 13)

In other words, local knowledge diffuses through the creation of 'similarities' and 'equivalences' between diverse sites. Such equivalence making requires a number of different devices and strategies, such as 'standardisation' and 'collective working', some of which we will explore further with empirical material.[11]

---

9 Thanks to Michael Lynch for framing this point in this way.

10 We are grateful here to Jamie Fleck for bringing this set of arguments to our attention.

11 We should also mention Timmermans and Berg's (1997) work as they have suggested that artefacts can be both universal and local at the same time. Putting forward the notion of the 'local universal', they argue that universals *do* exist but they emerge together with the local.

9.4 The Studies

We analyse two software packages which are at different stages in their 'biography' and characterised by different levels of product maturity and standardisation.[12] The first is a student administration system – the Campus Management module (CM) – developed by the German software house SAP, to integrate with its already highly successful ERP R/3 system. To develop CM the supplier had involved a number of universities as the 'surrogates' on which the software would be modelled before it would finally be launched to the wider market as a 'global university solution'. While SAP was new to the higher education sector, it has developed software for unfamiliar settings many times before. The second study is of the student accommodation system PAMS, which was built by a company we call 'Educational Systems'. PAMS was initially designed around the needs of one Scottish University but is now being used by over 40 other institutions in the UK, and the supplier is currently investigating the potential market overseas. PAMS has associated with it a growing and active 'user group' that meets regularly to learn about new product developments and petition for the building of further functionality. Whereas SAP already had in place established design methods and processes for software package design, Educational Systems did not; the latter company was new to both higher education and to the development of software packages.[13]

This is an important contribution. However, our interests are different in some respects. Their account is firmly on the side of work practice and the appropriation of a medical standard and how despite various 'local circumventions' and 'repairs' carried out by users of a particular protocol, the notion of 'one' standard still persists. Also, *local universal* is an analytical notion they invent to separate out the world of practice from the world of standards, and, then, to show how these worlds are reconciled with one another. Our concerns, in contrast, are with design practices and how actors *themselves* negotiate and establish the boundaries between what is particular and generic. And in this respect we view as sociologically interesting the way *suppliers* attempt to bring together and manage both of these aspects while building of a generic software package. Gieryn (1999) discusses a similar point in relation to the authority of science and how lay people understand what counts as good and bad science. It is important, he says, to focus on how actors perform this boundary work rather than privileging the analysts' view.

[12] For a more detailed discussion of the 'biography' of a software package see Pollock et al. (2003).

[13] The material presented here stems from observations (by NP) of what are sometimes called 'requirements prototyping' sessions (meetings in which suppliers demonstrate early versions of systems and elicit feedback), and user group meetings at the suppliers' premises. A number of semi-structured interviews and informal discussions were also conducted with supplier consultants, programmers, and users. Finally, one of the authors (NP) was commissioned to conduct a study on the suitability of launching PAMS abroad. Along with a co-researcher, Tasos Karadedos, NP met regularly with the management team to discuss strategies and potential markets. Material from this study is also presented here.

9.5 Birth of a Package

The 'birth' stages of the biography of a software package are the most dramatic. In this phase there are few users in place and the large community upon which the package will depend for its circulation is yet to be enrolled. Seemingly, there are many choices influencing the extent to which the package will become 'generic' and therefore attractive to the widest possible groups of users. Suppliers will spend time deciding which organisational practices will be catered for and which will not. In truth, however, and despite the seeming importance of this stage, the suppliers appeared initially to follow a strategy of simply and rapidly 'accumulating functionality'.

9.5.1 Accumulative Functionality[14]

Software packages are designed around a basic organisational functionality, what is sometimes described as the 'generic kernel'. The idea is to paint the organisational reality of adopters onto this kernel by developing numerous 'templates', which users can then choose between them and tailor to meet their local conditions. These templates form the 'outer layer' of the package, and are built up over time through interactions with past customers. Suppliers only reap benefits from developing new templates when they are able to use them again and again (thus recouping development costs). In the birth stages, both suppliers found that, rather than simply reusing templates, they were repeatedly forced to modify or build new ones. For instance, Educational Systems found that with each new customer for PAMS, the templates required modification. The sales director describes this in relation to the 'payment schedule' process:

> When we first wrote PAMS for [Scotia University] they produced a Payment Schedule that gave the student the choice of paying in 3 equal instalments (1 per term) or equal monthly instalments. The logic was therefore simple in that PAMS added up all of the charges and divided by the number of instalments.

However, when they made the next sale to 'Highbrow' university there were some differences which required changes to the software:

> The next customer, [Highbrow], also offered the choice of paying in termly instalments, but they massaged the amounts to take 40% in term 1, 40% in term 2, and 20% in term 3, as they wanted to get as much paid as possible before the student ran out of money. We therefore added a tick box on the payment plan to say 'use ratios', and this then gave access to an extra column that allows them to enter the % against each instalment.

He describes how they could accommodate the next user with the changes conducted for Highbrow: 'The next customer [Seaside] also produced a termly plan, but

[14] This discussion of Accumulative Functionality is partially drawn from Karadedos (2003).

used the number of days in each term to compute the amount. Fortunately, the work we had done for [Highbrow] was capable of managing this, as the days in each term could be entered as numbers as well as percentages'. But, once again, when another user adopted the package they were forced to make changes: '[Central] came along. And they offered students a discount if they paid by a certain date, so we had to add another (optional) column that stored the settlement date for each instalment and we added the code to compute the value of this discount'. The Sales Director goes on to describe the modifications required by two further universities: '[City], on the other hand, charges a penalty for late payments. So we added a process that calculated a charge for late payment'; And on the other hand, '[Rural] wanted this banded as their fees change according to the amount owed, so we added extra functions to band the charge according to the value'.

What is clear is that as each new site adopts the package, new and different requirements need to be catered for. Importantly, this occurs not simply in the payment schedule process but in all the other templates stored in the system library. The supplier appeared to be building into the system whatever functionality was asked for. However, it was becoming obvious to Educational Systems that accumulating and not reusing functionality was *particularising* PAMS. In the case of the payment schedule, for instance, every time a change was made to the template this would be accompanied by a modification to the graphical user interface. A user was then forced to view a screen which included buttons and menus specifically intended for other institutions. As a result, there was now a need for increased training where users were told which options and buttons related to them and which did not. However, this mode of redressing the particularisation of PAMS became problematic once the system was made available for operation by students over the Internet. One of the managers describes the problem:

> … how do you get rid of the things that a particular site doesn't want? For example, in our payment process we handle things like 'settlement discount'. Somewhere like [Welsh University] do not use settlement discount but they just ignore the fields on the screen. If you put that on the Web, all you do is end up with calls from customers, from students asking 'Why haven't I got any settlement discount?' When actually the answer is that 'We do not use it, so we do not want to display it'. So how do we get over that?

During the birth stage, then, suppliers are presented with choices. If they continue with the strategy of accumulative functionality, PAMS will become increasingly baroque, locked in to the particular requirements of their specific array of existing users. This realisation led to a switch in strategy. As the managing director of Educational Systems puts it, 'We are not going to accommodate as much diversity as we have in the past because it constrains our ability to grow and resell'. Any changes we make to the package from now on, he says, will have to have wider applicability: 'When we built change into the software we have always tried to build it in a way that isn't customer specific and we try to always broaden it a bit so that we have functionality that has a potentially wider audience'. During one particular conversation he described how they now try to 'discourage too much diversity'. Yet this presents the supplier with an interesting problem: How do they continue to make the software attractive to, and, indeed, encourage, a wider range of new users without having to include every demand for new functionality? Importantly, how do they ‘discourage too much diversity’ without discouraging the users attached to this diversity?

9.6 Management by Community

If the software is truly designed to travel, then it seems that the suppliers must avoid dealing with individual users. Indeed, the translation from a particular to a generic technology corresponds to a shift from a few isolated users to a larger extended ‘community’ (Cambrosio and Keating, 1995; de Laet and Mol, 2000). Moreover, it is through establishing and engaging with the users primarily through the kind of forum described above that suppliers are able to *shape* these communities and to extend the process of generification. In other words, through participating in community environments, such as the user-group meetings and requirement prototyping sessions, individual organisations were often dislodged from attachments to particular needs.[15]

9.6.1 Community Management Strategies

The suppliers had close ties with individual user organisations in the earlier phases, but they felt forced to shift to an alternative form of relationship as the technology matured and the user base grew. The openness of the software that was stressed during initial interactions was reversed: where they had previously negotiated on a one-to-one basis with users, they now appeared increasingly reluctant to differentiate users. Individual conversations about design issues were *shifted* to a more public forum. This shifting out is also demonstrated in the case of SAP, which had elaborate routines for managing its communities (and though the same strategies were visible within Educational Systems, they appeared much less developed). SAP had developed CM by gathering requirements during site visits and from other direct correspondence with users. The problem in accumulating functionality in this way was that they were ‘flooded with particular requests’.[16] How might they construct something more generic from these requests? Moreover, if they were to ‘discourage diversity’, how would users react if they felt their needs were not being met (and perhaps those of a neighbour were)? Thus, there was potential for this problem to become a focus of conflict (and the precious pilot sites on which the future of the product depended might be discouraged or, worse, lost).

[15] Here we loosely draw on Woolgar’s (1996) notion that a technology ‘performs’ a community. He uses the term in conjunction with the ‘technology as text’ metaphor to show how readers arrive at a preferred form of use. He suggests that within the technology/text certain identities and positions are offered with which the user can choose to align.

[16] This was taken from an email exchange between one of the pilot sites and the supplier. The author was discussing the danger of design that was focused on individual sites and not the community.

9.6.2 Witnessing

During the requirements prototyping sessions, a wide number of potential users were invited to the SAP University in Waldorf, Germany. The reported function of these meetings, which would last as long as 2 weeks, was to receive feedback on Beta versions of the software and to continue the requirements gathering process. It was the latter process that was the most striking. Participants from over a dozen universities and as many countries were seated in a room; each appeared determined to spell out in magnificent detail just how *their* particular requirements differed from the prototype on the screen in front of them, or, just as likely, from the view being articulated by their neighbour at the next desk. In the excerpt below, they discuss the storing of student transcripts and whether universities need to store details on both passed and failed courses. A consultant standing at the front attempts to make sense of the comments by scribbling them onto overhead projector (OHP) slides:

> **SAP consultant:** Does everyone want the ability to store two records?
> **America South Uni.:** We would maintain only one record …
> **SAP consultant:** Is there a need to go back into history? If transcript received and courses are missing do you need to store this?
> **America North Uni.:** … no record is needed.
> **America South Uni.:** We need both to update current record and then keep a history of that...
> **Belgium Uni.:** In our case, things are completely different …

This exchange points to the diversity of institutions present and the extent to which their requirements are similar or, at times, contradictory: where some users require one kind of record to be stored, others need a more comprehensive record, and one institution records things in a different manner altogether! Yet it is here that the supplier was finally able to observe the similarities and differences between institutions (and to begin to shape them in some way).

