

Scandinavian Journal of Information Systems

Volume 21 | Issue 1

Article 2

2009

Inter-Contextual Distributed Participatory Design

Hartmut Obendorf

C1 WPS, Germany, hartmut.obendorf@c1-wps.de

Monique Janneck

University of Hamburg, monique.janneck@uni-hamburg.de

Matthias Finck

effective webwork, Germany, matthias.finck@effective-webwork.de

Follow this and additional works at: <http://aisel.aisnet.org/sjis>

Recommended Citation

Obendorf, Hartmut; Janneck, Monique; and Finck, Matthias (2009) "Inter-Contextual Distributed Participatory Design," *Scandinavian Journal of Information Systems*: Vol. 21 : Iss. 1 , Article 2.

Available at: <http://aisel.aisnet.org/sjis/vol21/iss1/2>

This material is brought to you by the Journals at AIS Electronic Library (AISeL). It has been accepted for inclusion in Scandinavian Journal of Information Systems by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Inter-Contextual Distributed Participatory Design

Communicating design philosophy and enriching user experience

Hartmut Obendorf
C1 WPS, Germany
hartmut.obendorf@c1-wps.de

Monique Janneck
University of Hamburg, Germany
monique.janneck@uni-hamburg.de

Matthias Finck
effective webwork, Germany
matthias.finck@effective-webwork.de

Abstract. Most studies of participatory design examine the development of a single, customized software system that supports typical workflows within a single client organization. To cope with other use contexts and new forms of work – such as communities and virtual networks – the traditional repertoire of PD methodology needs to be expanded to deal with distribution and diversification of users. Based on a ten-year case study, we describe experiences with PD in the development of a groupware system that initially targeted a single use context, but was continually extended and adapted to new contexts of use with new requirements and work practices. To enable distributed participation across contexts, new methods had to be established: inter-contextual user workshops bring users from different contexts and developers together to reflect on the usage and design of the software and its further development. Commented case studies make this face-to-face interaction persistent, providing a written documentation of distributed use experiences and design decisions. In the process of building an inter-contextual community of users, the PD focus shifts from custom software development to empowering users in assessing their own practice and technology use.

Key words: Distributed participatory design, community of practice, community of interest, community building, workshop, case studies.

1 Introduction

Participatory design (PD) has traditionally focused on the quality aspect of software development (e.g., Bjerknæs & Bratteteig 1994): as users are considered to be the foremost experts for their work, their involvement in the development process yields better requirements specifications, and result in better system design and more usable software. The developed software must be adapted to the task, not otherwise.

Another motivation for participatory design originates in its background in the European labor movement following the goals of humanization and democratization of work (Czyzewski et al. 1990; see also Braa and Vidgen 1995 for a historical synopsis); the ‘empowerment’ of workers exemplified by the construction of tools enabling users to change their own work environment (cf., Elovaara et al. 2006; Karasti and Syrjänen 2004; Herrmann et al. 2004; DePaula 2004; Irestig et al. 2004). Users develop skills and acquire tools that empower them to decide how to change their work practice.

However, PD’s established focus on large organizations and ‘workers’—and on the development of a single, contiguous, customized software system representing and supporting typical workflows within one organization—limits the applicability of PD methods. While very successful in bringing together different stakeholders and involving them in design, they need to be extended to cope with new forms of work. New organizational structures, such as ‘virtual networks,’ are difficult to include in development using traditional PD methodology as organizational structures are missing and boundaries between stakeholders become more fluid (cf., Finck and Janneck 2005; Janneck et al. 2006).

The traditional PD approach includes many methods that depend on the possibility for immediate negotiation; it is thus difficult to follow when co-location and shared work contexts are no longer given for participating users. *Distributed participatory design* (DPD) needs to cope with settings where stakeholders are *distributed* across various dimensions of time, space, and/or organization (Gumm 2006; Fischer 2004; Franssila and Pehkonen 2006; D’Andrea et al. 2008; Barcellini et al. 2008).