These meetings were also interesting for the way in which they appeared to shape the users' attitudes towards the overall generification process and their determination to have particular needs represented in the system. Through spending time getting to know the size and complexity of the task at hand, the participants appeared far more accommodating towards collective requirements, even to the extent that they would often compare institutional practices ('Oh! You do that…'). They had to concede that, even though it was a generic system, the supplier was determined to search for each and every difference between sites. No differences were ignored. No one group, or so it seemed, was explicitly favoured. Towards the end of one particularly long session, some of the users even began to suggest that the SAP was perhaps 'over determined' to find and articulate differences. The America South Uni. participant, for instance, described to the others sitting at his table during a coffee break how he thought SAP had 'too much patience' in allowing everyone present to spell out their particularities in such detail.[17] This comment was insightful in that it suggested an interesting shift in the provenance of the generification process and in who takes responsibility for it. Problems were seen to be the result of users, who were intent on describing their particular needs, while the supplier, who had actually gathered them together in this way, was guilty only of being ‘too patient’.

In summary, by shifting design from the level of the individual to that of the community, the supplier moved the software package from the private domain of each user site, where only particular needs could be articulated, to a public setting, where community or generic requirements could be forged. A further advantage of allowing users to participate collectively was that they were able to ‘witness’ the continued openness of the process. Indeed, somewhat ironically, some participants express concerns that it was not the supplier who was prolonging or complicating the generification process but the users who were doing it to themselves.

9.7 Management by Content

Whilst management by community revealed diversity, there was also a need to shape and smooth this diversity; to *manage through content* (Knorr Cetina, 1999).[18] There were two aspects to these strategies: firstly, to translate collective requirements into functionality that might be used by all of the sites present; and, secondly, because these sites were surrogates for potentially *all other* universities, to then translate the community functionality into a much more generic functionality. One method of establishing such templates was through searching for *similarities* between sites. These similarities did not emerge easily, but had to be pursued and actively constructed. Consequently, we think it is useful to describe this process in more detail, and so we focus on a discussion of ‘progression’ within the CM module.

Process Alignment

One consultant had asked participants to describe their rules for progressing students from 1 year to another, and to explain how a student’s grades contribute to her overall programme of study. A complicated conversation develops with various people interjecting. The consultant struggles to bring the discussion back on topic by attempting to summarise and name the particular process being described:

---

[17] Indeed, the participants were becoming increasingly frustrated by the supplier’s attempts to understand each and every difference among all the universities present and to reconcile these with the needs of the others present. For the suppliers, such a process appeared to be useful, as they saw it as a means by which the module might become *more generic* and thus potentially applicable to the widest variety of higher education institutions.

[18] Knorr Cetina develops the notion of ‘management by content’ to describe how people are managed especially through the content of their work as opposed to management through organisational structure or hierarchy (Knorr Cetina, 1999, p. 172).

SAP consultant: We've got one aspect now. Just want to get some common things. How [do] we name the baby? Let's go to the grading issue. Want to specify if module will contribute to Programme of Study in any way as a credit or grade. Is there any rule how it contributes? Is it linked to students? What is it linked to that it gives credit?

> **Swiss Uni.:** Could be a rule or a decision given by someone?
> **South African Uni.:** The student can still do the exam and be graded but it might be true that the grade or credit did or did not influence the student's progression ...
> **Canadian Uni.:** We wouldn't use these rules: we take all courses into progression. We have rules based on courses students take.
> **SAP consultant:** It is the same at [America North]. It is the US model. It is the difference between the European and the US model.

There are a number of interesting aspects in this exchange. When faced with diverging requirements, the establishment of generic features seems impossible. However the consultant does not admit defeat, but accepts the next best thing to a single generic process: 'two' generic templates. Moreover, she constructs these two templates by aligning or superimposing processes that are already roughly similar to one another ('It is the same at America North'). This then leads to the establishment of a generic feature ('It is the US model') which means that the requirements of a large group of universities is now seen to have been captured under one process. We also see in this exchange the naming of a further generic template, described as the 'European model', which emerges to capture all the differences that do not fit into the 'US model'. From now on, there will be two modes of progressing students within the CM module (meaning that they will adopt either the US or the European process). Drawing on Epstein (forthcoming) we might describe this as both the production of 'generalised differences' and a form of 'process alignment'. Finally, once these two categories were established, they were continually compared: both the supplier and the participants acted if it was self-evident that everything inside each of these processes was identical, and that anything or anyone outside of one classification could be easily accommodated in the other. Indeed, only one of the participants, a South African University, was from an institution outside the US or Europe. And since interactions during these meetings had shown them that they had many similarities with other users, particularly the British participants, they appeared to be happy to align themselves with the European model.[19] Process alignment appeared to be a successful method, with supplier representatives routinely framing their questions in ways that promoted this form of generification ('Does everyone want the ability to ...?' 'Does anybody else have this?').

[19] We later found out during the final stages of drafting this paper that the South African University eventually decided not to implement Campus Management. Their reasons, and the continuing evolution of CM, are the subject of continuing research.

Having an Issue Recognised

An interesting, though not altogether surprising, development was that the users began to learn that if they were to have their particular needs represented in the system then they too should engage in alignment work. An America South Uni. participant makes a case that the system should record grades for failed courses, and very quickly other users begin to give their support:

America South Uni.: We have concepts called ‘forgiveness’: a student retakes a course he’s not done well in and he is ‘forgiven’. The old grade is recorded but not included in the GPA [Grade Point Average].

> **Canada West Uni.:** We do the same thing. When we have symbols that aren’t graded – like ‘withdrawn’ or ‘incomplete’.
> **SAP Consultant:** This is a big issue for everyone …?
> **Canada West Uni.:** We definitely have to store it. These non-grade things don’t have a pass value or fail value; they are a ‘third’ value.
> **SAP Consultant:** I call it ‘additional module results’.

Here, then, an issue is recognised as generic through this accumulation of support. Moreover, the consultant appears happy to include the feature in the system since she is both able to name it (as ‘additional module results’) and establish an equivalence among the other institutions whose needs are catered for under this one concept.

9.7.1 The Organisationally Particular

It was common during these sessions to find requests that could not be made compatible across sites. Consequently, they had to be rejected or sifted from the process. The most common method for doing so was simply to categorise requirements as ‘specific’. For instance, during a discussion around the storing of surnames, an America East Uni. participant describes how they have a specific need to record maiden names after marriage. They suggest adding a new field to the screen (an Info_Type) but the consultant dismisses this as unworkable: ‘If we went for country-specific or customer-specific Info_Types now, then we could not utilise R/3 resources. The resources would be too great’. On this issue, unlike previous ones, the other universities do not align and thus it is not recorded on the acetate. The official reason for this was that the change would not link back to the generic system (and this meant that CM would no longer integrate with the ERP system of which it was a small part).[20] The suggestion instead is that America East should create a new Info_Type themselves when they customise the module back at their own institution. In other words, making the system fit America East’s needs is *postponed* and shifted onto the customisation stage at the user site (Hartswood et al., 2002, p. 28).

---

[20] There is an interesting issue here of how the universities were squeezed into existing software models that had nothing to do with higher education. We have explored this issue in Pollock and Cornford (2004).

9.7.2 Smoothing Strategies

Throughout these requirements-gathering sessions, many of the participants would go into great detail concerning their specific needs. The consultants would often use an interesting range of social strategies and devices to simplify and curtail particular requests, and we explore one such strategy with OHP slides (acetates).

Working the Acetate

In response to one lengthy description, the consultant used the physical limitations of the acetate to abbreviate a request ('Just trying to think how this can fit all on one line'). On other occasions, particular issues would be rejected for being already covered under existing themes. Pointing to the acetate, the consultants would say 'we had that issue already'; even when it was not always clear just how the new issue had been covered. Indeed the acetate was something of an 'obligatory point of passage' (a device or gateway through which the requirements needed to pass, see Callon, 1986); once scribbled down, an issue could be considered to have been recognised by the supplier, but, of course, it was far from easy to inscribe it on the acetate. The participants also recognised the importance of the acetates. In one discussion, the university representatives' sites are describing progression rules and an America South Uni. participant prefaces his intervention by stating that 'you'll need a new page'. While, of course, he is attempting to signal his university's uniqueness, the consultant dismisses this by pointing to the existing, well-annotated acetate and stating how there is 'one line left'. Later, when the America South participant appears to be about to list a further set of differences the consultant states that 'the page is full'. We would say that this *working of the acetate* was a particularly strong form of smoothing because it appeared as a simple material necessity and was thus not recognised as generification work.

9.7.3 From Generification to Generifiers

In the final stages of the CM project there was once again a notable shift concerning the shaping of the package and the locus of generification. Dragging the design from the private domain of direct-user engagement to a public setting had, apparently, been a drain on the supplier's resources, and the requirements prototyping meetings were no longer seen to be as 'productive' as they once had been. Below, one participant from a Belgium University writes in a report that:

> The current way of working with workshops is very labour intensive for the people of product management and development at SAP Waldorf. The biggest problem is that there is a very mixed public attending these workshops. Some of them already have a lot of expertise in CM and they see the workshops as a roll-in of requirements and for giving feedback after testing. For others this is their first experience with CM and they see it more as a kind of training. SAP wants to change this. In the future there will be standard training courses for larger groups. For roll-in activities there will be focus group meetings. These will only be attended by experts on the subject (limited groups of people) and they will focus on narrow subjects.

This shift was met with objections from users who stated a preference for collective engagement rather than the smaller group or individual interactions. While this appears somewhat counter-intuitive, the reason for the objections became clear some weeks later when one user reported that it was now increasingly common for their requests for functionality to be rejected. This was because it was said, by SAP, to be functionality required by only one university. In other words, because there were no longer community meetings, it now appeared difficult for the supplier to work out, and for the user to determine, what a generic need was and what was not. And it appeared that they had decided to assume that the majority of the requests did not represent generic needs. In order to prove that their needs were generic and not particular the universities had begun to search for similarities between themselves and the other sites (see Pollock and Cornford, 2004). In other words, once back in the private domain, the burden of generification was pushed onto the users. The participants had no choice but to become 'generifiers' themselves. If they did not fully participate in the generification process, if they were not good generifiers, their needs would not be effectively represented within the package. And it appeared to be better to have your needs represented in a generic format than not at all!

Management by Social Authority

The ability of a software package to become mobile is a result of the successful extension of a particularised application, and, at the same time, the extension of the community attached to that system. It is the latter aspect which is of interest; specifically how the process requires the enrolment and configuring of a user community that is subject to, and actively participates in, this generification process. However, the kind of work required in this form of ordering varies from the sophisticated smoothing/sifting strategies and boundary work described above, to what might be described as more direct 'social authority' strategies. This was particularly evident in later phases of the packages' development when the heterogeneous nature of the user base and the fact that it was beginning to swell with 'late comers' resulted in pressures to pull the packages in different directions.

9.7.4 Segmenting the User Base

The initial 'openness' of the package was a useful strategy for building the community by enrolling users into the design process. Now, in the later stages of the package biography, this openness was something of a drawback. As was evident in the quote from the Belgium University above, users were still expecting to have their particular requests met, and what was unsettling some of the established pilots was that the late comers were also making additional demands that might slow or complicate progress. This also occurred in the case of PAMS. The sales director describes how early on, when the company did not yet have a finished system, it had had to create an expectation among users that their specific needs would be met. It was now difficult to correct this view:

> … but, of course, it raises a level of expectation … you can be a year downstream in an implementation with somebody, and suddenly they throw up this

> requirement that has never been vocalised before, but because they bought as an early adopter they perceive that they have that type of relationship that means that you will do it for them. Even though they may well be the only people in the UK that actually want it!

Rather than simply refuse to cater for any kind of particular requirement, however, the supplier had segmented the community into three distinct categories: as either 'strategic', 'consultative', or 'transactional' customers. While these terms were part of the vernacular of the PAMS team, they were still thought to warrant some explanation by the managing director, when he mentioned them to us:

> ... [I]t is where we perceive it is worth putting the effort: Strategic Customers, Consultative Customers and Transactional Customers. Transactional customers don't want to spend money. They want everything for nothing. So for every day you put into them you get nothing back. So you put your days into Consultative customers who want to work with and spend with you. Whereas Strategic are all about people who help share the vision of where the product is going to go over the coming years.

From his point of view, strategic and consultative customers were central to the future development of PAMS whereas transactional customers were peripheral to its evolution. The former were regularly quizzed and consulted on the addition of new features and the general direction of the package, while the latter were actively kept at a distance. One example of how this strategy structured the users' interactions with the package was seen in the issue of 'customisation' and the question as to whether a user could modify the generic kernel.[21] During a conversation we had with a PAMS programmer, for instance, he praises a modification carried out by one early adopter and describes how this has even been fed back into the generic package for use at other sites: '[The London Uni] have done a fair bit … 80% of that has been incorporated into the standard package…They were willing to run ahead … they had the resources'. During the same conversation, he criticises another user for making a modification to the kernel and describes how it was explicitly stated that they are not allowed to make changes to the source code: 'We make sure that it's in the contract that they don't do things like that. We have had customers manipulating the data … from the back-end … Very dangerous … They promised not to do it again'. This suggested that the ability of a user to customise PAMs, and still have their system supported by the supplier, was directly related to the status they held at that time.

[21] Usually changes to the source code provide suppliers with something of a dilemma. On the one hand, modifications developed by users are an important source of innovation and are often fed back into the generic package for use at other sites. On the other hand, such evolution can be disruptive and if things go wrong during such modifications, this often leads to disputes about where responsibility rests for sorting things out. See Pollock (2005) for a lengthy discussion of this issue in relation to the authorised and unauthorised customisations and 'workarounds' conducted on standardised computer systems.

This, of course, begs the question as to just how a user might find themselves placed in one or another category.

Good Generifiers

Typically the status of a user was simply related to 'when' they adopted the system: the first group of users being closer to, and later-comers further from, the supplier. One other key criterion was related to how willing a user was to reshape practices to conform to the templates embodied within the system. The managing director of Educational Systems describes how:

> One of the other things we found about Consultative customers where they have entered into a dialogue with us is about how they might change how they do things. There is a lot of functionality in PAMS and there are areas where the universities aren't particularly efficient … So the Consultative customers are more willing to look at how they do their business and how they might improve their business based on suggestions for us based on existing functionality or commissioning us to add extra functionality.

Encouraging users to carry out organisational change to align with the system is an important strategy for managing the user base, and also a way to reduce the need for the further accumulation of particular functionality. It is a method, in other words, of moving users towards the 'organisationally generic'. Moreover, suppliers actively recruit customers who appear willing to engage in such change, and they reward them with greater access to the shaping process.[22]

In summary, Educational Systems does not have the large user base enjoyed by suppliers like SAP, and thus it has to be sophisticated in how it brings pressure to bear on users. We saw a form of selection where the supplier was prioritising which functionality might be allowed into the package. Users were divided into those who sought to align with the organisationally generic features being developed, often through conducting processes of change within their own organisations, and those who did not. The former group, as a reward for being 'good' surrogates, were actively involved in shaping the evolution of the package and were regularly consulted on which features they would like to see in the package. The latter, by contrast, were pushed to the margins of this shaping process, where they were not consulted or involved in design or evolution. Just what they could do with the system was policed (see Fig. 9.1).[23]

---

[22] Interestingly, we also routinely witnessed how a user might shift from one classification to another. The very first adopter of PAMS, for instance, was in the process of moving from the centre to the periphery (and there was even talk that it was now becoming 'transactional').

[23] This diagram is a development of one found in Karadedos (2003). Permission to reproduce it has been granted.

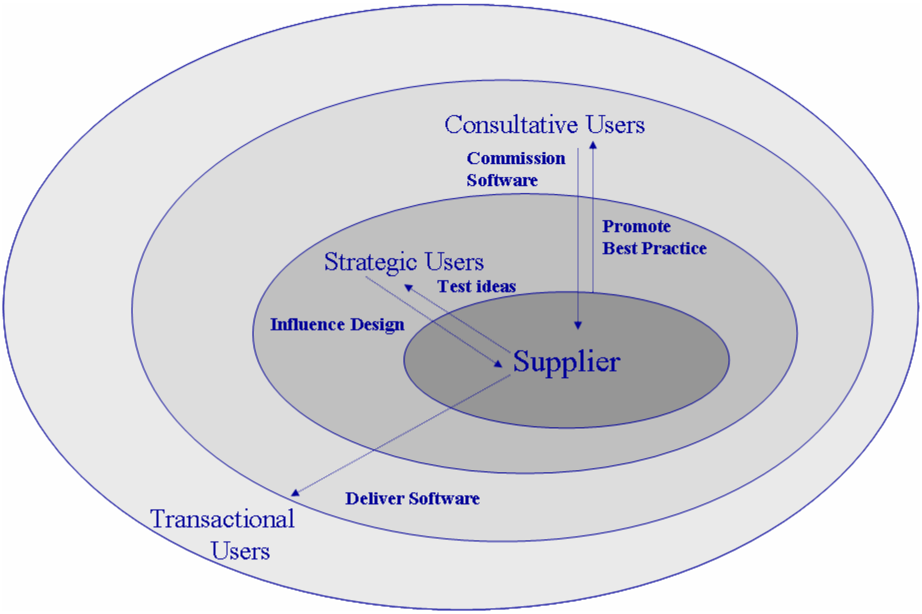

Fig. 9.1. Proximity of users to artefact

9.8. Promising Future?

We now delineate a final stage of the software packages biography: *the future*. The software packages' might be thought to have a promising future or 'career' ahead of them; promising because the effort to create a generic technology required moving towards maturity in order to escape particularisation. As a result there are still many places to which the software can travel. In its promotional literature, for instance, SAP boldly states how the CM module embodies 'no country specifics'. Yet, despite what this says, there were times when specific requirements appeared valuable for the circulation of the software package. Or, perhaps, it was simply impossible to avoid including the particular within the generic technologies being built.

Surrogate for Whom?

Some users were able to convince the suppliers that their needs had 'generic potential'. One criterion determining the ability of a user to get features embodied in the system revolved around the issue of just 'who' they were a surrogate for. The UK market was seen as a 'strong subsidiary' by SAP, meaning that the inclusion of a British university in the community might open up potential markets elsewhere. And as a result, the British university was able to wield significant influence. For instance, the supplier agreed to build the 'UCAS admissions link', a piece of functionality that would be a significant drain on resource and, importantly, one that could *not* be applied in other countries. During our research we began to learn that the CM module embodied many other particular features. One document describes how: 'In addition to generic functions, Campus Management also offers country-specific functions. These are functions that are only used in a particular country and cover needs arising from local legislation or business practices'. In other words, including particular functionality allowed the CM module to move not only within the same sector but also to *different* countries.

The case of Educational Systems raised a different issue as the addition of particular functionality offered PAMS the potential to move both into a new country and *across* an industrial sector. The supplier was considering whether to launch PAMS in the US and, of course, one issue of import was how well PAMS would fit with the peculiarities found there. One area where a difference was perceived was in how student rooms were allocated. Whereas UK students are simply assigned individual rooms, US students typically share a room and can therefore state their preferred type of room mate. The managing director described how this difference would require that 'social engineering' software be added to PAMS. Initially sceptical about the costs of such a development, he also saw how this might be useful for the evolution of PAMS:

> That is a piece of functionality that we could add-in and usefully use over here. So it may well be something we can use. One of the things we can certainly use is the ability to have multiple layouts in a room …. So we can build those changes into the software in a way that actually positively impacts on our ability to sell the software in the UK.

The addition of this 'social engineering' functionality would mean that PAMS would have more utility in existing UK universities *and* the private sector hotel industry, one area the Supplier had recently targeted. Their aim, in other words, was to identify where particular characteristics could have a more general appeal. We might describe particular features that aid the circulation of the package ('the UCAS admission link', the 'Social Engineering' etc.) as 'generic examples of the particular'.

Paths of Diversity

There were other forms of diversity included in the system. Earlier, we discussed the template for the 'progression' of students and how the consultant had developed not one but 'two' templates. This was interesting as it was one of the rare occasions when the supplier had to create 'multiple' templates for the same process – what we might describe as *polygeneric* templates. In their promotional literature, the supplier describes these polygeneric templates as giving the system extra flexibility through allowing adopters more choice:

> Progression – Depending on your particular environment, you may want to measure the progress of your students in different ways. One option is to determine the academic standing …. Another option is to evaluate a student's progress. … SAP Campus Management supports several progression methods thanks to our global approach to solution design. The flexibility of this appli-

cation allows an institution to change processes in the future without the need to install a new student information system.