In this paper, we report on experiences with user participation in a long-term Open Source development project. The Open Source software, CommSy, is a web-based groupware system developed to support communication and coordination in working and learning groups by facilitating the exchange of documents and the sharing of important notes and dates between users, comparable to, e.g., BSCW, phpBB, or Moodle (Schümmer and Lukosch 2007; Stefanov et al. 2005; Cole and Foster 2007). The development of CommSy was initiated in an academic working group that experimented with groupware technology for their own needs. Over the years, use of CommSy spread rapidly to other departments and universities, and—most recently—secondary schools and vocational communities. The participatory development process had to deal with changing user requirements and also with less and less familiar and increasingly distributed contexts of use—in addition to CommSy becoming the base for a commercial venture (cf., Kensing 2000). The software was used in new work environments and for new work practices, and the amount of users and application domains increased continuously. Users as well as designers became more and more distributed, both organizationally and physically. This challenged and changed the PD process that was started when the user group was still small and

development was confined to a single location (cf., Gumm et al. 2006). Today, the CommSy development process shows typical characteristics of a DPD process regarding the temporal, physical and especially organizational dimension of distribution (cf., Gumm 2006).

In CommSy's ten project years, the development team continually applied participatory design methods. However, while today's methods still openly display their PD roots, the intention for involving users has shifted from the traditional PD paradigm of designing and tailoring custom software to enabling reflection and appropriation of software by building a community that is able to communicate and negotiate even as the user base broadens and the application contexts diversify.

Instead of developing a single application for tasks and work practice within a single organization, one application needed to meet the needs of distributed groups of users without losing the focus on 'empowerment'. Much as in the related PD work for designing for communities (e.g., Braa 1995; Bødker et al. 1995; Henderson and Kyng 1991; Korpela et al. 1998; Merkel et al. 2004; Karasti and Syrjänen 2004), 'empowerment' of users means to encourage them to reflect on their own tasks and work processes and to analyze organizational structure and processes. As several very different communities of practice (Wenger 1998) who all used CommSy were targeted, another layer was added to analysis: the reflection of the work of users in completely different contexts, creating an awareness for the requirements of others—a process we termed *inter-contextual community building*.

As a consequence, methods for building a community of interest (Fischer 2001) were developed – people linked not by common practice in their respective work contexts, but rather by a shared interest, manifested here in the use of the same piece of software (cf., Janneck and Finck 2006), across several communities of practice. Thus, inter-contextual participatory design is a form of distributed participatory design focusing on users who are not part of a single community of practice (examples for distributed CoPs are described, e.g., in Amaury et al. 2006; D'Andrea et al. 2008) but distributed across several organizational contexts—with spatial and temporal distribution playing an additional role.

In the following sections, we will describe experiences and results based on our case study exploring the distributed participation in the development of CommSy. Throughout the CommSy development process, system development and research on system development and design have been intertwined in the sense of an action research approach (cf., Braa and Vidgen 1995; Checkland 1991; Mathiassen 1998; Vidgen and Braa 1997); researchers embodied multiple roles as system developers, evaluators, and support staff for users, always in close contact with their clients to make the system fit their needs better. Two of the authors of this paper were directly involved in system development; the third was not member of the development team, but helped to analyze the project practices. Thus, in this paper we combine an inside view of the development process with an external analysis and reflection to answer the following research questions:

- How can PD approaches be carried out in a distributed setting, when one system is developed for and within several different contexts of use?
- Which traditional PD methods prove difficult in distributed context and what new problems arise, respectively?

- What kind of new approaches are necessary to carry out participatory design processes in distributed settings?

The paper is structured as follows: In section 2, we will introduce the case study in detail. In section 3, we sum up the resulting challenges and introduce the new techniques that were developed to meet these challenges. We reflect on different levels of (D)PD and corresponding techniques in section 4 and finally sketch perspectives for the future development of both the methodology and (D)PD as a discipline in section 5.