By allowing polygeneric templates the supplier has created the basis for internally segmenting the user community, so that the templates allow users to follow different routes depending on their particular circumstances. They have, in other words, established 'paths of diversity' through which users might navigate. This was still a form of generification as the supplier was allowing users to choose between one of several large groupings. In this final section we consider what the inclusion of diversity and generality means for shaping the generic system and the community of users.

9.8.1 Opening the Black-Box (and Finding a 'Black-Blob')

We have shown how the generic system results from various kinds of boundary work. With the drawing and redrawing of borders the system embodies a range of features and potentially caters to a wide range of organisations (see Fig. 9.2).

Let us describe the system. The bulk of its features are the organisationally generic templates that suppliers attempt to build. These form the majority of the organisational 'outer layer', where suppliers hypothesise that organisations are similar and that the participating sites are good surrogates for all others in that class of organisation.[24] There are also compromises in which designers, unable to devise a single template, build in several templates to carry out broadly equivalent bundles of organisational processes. These 'polygeneric' features reflect the diversity of user organisation practices and contexts that cannot be readily captured within a single template. Finally, there are 'generic particulars', where idiosyncratic requirements are deemed to be important for aiding the future circulation of the package. These are only a few examples of how the generic and the particular are made to fit together. With further research, we would be able to generate further instances and a more complex picture. But our point should be clear: when examined closely, generic solutions are not the monolithic systems that much of the literature seems to suppose (see, e.g. Walsham, 2001 and Avgerou, 2002). Rather, they are the result of intricate boundary work involving generification (the creation of generic templates), the particularisation of the generic (the polygeneric templates), and, at times, the generification of the particular (the generic particular templates). In our view, the design and evolution of software packages are characterised by the working out of the relationship between the generic and the particular.[25] Indeed, this occurs not simply in design but throughout the lifetime of the software package.

[24] These are, of course, equivalences only in the *realm of design* and whether they emerge in the *realm of practice* will depend on other generification strategies.

[25] Indeed, the globalisation theorist Roland Robertson (1992, p. 102) has gone as far as to describe 'contemporary globalisation' as marked by a similar process or what he describes as the '…institutionalisation of the two-fold process involving the universalisation of particularism and the particularisation of universalisation'.

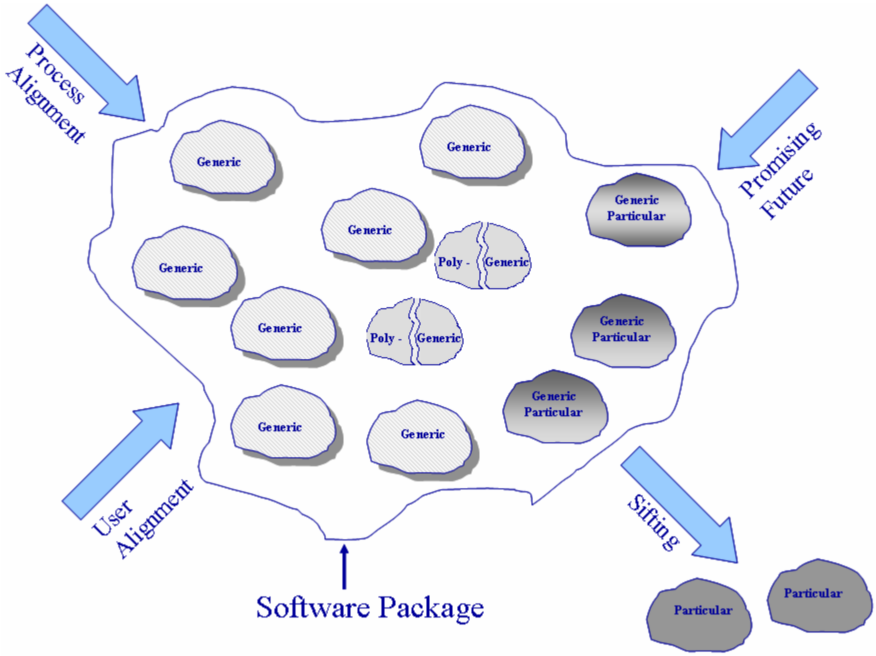

Fig. 9.2. Generic solution as a 'Black Blob'

During the research we thus began to recharacterise these generic solutions as 'black-blobs' (Michael, 2000). Within STS, technologies are commonly described as 'black-boxes' in order to emphasise how their form and function are stable, that prior processes of shaping are obscured, and that the user is configured into using the object in certain ways. By contrast, the software packages are also bounded objects, but their internal workings continually contort as they move around and then as new functionality is added. While the overall appearance of the software package (and in the case of the highly modularised packages like SAP, as its core 'kernel') may seem to remain intact, the addition of a new template, for example, causes the packages to morph and extend themselves in different directions. It is through this morphing/extension process that software packages are able to move from place to place, and to reach out into new settings. Such amoeboid movements, in turn, enable users to grab onto and then align themselves with the various protuberances and protrusions.

9.9 Conclusion: Black-Blobs Travel Better Than Black-Boxes

Certain software packages can be made to travel with 'unique efficiency', to borrow Shapin's (1998) description of scientific knowledge. In doing so, they unsettle prevailing core assumptions in the sociological understanding of organisational technologies. Put simply, much of the sociological and STS literature pays particular attention to the mismatch between system and actual work practices and emphasises the local adaptation necessary to bridge the gulf (McLaughlin et al., 1999; Walsham, 2001; Avgerou, 2002). While we do not downgrade the importance of this focus on how technologies are imported, we point instead to the need to go beyond studies of 'simple location' and also examine how systems are able to work *across* different organisational contexts and how they are exported. Rather than focus on the collision between unique organisational practices and the generic solution we should *also* address how technologies are made (and continuously remade) to bridge these different locales, as part of our enquiry into the broader and longer-term coevolution of artefacts and their social settings of use. We have argued that generic solutions *do* exist and that they do travel to many different places; though, of course, they don't go everywhere. They arise through the broader extension of a particularised software application and, at the same time, the management of the user community attached to that solution.

We noted some interrelated moments in the biography of these solutions. There was a distinct *birth stage* at which suppliers designed specific user requirements into the software. This was followed by a number of delimited responses in the subsequent maturation of the package, when the suppliers attempted to move away from the simple accumulation of particular functionality. One interesting aspect was the shift to capture collective rather than individual requirements in order to establish organisationally generic features through alignment and smoothing practices. Such practices helped establish greater compatibility across sites, as equivalencies were established in organisational practices, and differences were worked together and generalised. Suppliers attempted to align processes that were already roughly similar, what we called 'process alignment work'. The collective gathering of requirements also had a secondary consequence of shifting expectations about the kinds of need that would be met by the system. Through 'witnessing' the level of user diversity, and realising that the only way to represent needs was to engage in the process, the users' conceptions of their own needs shifted in a way that aligned with those of other participants. In other words, users were in some respects self-governing concerning the articulation of their level of particularity and generality. This raises questions about which users have the capacity to extend and broaden a template: On what grounds and by which methods?

To summarise, it is not just sociologists of science and technology who are interested in the relations between the particular and the generic, and how the boundary between them is established, managed and shifted (O'Connell, 1993). Software packages are a high-value industrial product, necessitating extensive interactions between suppliers and users. Building software packages calls for suppliers to develop and sustain sophisticated strategies for managing diversity, and setting boundaries and priorities for dealing with their market of user organisations. User organisations similarly need to learn how to respond to and interact with such strategies. As communities grow and inevitably encompass a wider range of organisational types and requirements, this user base also needs to be organised if the supplier is to avoid being confronted with a potentially overwhelming array of requirements. This, as we have shown, involves different kinds of boundary work: in terms of understandings of which types of organisations lay 'close to' and which 'further from' the supplier's conception of the ideal type of user; and in terms of the willingness of the supplier to accept or sift particular requests from users. The 'black-box' view of the generic solution where it simply 'invades' and 'disciplines' is too crude. What we have shown is that establishing a generic solution is a precarious achievement of various kinds of generification strategies. These are strategies in which the suppliers and users of software packages constantly work towards a pragmatic resolution of the tension between the generic and particular. As a result of this generification work, software packages can circulate and user communities can grow; that is to say, diverse organisations and standard technologies *can* be brought together.

Acknowledgements

We acknowledge the support of the UK Economic & Social Research Council (ESRC) who funded the research project ('The Biography and Evolution of Standardised Software') on which this paper was based. We warmly thank all those people at the software supplier organisations and the user communities who contributed to the paper in various ways. We acknowledge the contribution of Tasos Karadedos who accompanied us during interviews at Educational Systems. His assistance and final dissertation were very helpful in preparing this chapter. Also thanks to Jamie Fleck, Alex Voss, Christian Koch, Geoffrey Bowker, Barbera Czarniawska, Dave Stearns, Sampsa Hyysalo, Mei Wang, Christine Grimm, Wendy Faulkner, Michael Lynch, and the 'Writing Circle' at Edinburgh University, who all provided useful comments and suggestions on early drafts.

References

Avgerou, C. (2002). *Information Systems and Diversity*. Oxford: Oxford University Press.

Bansler, J. and Havn, E. (1996). Industrialised Information Systems Development, *CTI Working Paper* No 22. Technical University of Denmark.

Berg, M. (1997). *Rationalizing Medical Work: Decision-Support Techniques and Medical Practices*. Cambridge, MA: MIT Press.

Brady, T., Tierney, M. and Williams, R. (1992). The Commodification of Industry Applications Software, *Industrial and Corporate Change,* v. 3, n. 1, pp. 489–514.

Callon, M. (1986). The Sociology of an Actor-Network: The Case of the Electric Vehicle. In Michel Callon, John Law and Arie Rip (eds.), *Mapping the Dynamics of Science and Technology.* London: MacMillan, pp. 19–34.

Cambrosio, A. and Keating, P. (1995). *Exquisite Specificity: The Monoclonal Antibody Revolution.* New York: Oxford University Press.

Cooper, R. (1998). Assemblage Notes, in Robert Chia, (ed.), *Organized Worlds: Explorations in Technology and Organization with Robert Cooper*. London: Routledge, pp. 108–129.

Davenport, T. (2000). *Mission Critical: Realising the Promise of Enterprise Systems*. Boston: Harvard Business School Press.

de Laet, M. and Mol, A. (2000). The Zimbabwe Bush Pump: Mechanics of a Fluid Technology, *Social Studies of Science,* v. 30, n. 2, pp. 225–263.

Epstein, S. (forthcoming) Institutionalizing the New Politics of Difference in U.S. Biomedical Research: Thinking across the Science/State/Society Divides. In Scott Frickel and Kelly

Moore (eds.), *The New Political Sociology of Science: Institutions, Networks and Power*. Madison: University of Wisconsin Press.

Errington, F. and Gewertz, D. (2001). On the Generification of Culture: From Blow Fish to Melanesian, *The Journal of the Royal Anthropological Institute*, v. 7, n. 3, pp. 509–525.

Fincham, R., Fleck, J., Procter, R., Scarbrough, H., Tierney and Williams, R. (1994). *Expertise and Innovation*. Oxford: Claredon Press.

Friedman, A. and Cornford, D. (1989). *Computer Systems Development: History, Organization and Implementation*. Chichester: John Wiley.

Gasser, L. (1986). The Integration of Computing and Routine Work, *ACM Transactions on Office Information Systems*. v. 4, n. 3, pp. 205–225.

Gieryn, T. (1999). *Cultural Boundaries of Science: Credibility on the Line*. Chicago: University of Chicago Press.

Hales, M. (1994). Where Are Designers? Styles of Design Practice, Objects of Design and Views of Users in CSCW. In Duska Rosenberg & Chris Hutchinson (eds.), *Design Issues in CSCW*. London: Springer-Verlag, pp. 151–178.

Hanseth, O. and Braa, K. (2001). Hunting for the Treasure at the End of the Rainbow: Standardising Corporate IT Infrastructure, *Computer Supported Cooperative Work (CSCW),* v. 10, pp. 261–292.

Hartswood, M., Procter, R., Slack, R., Voss, A., Buscher, M., Rouncefield, M. and Rouchy, P. (2002). Towards a Principled Synthesis of Ethnomethodology and Participatory Design, *Scandinavian Journal of Information Systems,* v. 14, n. 2, pp. 9–30.

Karadedos, T. (2003) The Biography of a Software Package, Unpublished MSc Dissertation, University of Edinburgh.

Knorr Cetina, K. (1981). The Manufacture of Knowledge: An Essay on the Constructivist and Contextual Nature of Science, Oxford: Pergamon Press.

Knorr Cetina, K. (1999). *Epistemic Cultures: How the Sciences Make Knowledge*. Cambridge, MA: Harvard University Press.

Knorr Cetina, K. and Amann, K. (1990). Image Dissection in Natural Scientific Enquiry, *Science, Technology & Human Values*. v. 15, n. 3, pp. 259–283.

Latour, B. (1987). *Science in Action: How to Follow Scientists and Engineers Through Society*. Cambridge, MA: Harvard University Press.

Latour, B. (1999). *Pandora's Hope: Essays on the Reality of Science Studies*. Cambridge, MA: Harvard University Press.

MacKay, H. and Gillespie, G. (1992). Extending the Social Shaping of Technology Approach: Ideology and Appropriation. *Social Studies of Science,* v. 22, pp. 685–716.

McLaughlin, J., Rosen, P., Skinner, D. and Webster, A. (1999). *Valuing Technology: Organizations, Culture and Change*. London: Routledge.

Michael, M. (2000). *Reconnecting Culture, Technology and Nature*. London: Routledge.

Muscatello, J., Small, M. and Chen, I. (2003). Implementing Enterprise Resource Planning (ERP) Systems in Small and Midsize Manufacturing Firms, *International Journal of Operations & Production Management,* v. 23 n.7–8, pp. 850–866.

North Whitehead, A. (1967). *Science and the Modern World* New York: The Free Press.

O'Connell, Joseph (1993) Metrology: The Creation of Universality by the Circulation of Particulars, *Social Studies of Science,* v. 23, pp. 129–173.

Ophir, A. and Shapin, S. (1991). The Place of Knowledge: A Methodological Survey, *Science in Context,* n. 4, pp 3–21.

Pollock, N. (2005). When is a Work-around? Conflict and Stuggle in Computer Systems Development, *Science, Technology & Human Values,* v. 30, n. 4, pp. 496–514.

Pollock, N. and Cornford, J. (2004). ERP Systems and the University as a "Unique Organization", *Information Technology & People,* v. 17, n. 1, pp. 31–52.

Pollock, N., Williams, R. and Procter, R. (2003). Fitting Standard Software Packages to Non-Standard Organizations: The Biography of an Enterprise-wide System, *Technology Analysis & Strategic Management,* v. 15, n. 3, pp. 317–332.

Quintas, P. (1994). Programmed Innovation? Trajectories of Change in Software Development, *Information Technology & People,* v. 7, n. 1, pp. 25–47.

Ravetz, J. (1972). *Scientific Knowledge and its Social Problems*. Oxford: Oxford University Press.

Robertson, R. (1992). *Globalisation; Social Theory and Global Culture*. London: Sage.

Rolland, K. and Monteiro, E. (2002). Balancing the Local and the Global in Infrastructural Information Systems, *The Information Society*, v. 18, n. 2, pp. 87–100.

Salzman, H. and Rosenthal, S. (1994). *Software by Design: Shaping Technology and the Workplace*. Oxford: Oxford University Press.

Sawyer, S. (2000). Packaged Software: Implications of the Differences from Custom Approaches to Software Development, *European Journal of Information Systems,* v. 9, pp. 47–58.

Sawyer, S. (2001). A Market-Based Perspective on Information Systems Development, *Communications of the ACM,* November, v. 44, n. 11, pp. 97–101.

Shapin, S. (1998) Placing the View From Nowhere: Historical and Sociological Problems in the Location of Science, *Transactions of the Institute of British Geographers,* v. 23, n. 1, pp. 5–12.

Sorensen, K. (1996). Learning Technology, Constructing Culture: Socio-technical Change as Social Learning, *STS Working Paper* No18/96, Centre for Technology & Society, Trondheim, Norway.

Star, S.L. and Ruhleder, K. (1996). Steps Toward an Ecology of Infrastructure: Design and Access for Large Information Spaces, *Information Systems Research,* v. 7, n. 1, pp. 111–134.

Suchman, L. (1994). Working Relations of Technology Production and Use *Computer Supported Cooperative Work (CSCW),* v. 2, n. 30, pp. 21–39.

Timmermans, S. and Berg, M. (1997). Standardization in Action: Achieving Local Universality Through Medical Protocols, *Social Studies of Science*, v. 27, pp. 273–305.

Turnbull, D. (2000). *Masons, Tricksters and Cartographers*. London: Routledge.

Walsham, G. (2001). *Making a World of Difference: IT in a Global Context Chicester*. New York: Wiley.

Webster, J. and Williams, R. (1993). Mismatch and Tension: Standard Packages and Non-standard Users in Paul Quintas (ed.) *Social Dimensions of Systems Engineering*. Hemel Hempstead: Ellis Horwood, pp. 179–196.

Williams, R., Stewart, J. and Slack, R. (2005). *Experimenting with Information and Communication Technologies: Social Learning in Technological Innovation*. Cheltenham: Edward Elgar.

Woolgar, S. (1996). Technologies as Cultural Artefacts, in William Dutton (ed.), *Information & Communication Technologies: Visions & Realities*. Oxford: Oxford University Press. pp. 87–102.

Woolgar, S. and Cooper, G. (1999). Do Artefacts Have Ambivalence: Moses' Bridges, Winner's Bridges and Other Urban Legends in STS, *Social Studies of Science*. v. 29, n. 3, pp. 433– 449.

Chapter 10

Concluding Remarks

Rob Procter[1], Mark Hartswood[2], Alex Voss[1], Roger Slack[3], Mark Rouncefield[4], and Monika Büscher[5]

[1] National Centre for e-Social Science, University of Manchester, rob.procter@manchester.ed.ac.uk, alex.voss@ncess.ac.uk
[2] School of Informatics, University of Edinburgh, mjh@inf.ed.ac.uk
[3] School of Social Sciences, University of Wales at Bangor, r.slack@bangor.ac.uk
[4] Computing Department, University of Lancaster, m.rouncefield@lancaster.ac.uk
[5] Department of Sociology, University of Lancaster, m.buscher@lancaster.ac.uk

10.1 Introduction

In the face of increasingly powerful and complex ICTs, growing integration and interdependencies between services and organisations, and multifaceted, complex work processes, effective user–designer relations are critical if our expectations of ICTs are to be realised in timely, economic, usable, and dependable ways. Users and designers must seek to work more closely and productively together if the transformative potential of new technologies in use is to be realised. In doing so, however, both users and designers have to face the practical realities of the settings in which they operate, realities that have as much to do with their political and commercial contexts, and the aspirations of the organisations creating software products, as they do with the capacity of a given intervention to deliver effective design recommendations.

The chapters in this volume describe and reflect upon different ways of managing user–designer relations and the ways in which new challenges may call for us, as PD practitioners to move beyond current approaches and invent new ones. In drawing on the practical lessons, its contributors have gleaned from their involvement in a range of projects of different types, we hope that this volume will provide a resource for users and designers practicing or considering participatory design as an approach.

As we argued in the Chapter 1, we are not looking for a remedy to the 'problem' of user–designer relations. This would be a futile objective, partly because of the specificity of the problems to contexts and, not least, because the backdrop to user–designer relations is itself constantly shifting. Changes to working patterns and working arrangements, inter- and intra-organisational modes of working, modes of technology supply, and the emergence of new technologies with concomitant possibilities for transforming working practices and home life (and new ways of making a profit or a living) provide for a continually shifting terrain for user–designer relations to populate and adapt to. That is not to say, though, that where the case studies in this

A. Voss et al. (eds.), *Configuring User-Designer Relations*,
DOI: 10.1007/978-1-84628-925-5_10, © Springer-Verlag London Limited 2009

volume highlight difficulties with realising appropriate user–designer relations to achieve a particular end (as perspicuously explored in Jenkings' and in Martin et al.'s studies of the UK NHS Connecting for Health programme) or where approaches or remedies are suggested, that these are pointless exercises immediately dated by shifting circumstances and practice. Rather, we would draw attention to Hyysalo's point in Chapter 6 that particular modes of user–designer relations tend to be taken up, dropped, and reinstated, sometimes over the various phases of a single product's evolution. With this in mind, we hope that the case studies in this volume might provide PD practitioners with an antidote to the tendency noted by Hyysalo for organisational forgetfulness of the benefits of user engagement.