2 Case study: Inter-contextual groupware development

CommSy is an open source web-based groupware system. In contrast to other CSCW systems, community support was always a top priority, hence the name that stands for community system. It supports communication (e.g., with discussion forums and a bulletin board) and the exchange of working materials (with, e.g., file uploads and a group editor) as well as the organization of work (aided by a shared calendar, to-do lists, etc.). CommSy was originally introduced as an educational tool for the post-secondary level which was used in a variety of fields including History, Language studies, Education, Economics, and Informatics. Today, CommSy is no longer bound to the university: It is used in the public as well as private sector, ranging from industry to secondary schools. A recent poll of the main hosting site resulted in more than 60,000 users (Table 1).

<i>Use context</i>	<i>University</i>	<i>Secondary schools</i>	<i>Industry</i>	<i>Other</i>
<i>No. of users</i>	32,800	29,500	2,700	500

Table 1: Number of users in different contexts using the CommSy installation hosted by the primary application service provider, effective webwork GmbH (April 2009).

Two key features of CommSy are:

- *Project Workspaces* designed for closed groups of approximately 10 to 30 members (e.g., student groups), offering typical groupware functionalities for asynchronous communication, project management, and information storage and retrieval.
- *Community Workspaces* incorporate project workspaces into a larger structure supporting not only small groups but a community of users (e.g., all members of a school or university) over a period of time, similar to an intranet structure (cf., Pape et al. 2002).

The user interface design of CommSy is based on abstract design principles which communicate the design philosophy behind the development process: (1) Ease of use, (2) socially translucent cooperation (cf., Erickson et al. 2002), and (3) simple integration into existing technical infrastructures are the current design principles describing the shared vision and development goals.

CommSy—with the system originating in a University context—has always been the subject matter of various research projects. The development of CommSy started in May 1999 at the Department of Informatics at the University of Hamburg. Initially, the system was designed to support the communication and coordination in learning groups in the department. About half of the initial development team were non-programming domain experts; this founded their explicit view of “users as the experts – the ones with the most knowledge about what they do and what they need—and the designers as technical consultants” (cf., Schuler and Namioka 1993, p. xiii).

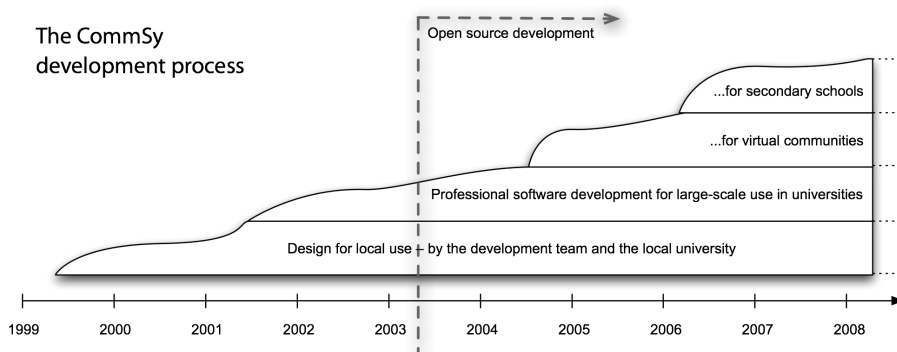


Figure 1: The CommSy development process

In the spring of 2001, some of the CommSy developers joined a research project dealing with computer supported cooperative learning (CSCL) and participatory software development (cf., Pape et al. 2002). As a consequence, the CommSy development was no longer a voluntary, unpaid activity, but grew into a professional software development project, following the STEPS (Software Technology for Evolutionary and Participative System Development) software engineering framework (cf., Floyd et al. 1989). STEPS is a methodological framework for evolutionary development in the PD tradition, stressing the importance of close cooperation with users as well as an extensive use of prototyping to facilitate the embedding of software systems in meaningful work processes in the user organization.

As a result of the three-year research project, CommSy had evolved into a sophisticated groupware tool that had been extensively field-tested and refined in the context of university education. As the CSCL research project ended in 2003, development of CommSy continued in form of an Open Source project. However, hosting and support services could no longer be offered free of charge for the universities using CommSy, as this had been the case while development was funded by the research project. Instead, a spin-off company began to provide application services (ASP) for anyone wishing to use CommSy without hosting their own webserver. This new commercial perspective led to an exploration of new contexts of use.

One use context that seemed promising commercially were secondary schools. During its development, CommSy had been used to a small extent in secondary education by schoolteachers who had been introduced to CommSy during their studies at the University of Hamburg.