In this final chapter, we will attempt to summarise the important insights arising from the chapters presented in this volume, be frank about what is missing, and to formulate questions for use, design, and research. We will begin by exploring how we might categorise and compare the kinds of user–designer encounters reflected in the chapters.

10.2 A Taxonomy of PD Practices Revisited

In the years since Bowers (1991) first pointed to some of the complexities involved in ICT systems design, the 'Janus face' that necessarily presents itself differently to users and designers, continuing research and hard won experience have highlighted not just the difficulties and intricacies involved in attending to any one 'face', of understanding either the problems of design or the problems of use, but the necessity and the sheer difficulty of addressing these issues simultaneously. Perhaps the most important message for readers to take away from this volume is that there is no single, best approach to effective user–designer relations. This is not a straightforward message, nor is it easily implemented, since we have now moved way beyond simple notions of 'configuring the user' (Woolgar, 1991). As Törpel et al.'s review of PD in Chapter 2 has shown the meaning and practice of PD has evolved as its adherents have responded to new opportunities and challenges. PD might now be best described as an 'eclectic mix' of techniques from which practitioners may choose. And, as each subsequent chapter has demonstrated, in choosing, PD practitioners must be sensitive to how the various factors such as scale, novelty, organisational type, etc., that we might use to define and distinguish projects might play out in the context of any given project and adapt their approach to suit, not only when deciding how to begin but also as a project unfolds. This also invites us to reflect on the relevance of different kinds of expertise and knowledge, how user engagement changes over the course of a project, the impact of different modes of technology supply (from bespoke software solutions to generic packages), the maturity of the technologies, and the need to orient to commercial and political objectives and the pressures these exert on organisational members, whether users or designers.

Adapting and expanding the taxonomy of PD practices devised by Muller et al. (1993), we attempt below to map the case studies in this volume along a number of dimensions which seem to us, on the basis of the evidence, to have a significant bearing on their outcomes. These dimensions reflect, *inter alia*, the kinds of user engagement that the case studies embody, the design issues they implicate, and aspects of the socio-economic circumstances in which they are set.

10.2.1 Context of Engagement

Drawing upon the first of the two principle dimensions of Muller et al.'s original taxonomy ('who participates with whom in what?'), our aim here is to unpack some of the complexity associated with the notion of context of engagement. First, we can distinguish between, on the one hand, designers entering the users' setting and, on the other hand, users entering the setting of the designer. The case studies in this book document examples of each: *user setting* (Jenkings; Büscher et al.; Martin et al.; Voss et al.); *designer setting* (Bonner; Hyysalo); as well as ones which span both (Hyysalo; Pollock and Williams). With the exception of the approach advocated by Voss et al. in Chapter 3, the setting for engagement seems to correlate strongly with the timing of engagement within the software lifecycle. In particular, with the growing recognition of the value of ethnography, engagement in the users' setting has become an increasingly preferred mode for the early phases of requirements gathering. What is striking, then, about the approach advocated by Voss et al. is its goal of making the users' setting the primary context for engagement throughout the development cycle.

Second, as in Muller et al.'s original taxonomy, we can distinguish between commercial product development and research projects, the significance of that the latter being that it may more accommodating of 'experimental' approaches to user engagement. The majority of case studies in this volume fall into the *commercial product development* category (Bonner; Hyysalo; Pollock and Williams; Martin et al.; Voss et al.), with only one case study (Büscher et al.) representing *research projects*. Finally, we can distinguish between the following types of application setting: *organisational working environments* (Jenkings; Pollock and Williams; Martin et al.; Voss et al.); and one which we will refer to as *domestic environments* (Bonner; Hyysalo), whose growing significance is a reflection of the expanding market for – and ubiquity of – ICT products and services.

10.2.2 Timing of Engagement

The second of the two principle dimensions of Muller et al.'s taxonomy ('time during the development lifecycle'), the timing of engagement within the system lifecycle is important for the ways in which it may influence the role of users and their capacity to shape the system being produced. Broadly speaking, Muller et al.'s typology divided this dimension into three periods – early, middle, and late. In our revised taxonomy, we can equate these with, respectively: *requirements gathering* (when users have the opportunity to be involved in shaping system fundamentals, as in Bonner; Büscher et al.; Voss et al.); *development* (as the system begins to take shape and users have an opportunity to provide feedback and refine it before it is completed, as in Hyysalo; Büscher et al.; Voss et al.); *implementation*, or 'localisation' (as the finished system is deployed and users have an opportunity to influence how it is adapted to their needs, as in Martin et al.; Pollock and Williams; Voss et al.) phases of a project. To these, however, we would add *deployment* (i.e., where users have the opportunity to influence how the system evolves, as in Voss et al.) and which has typically been ignored as a site of user–designer relations the PD community (as it has also been by software engineering which relegates it to the unglamorous category of 'maintenance'). The case studies in this book have illustrated how different techniques for facilitating user–designer relations come into their own during the software lifecycle. This questions notions of a status of 'completion' of technological innovation, and the separation between design, implementation, and appropriation. In our experience, it is unusual to find user engagement being pursued with equal commitment throughout the software lifecycle and this, we feel, is a challenge that the user–designer research community must address.

10.2.3 Scale of Engagement

One of a small number of secondary dimensions in Muller et al.'s taxonomy, scale of engagement is important for its practical impact on user-engagement techniques. We find examples in this book of *small scale* (Bonner; Büscher et al.; Voss et al.); *medium scale* (Martin et al.; Hyysalo; Voss et al.); and *large scale* (Jenkings; Pollock and Williams) user engagement, together with clear evidence of the influence of scale of engagement on the suitability of different techniques. As we noted in Chapter 1, many of the techniques most closely associated with PD do not scale well, if at all, and this situation has not changed significantly since Muller et al. devised their taxonomy. It is worth noting, however, that even though (as in the NHS project in the chapter by Jenkings) the scale of a ICT project may be huge when measured by the size of the overall user population, implementation (as illustrated by Martin et al.'s chapter) is nevertheless often a local exercise – and hence potentially much more manageable and amenable to meaningful user engagement. Perhaps one lesson for those responsible for the execution of large-scale projects is to adopt, where possible, technical strategies that maximise the scope for locally satisfying user requirements while remaining consistent with the overall project aims (e.g. Eason, 2007).

10.2.4 Purpose of Engagement

The case studies in this volume provide evidence of a diversity of purposes for user engagement. We can find examples of *building consensus* (Jenkings; Hyysalo (PDMS); Pollock and Williams); *informing design* (Bonner; Büscher et al.; Voss et al.); *informing development* (Hyysalo; Büscher et al.; Voss et al.); and *informing implementation* (Martin et al.; Voss et al.). These categories might seem to be similar to those enumerated under timing of engagement. We argue, however, that it is worthwhile distinguishing between timing and purpose in order to remind ourselves of the potential (indeed, very likely) asymmetries in the influence exerted by the different stakeholders in ICT projects (see Törpel et al., Chapter 2). Perhaps the most common and significant example of how these asymmetries make themselves felt is through the ways in which ICT projects are commissioned. Hence, the timing of engagement is very likely to be dictated by its purpose. Part of the reality of being an 'ordinary' organisational user is to be excluded from the making of strategy. ICT projects may therefore come burdened with a set of requirements over which users and designers have little or no control and must, in their different ways, then make the best of. For designers, in particular, this may be no easy task as they attempt to translate the project vision into an achievable end-product, and deliver it to schedule and on budget. As a consequence, for the designer, the choice of user engagement approach needs to keep a balance between improving the quality of the outcome while ensuring that its impact on the project remains manageable. We saw in Chapter 2 how the MUST approach to PD addresses this issue by splitting the project into a 'design' and an 'implementation' project, where the design project is prior to commissioning of the full system and informs a decision to build or buy, etc.

10.2.5 User Experience

The rapid expansion of the market for ICT products and services over the past 15 years has had a dramatic impact on the experience of being a 'user' and hence, we would argue, on the likely character of the user–designer relationship. In the case studies, we can discern the following kinds of user experience: *organisational users* (Martin et al.; Jenkings; Pollock and Williams; Voss et al.), reflecting those users who are subject to the dictates of organisational policies about the timing and form of technical change – but with the power, nevertheless, to influence how it takes shape 'on the ground', if given the opportunity; *discretionary users* (Bonner; Hyysalo (Wristcare)), reflecting users who have a choice over whether or not they will use a particular system or product and when (or perhaps which of several offerings they will choose); and, finally, *early adopters* (Büscher et al.), users who see themselves as innovators and who possess a strong vision of what they wish to accomplish and so are prepared to invest significant time and effort in order to drive a project from beginning to end. One lesson we would draw from this is simply that designers should be prepared for users' commitment to engagement to reflect their experience of being a user, and to adapt their expectations and approaches accordingly. For example, and as borne out by the findings of Martin et al. and Büscher et al. respectively, it would be surprising if organisational users would show the enthusiasm for – and hence be prepared to invest the degree of effort in – engagement that would be typical of early adopters. This, in turn, must influence which kinds of techniques for user engagement are likely to be successful. For example, to our minds, maximising the opportunities for organisational users to engage with ICT system design necessitates new forms of – longitudinal – engagement and a willingness to learn what matters to users in and as a part of their everyday work.

10.2.6 Summary

Our purpose in elaborating this taxonomy has been to attempt to identify and to categorize important dimensions of user engagement. Of course, any attempt at mapping user–designer relations in this way must be subject to a number of qualifications. First and most importantly, as with taxonomies generally, the categories we have chosen are a compromise, a necessary but nonetheless crude approximation to the messy realities of projects. Second, their selection, informed as it is by the case studies in this volume, may be criticised for being incomplete, and they may be subject to change (not least, as the markets for ICT products and services continue to coevolve with innovations in the technology itself). Part of the problem is perhaps, as Voss et al. argue, that unlike other software project activities, approaches to user–designer relations are hard to codify. In any case, it should be clear that such a taxonomy can only go so far in helping the PD practitioner answer the question of what kind of user engagement would be appropriate. With this in mind, we now turn to summarising some of the important themes which we feel the case studies reflect.

10.3 A Collaborative Endeavour

A question to which anyone with an interest in user–designer relations (as a user or designer) needs to have an answer is 'how much user engagement is sufficient to get the job done'? For many years, it would seem that the answer has been 'more than is being practised at present' which, of course, has been grist to the mill of the growing number of academic researchers working in this field. From the early 1990s onwards, the ICT design research literature has steadily accumulated evidence of how studies of work practice and the reflections on professional experience of work provide rich input to design. For many researchers, it has made an irrefutable argument for the 'turn to the social' (Grudin, 1990; Hughes et al., 1994) and, for some, as Törpel et al., note in Chapter 2, it pointed also to gaps in PD practices of that time which needed to be addressed. Some of these, we argue, persist today and this message is reinforced in several of the other contributions in this volume.

Büscher et al.'s chapter emphasises the limits of 'pre-prepared' design in ICT applications where it is impossible to anticipate users' requirements in each and every situation. Their message is that creative design and technological potential may lead to the discovery of an excess of possibilities, more than could ever be explored by conventional PD approaches. What Büscher et al.'s chapter confirms is that new kinds of ICT applications are likely to challenge existing techniques for, and assumptions about, PD, a point echoed by Lin et al. (2008) in their study of user engagement in the context of a Web 2.0 project. It seems to us that the only possible response from the PD community must be to continue to be innovative in its practices. Such innovation is required, especially in the face of challenges of achieving effective engagement with users in, for example, domestic settings, challenges which have risen up the agenda of both academic researchers and of commercial ICT product developers in recent years (Crabtree and Rodden, 2004). Another significant example of a gap in current PD practices is its failure to take an interest in deployment and use as a significant locus for user–designer relations and for the redesign and even reinvention of ICTs (Williams et al., 2005) It is a concern to address this deficiency, which is woven through Voss et al.'s chapter, and its argument that designers should be prepared for a 'long-term engagement' with users (Hartswood et al., 2002). As Voss et al. note, the specificities and possibilities raised through co-realisation (Hartswood et al., 2007) can exceed anything that (a priori) design can achieve, but it too has its limitations. The case for acknowledging the 'unknowability' of the future and for creating circumstances that are conducive to its emergence – and even for fostering potentially contrary visions – are at the core of our work. However, we must also remember that this has to be done in a world of finite resource. Design choices are, then, thoroughgoingly pragmatic and situated.

If design is a matter of making informed choices, it is often also a matter of 'satisficing'. In Chapter 3, Voss et al. have argued for taking on the challenge of meshing ethnography with ICT system design. One important message from their work is that practitioners of this approach to PD must grapple with the problems of matching the rich and detailed findings with what a project's technical staff can accommodate, that is, what they are practically able to do at any given time. This reminds us that, for very practical reasons, PD is a collaborative endeavour, a partnership between users and designers and not simply, as some more naïve interpretations of the craft might have it, a matter of putting users 'in charge'.

It is all too easy for academic researchers to criticise commercial approaches to user–designer relations for falling short of some sort of ideal where the process is valid, rigorous, where maximum information is extracted, and the best possible design produced, but this would be to ignore the realities of commercial contexts where the goal is to produce a competitive design using a process that has a high predictability and low overheads. In Bonner's chapter, we can see how some sort of alignment is negotiated between these two discourses, which is similar in some respects to the sorts of strategies outlined by Pollock and Williams (Chapter 9) in keeping different user constituencies aligned with each other and the company's generic offering. In a similar vein, Hyysalo in Chapter 6 observes how the shutting down by ProWellness of its engagement with its potential user base was due to the problems the company was experiencing with 'managing the cacophony of opinions'. The lesson from these chapters is that the modes of user–designer relations become shaped to meet business objectives (and not necessarily in a way that is always detrimental to end-users) that involves a (sometimes considerable) element of managing users and their demands.

We would argue that management is a feature of all user–designer relations, but that, on some occasions and in some settings, it is more pronounced than at others. It is interesting that academic offerings have increasingly focused on an ideal of accessing the skills of the worker in order to inform better (work-affording) designs, and moved away from earlier emancipatory ideals, as Törpel et al. explain in Chapter 2. The emphasis has shifted to how to get at this information, rather than on helping provide effective strategies for users and designers to manage better the demands and expectations that each might have of the other and this, we argue, is an imbalance that needs to be addressed in future work. In some senses, the rise of what we might call a 'pragmatist' participatory design has resulted in the attenuation of some of the 'idealism' that, to our minds, was a hallmark of early PD. We suggest that those seeking to deploy PD in the future might shift the balance back. This is another of the messages of co-realisation.

10.4 User Engagement in the Wild

It is difficult to translate 'utopian' and research-based approaches to user–designer relations into the prevailing socio-economic context and to related 'ideological' understandings of what 'design' and 'use' are, what 'technologies' are and what their relationship is to existing practices. This is, as the chapter by Törpel et al. makes very clear, an issue with which PD has grappled over many years. The chapters by Bonner, Hyysalo, Pollock and Williams, Martin et al. and Jenkings put this issue into a contemporary context and, to some extent, provide some benchmarks as to what can be achieved, as well as some strategies for survival, while being quite honest about the nature of the challenges and the limitations they impose on PD practices. They cover a range of contexts and so help us to be sensitised to the different kinds of tensions that a user or designer might be faced with, and why some user–designer partnerships flourish 'in the wild' (Dittrich et al., 2002), while others fail.

The chapter by Bonner tells a story of how designers in a commercial setting appropriate and refashion user-engagement strategies that have their origins in academia. We learn from Bonner that the 'organisational survival' of PD techniques requires that practitioners be alert to the contours of organisational power and be skilled in playing to organisational agendas. In the commercial world, modes of user–designer relations (and resources they consume) are evaluated against the competitive advantage they might confer, rather against their validity or rigour. The chapter by Hyysalo reinforces this point and helps to provide some arguments for the use of PD. His study reveals a shifting emphasis between different sorts of user–designer relations depending on commercial objectives and, crucially, the 'forgetting' of the benefits of previous sorts of engagements. The conclusion can only be that users and designers need to be prepared not only to make the case for PD but also to keep on making it over and over again – often in the face of more managerialist demands. It follows, then, that part of our message is that reconfiguring user–designer relations is the education of those who seek to manage contingencies out of development.

Regarding the use of specific techniques for PD, both Bonner and Jenkings expose how the ability to deliver an effective PD workshop depends on the skill of the facilitator and the commitment of the participants as much as the props used and the format of the exercise. Such skills may turn on fairly local and specific knowledge: how to deal with this designer or this customer, a point which emphasises again the benefits of the 'long engagement'.

A notable feature of Hyysalo's study was the generic nature of the software product. This was also a feature of the case study featured in Pollock and Williams' chapter, but differences in the scale of both the systems and organisations involved led to quite different approaches to user–designer relations. In the latter case, the conduct of user–designer relations is not left to chance and there is certainly no evidence of organisational 'forgetting'. The company management knows the stakes are too high for a haphazard approach and their response is designed to ensure users' requirements are contained within an envelope of what is practical for a generic, high value, and complex software product.

The case study in the chapter by Jenkings matches that of Pollock and Williams in terms of the scale of the end-product but is more challenging still because of its massive potential user base. It also exemplifies some of the problems of user engagement in politically driven, public sector ICT projects. In some respects, the solution Jenkings offers is a technological repackaging and updating of a well-known and well-used PD device: the mock-up or demonstrator (Kyng, 1995). The Animator is a tool to help NHS staff imagine the future and its implications, and not just technical ones but also organisational. As Jenkings explains, the Animator evolved from a concrete depiction of an EHR system in use into a much more conceptually oriented device designed to inform users about changes and solicit their ‘buy in’ for the system, rather than provide an opportunity for meaningful discussion about the details of its implementation. As Jenkings concedes, engagement with users over the organisational vision is quite different from engagement for the purposes of shaping the concrete details of its design and implementation and should be seen as a supplement to other methods and not a substitute for them. The Animator was therefore unlikely to have an impact on aspects of the system, such as user interfaces, workflow, etc, but it certainly could have shaped the ‘broader’, higher-level aspects of the programme if the political will had been there to let it do so. Indeed, it is striking that the issues that were raised in the Animator focus groups were prescient of the sorts of problems that have since bedevilled the NHS CfH programme. Jenkings’ case study exposes the boundaries of user engagement in a large-scale, bureaucratic organisation where end-users are separated by several layers of management from the key decision-making processes. Finally, and once again, Jenkings’ case study also demonstrates the influence of ‘organisational politics’ on the survival of an otherwise successful exercise in user engagement.

The chapter by Martin et al. is of an EPR project team that wants to engage users but discovers that finding the users is hard, and the degree to which they can engage with them constrained by competing technical concerns. In a sense, this case study picks up where those of both Jenkings, and Pollock and Williams left off and illustrates how end-users must grapple in the implementation phase with the consequences of previous user engagement (or the lack of it). In Martin et al.’s case study, the project team constantly battles with the inevitable contingencies of real organisational life and, no matter how committed the project team is to meaningful user engagement, this has to compete with the need to keep the project ‘on track’. We might think of this as an example of ‘bounded’ user engagement – there is not only a need for user engagement but also a need to deliver. There is clearly scope here, however, for some rethinking of the mechanisms employed to support effective user–designer relations and make them more able to withstand these organisational contingencies, which include overcoming various sorts of barriers to user involvement. Martin et al. suggest the use of small-scale ethnographies that are timed and targeted, informed through a process of tracking where user participation within a project is succeeding or failing. We will pick up this issue again in the next and final section.

10.5 Taking User–Designer Relations Forward

As with any attempt to cover such a wide area as user–designer relations, we are all too aware of the various omissions and the disparity of treatments that appear in this volume. The first omission we would highlight concerns our underemphasis on product design, the way design processes may be seen from within a company and the concomitant issues of getting user–designer relations and usability taken seriously or put onto organisational agendas that are often dominated by cost–benefit rationalities. Such analyses are available elsewhere (Lindholm et al., 2003), though, certainly, some of the observations made therein, for example, ‘our experience has been that the intersection of user needs and the industry interests takes place only *after* product launch’ are reflected in the experiences recounted in this volume. Moreover, many of the insights developed in this volume are relevant to a product-design process that involves the challenge of developing in tandem the physical as well as the digital experience with products. This co-development presents a dilemma in current practice because physical design and interaction design involve different knowledge, skills, tools, timelines, and work approaches, and consequently, user–designer relations of various kinds become paramount.

Another obvious omission concerns the relative absence of investigations into user–designer relations in everyday domestic settings. Bonner’s chapter does look at design for the home in that it is concerned with the development of novel cooker interfaces. Similarly, Hyysalo’s study concerned technologies to be used in homes – but of a distinct kind (managed care settings). In both cases, the approaches fall short of the sorts of recommendations made by Edwards and Grinter (2001) in that involvement of users is still at ‘arms length’ – there is no real commitment by either company to really get ‘under the skin’ of home life. For example, in Bonner’s case study, users are brought into the designers’ setting for workshops but the designers themselves never set foot in a ‘real’ kitchen as a part of their explorations.

Again, many of the conclusions and recommendations outlined here would be supported in the domestic sphere as designers in such sensitive settings increasingly turn towards methods that bring them closer to users’ aspirations and lives as really lived. This enables them to meet what Edwards and Grinter (2001) identify as the major challenges for designers: ‘to pay heed to the stable and compelling routines of the home […] subtle, complex and ill-articulated. Only by grounding our designs in such realities of the home will we have a better chance to minimise, or at least predict, the effects of our technologies’. Attention to user–designer relations in these settings brings design closer to the ideal of ‘inclusive design’, which emphasises the importance of social, human factors in system use. In this view, designers recognise that solutions devised on the basis of inappropriate investigative strategies and techniques can be debilitating and disempowering. Consequently, traditional technological approaches need to be complemented by detailed investigations into everyday life and user needs, involving the users themselves in the process of investigation and requirements specification as a feature of co-development or ‘co-realisation’ (Hartswood et al., 2007).

Looking beyond these omissions, it seems inevitable that continuing innovations in ICT products and services will set new challenges for users and designers and that meeting these will require the community to continue to push the boundaries of existing approaches and, perhaps, to come up with new ones. Practitioners grappling with user–designer relations will need to master the trick of being ‘agile’, in other words, of being able to shift strategies to adapt to changing circumstances, including technical change – from the perspective both of the demands it makes on existing user-engagement practices and opportunities it affords for experimenting with new ones – without having to ‘reinvent the wheel’ each time, while also being open to devising new forms and mechanisms for user engagement. Practitioners must be able to tap into the organisation's own experiences of the conduct and practice of user–designer relations, as well as those approaches and methods emerging from academic communities, both now and in the future.

At the same time, however, we would also argue that there are limits to what being agile can achieve for taking the practice of user–designer relations forward. In particular, practitioners must also ask themselves whether they should be content to operate within the established parameters for the conduct of user–designer relations as set by the context in which they work and, if not, the question must be what can be done about this. There is, then, as Voss et al. have argued, a need to create the space and time for co-realisation. What this can be is, in part, an achievement in the setting – but it is also a matter of principle. Taking the reconfiguration of user–designer relations seriously involves a commitment from all parties.

The relationship between design interventions and the political, commercial, and organisation context in which they are embedded have been recurrent themes throughout all of the chapters in this volume, which highlights not only the importance of skill and responsiveness to circumstances but also makes a strong case for some wider engagement by practitioners in shaping organisational and social agendas, that is, to be prepared to make the case for improving user–designer relations higher up within organisational decision-making structures. For example, techniques or approaches to user–designer relations that appear overtly to be exploratory or open ended, such as ethnographically informed design, are perhaps always going to find it hard to gain mainstream commercial acceptance because they do not provide the sorts of predictability demanded by the calculus of project management and this is despite the evidence that the risks of applying these approaches are arguably less than the risks to the project of not doing so (Shapiro, 2005; Kanstrup and Bertlesen, 2006). From an ICT project management perspective, taking user engagement seriously can seem to be expensive in terms of resources, may seem unpredictable in terms of producing well-defined outputs to a schedule, challenging in that it can privilege 'shop floor' rather than management concerns and, perhaps, even mobilise resistance to the rationale for undertaking a project in the first place. On the other hand, not engaging users can lead to systems that do not fit with or seek to build upon existing practice, may turn out to be unworkable and may even be rejected by users.

Perhaps, however, the case studies in this volume paint an overly pessimistic picture, in that the project management strategies they documented are not representative of some of the more promising innovations taking place in this field, including those which have clearly been influenced by the challenges of user engagement and promise a more responsive management regime for its practice. In particular, so-called 'agile' approaches have become quite popular as an answer to the rigidities of traditional, strictly phased methodologies (Beck et al., 2001; Lin et al., 2008). Key elements of agile approaches include a focus on working code, involving early-release and short-release cycles and an incremental-planning approach that allows changes to be made according to evolving circumstances and user requirements (Beck, 1998; 2000).

To us, it seems inevitable that something has to give if user–designer relations are to progress. We argue that, as PD practitioners, we have a choice: should we be content with making different sorts of trade-offs or compromises (such as following the suggestion of Martin et al. above) in order to do what can best be done in the circumstances or should we seek to influence those circumstances? Accepting the latter would require us to take seriously the objective of convincing organisational decision makers of the need to accept new priorities in the commissioning and conduct of ICT projects, ones that make fostering effective user–designer relations a priority from the start of a project and would make it easier to defend them in the face of all the usual project contingencies.

There are some promising signs that useful strategies are beginning to emerge, which may yet be capable of bringing about a shift in priorities from those that privilege technical considerations to ones that treat user engagement seriously. For example, drawing on lessons from the NHS CfH programme (and which looks set to be a rich source of arguments for those who argue for the importance of taking user–designer relations more seriously), Eason (2007) has advocated a socio-technical focus to the introduction of ICT systems in healthcare where the pace of implementation is geared more strongly to users' capacity to accommodate and contribute meaningfully to a programme of change and innovation, supported by a technical approach that emphasises building upon what is already in place with discrete and phased implementations. Perhaps, most significantly, Eason has argued for the establishing of a 'socio-technical committee' as part of a project's management structure, with a remit to facilitate the embedding of a broad series of socio-technical concerns. Similarly, in the MUST approach (Bødker et al., 2004) described in Törpel et al.'s chapter, user engagement is woven into the fabric of a project much more strongly than is the case in many other approaches, bringing it to bear on the structure of a project rather than treating it as an add-on or a challenge as is often the case in other approaches.

With this thought in mind, it seems appropriate that we conclude by echoing Suchman (2002) and urge that practitioners of user–designer relations heed the ambition '… that system developers become responsible for locating themselves within the extended networks of sociomaterial relations and forms of work that constitute technical systems.' That is, we should challenge the existing and taken-for-granted divisions of labour in the development of ICTs and the separation that currently prevails between the contexts for their development and subsequent use. It seems to us that this is a way forward and possibly the difference between user engagement being pursued as a means to 'fix things up', as opposed to dealing with problems in the first place.

References

Beck, K. (1998). Extreme Programming: A Humanistic Discipline of Software Development. In Astesiano, E. (Ed.) *Proceedings of the 1st International Conference on Fundamental Approaches to Software Engineering, Lecture Notes in Computer Science 1382*, New York: Springer. pp. 1–6.

Beck, K. (2000). Extreme Programming Explained: Embracing Change. Addison Wesley.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J. and Thomas, D. (2001). Manifesto for Agile Software Development. URL: http://agilemanifesto.org/

Bowers, J. (1991). The Janus Faces of Design: Some Critical Questions for CSCW. In Bowers, J. and Benford, S. (Eds.) *Studies in Computer Supported Cooperative Work*, North Holland: Amsterdam.

Bødker, K., Kensing, F. and Simonsen, J. (2004). *Participatory IT Design. Designing for Business and Workplace Realities*. Cambridge, MA: MIT Press.

Crabtree, A. and Rodden, T. (2004). Domestic Routines and Design for the Home. *Computer Supported Cooperative Work* 13(2), April. pp. 191–220.

Dittrich, Y., Eriksen, S. and Hansson, C. (2002). PD in the wild: Evolving practices of design in use. In *Proceedings for the Participatory Design Conference*, Malmo, pp. 124–134.

Eason, K. (2007). Local sociotechnical system development in the NHS National Programme for Information Technology. *Journal of Information Technology*. v. 22 pp. 257–264.

Edwards, K. and Grinter, R. (2001). At home with ubiquitous computing. In *Proceedings of UbiComp*. pp. 256–272. Heidelberg: Springer-Verlag.

Grudin, J. (1990). The Computer Reaches Out: The historical continuity of the interface. In *Proceedings of the ACM conference on Human Factors in Computing Systems (CHI)*, Seattle, Washington, pp. 261–268.

Hartswood, M., Procter, R., Rouncefield, M, Slack, R., Soutter, J. and Voss, A. (2002). The Benefits of a Long Engagement: Some Critical Comments on Contextual Design. In *Proceedings of the 2nd Nordic Conference on Human-Computer Interaction*, Aarhus, October 21st–23rd.

Hartswood, M., Procter, R., Rouncefield, M, Slack, R., Voss, A., Büscher, M. and Rouchy, P. (2007). Co-realisation: Evolving IT Artefacts by Design. In Ackerman, M., Halverson, C., Erickson, T. and Kellogg, W. (Eds.) *Resources, Co-Evolution and Artefacts*, New York: Springer, pp. 59–94.

Hughes, J., King, V., Rodden, T. and Andersen, H. (1994). Moving out of the control room: Ethnography in systems design. In *Proceedings of the ACM conference on Computer-Supported Cooperative Work (CSCW)*, Chapel Hill, North Carolina, 429–439.

Kanstrup, M. and Bertlesen, P. (2006). Participatory IT-Support. Proceedings of the ninth Participatory Design Conference, Trento, Italy.

Kyng, M. (1995). Making representations work. *Special issue of Communications of the ACM*, v. 38(9), pp. 46–55.

Lin, Y., Poschen, M., Procter, R., Voss, A., Goble, C., Bhagat, J., De Roure, D., Cruickshank, D. and Rouncefield, M. (2008). Agile Management: Strategies for Developing a Social Networking Site for Scientists. *Proceedings of 4th International Conference on e-Social Science*, Manchester, UK, 18–20 June.

Lindholn, C., Keinonen, T. and Kiljander, H. (2003). *Mobile Usability: How Nokia Changed The Face of the Mobile Phone*, New York: McGraw-Hill.

Muller, M.J., Wildman, D.M. and White, E.A. (1993). Taxonomy of PD Practices: A Brief Practitioner's Guide. Muller, M. and Kuhn, S. (Eds.) Special Issue on Participatory Design, *Communications of the ACM*, v. 36(4), June.

Shapiro, D. (2005). Participatory design: the will to succeed. Proceedings of the 4th decennial conference on Critical computing: between sense and sensibility table of contents, pp. 29–38.

Suchman, L. (2002). Located accountabilities in technology production. Scandinavian Journal of Information Systems, v. 14(2), pp. 91–105.

Williams, R., Slack, R.S. and Stewart, J. (2005). *Social Learning in Technological Innovation: Experimenting with Information and Communication Technologies*. Aldershot: Edward Elgar.

Woolgar, S. (1991). Configuring the User: The Case of Usability Trials. In J. Law (Ed.), A Sociology of Monsters: Essays on Power, Technology, and Domination. New York: Routledge.

Index