As existing relations were intensified and CommSy use spread to more and more schools, new requirements arose to adapt to the specific teaching situation and the needs of especially younger pupils.

Later in 2004, another research project was launched that focused on software support of highly networked virtual organizations (cf., Finck and Janneck 2005; Janneck and Finck 2006). In this project, CommSy was taken as a prototype that was continually adapted and customized due to the networks' needs. In this context, a number of requirements regarding specific project management-related features emerged. The networks wanted to use CommSy to support acquisition processes and also—in case of successful bids—to coordinate their joint business projects.

To sum up, during the now ten years of development, CommSy turned Open Source and new use contexts appeared, ranging from support for learning processes in higher education and in secondary schools, to the support of virtual organizations. Use in these different contexts lead to a variety of different and sometimes outright conflicting requirements. The participatory design process had to meet the challenge of increased physical, in some cases also temporal, but especially organizational distribution (cf., Gumm et al. 2006).

3 Bringing them all together: techniques for inter-contextual participatory design

Rooted in Floyd's STEPS software development model (cf., Floyd et al. 1989), CommSy was developed from the start using an evolutionary participatory design process (cf., Pape et al. 2002).

To get started a close cooperation with users, a mix of established PD methods such as paper prototyping (cf., Snyder 2003), user workshops (cf., Greenbaum and Kyng 1991; Braa 1995), interviews and scenarios (cf., Greenbaum and Kyng 1991; Kyng 1995; Rosson and Carrol 2002) were used, typically including a smaller number of especially active users in the process. Furthermore, various feedback channels were established, such as telephone and e-mail support, and a large user survey was carried out regularly to collect feedback on system use from a larger number of users (cf., Finck et al. 2004a).

User workshops were a particularly valuable PD instrument in all development phases to exchange experiences, discuss different uses, and elaborate requirements for the future development of CommSy. As a result, users involved in these processes started to develop a shared understanding of software support in their respective context. Typical characteristics of communities of practice (Fischer 2004; Wenger 1998) emerged—such as the negotiation of meaning among the members, mutual engagement in joint enterprises, and a shared repertoire of activities, symbols, and artifacts.

However, some of the specific requirements developed within these closed communities of practice turned out to be of little or no significance in other contexts. Also, users from different contexts had little in common regarding their ways of working, in their daily tasks, in the strategies for applying the software to challenges in their work practice, and in the language they used

to describe their work. They were rooted in increasingly diverging backgrounds; naturally, this limited the number of shared practices. Consequently, the ability of users from different communities of practice to discuss system use on the level of their actual work tasks decreased. There were still more abstract topics that were of interest to different practices; the interest for specific ways of working, however, was limited as users shared nothing beyond the software, and their practice was certain to be independent to that of other users.

This grew into a problem for the development team. On the one hand, exploring new contexts of use—and thus, new customers—was vital to the commercial interests of the spin-off company. On the other hand, the development team needed to bundle resources and tried to avoid parallel implementations that would increase the complexity of the software as well as administration and maintenance. Another goal was to avoid the fragmentation of the development team and process, called *forking* in open source projects (cf., Stalder and Hirsh 2002). Also, the different requirements had to be aligned with the original design philosophy.

To meet this new challenge of pooling the interests of different distributed communities, new ways of bringing users from different contexts together had to be established to balance their respective needs and emerging requirements. The goal was to enable communication of members of different communities of practice, both with each other and with the development team.

This challenge has been described by Arias et al. (2000) and Fischer (2001, 2004) as creating a *community of interest* (CoI). A community of interest spans across different communities of practice (CoP). Unlike CoPs, a CoI is not characterized by common practices of its members in their respective work context, but rather by a shared interest—in this case study, the object of shared interest is CommSy. For a community of interest, greater communication problems must be expected than within communities of practice. Fischer (2001, p. 4) writes, “fundamental challenges facing communities of interest are found in building a shared understanding of the task at hand, which often does not exist upfront, but is evolved incrementally and collaboratively [...] Members of communities of interest must learn to communicate with and learn from others [...] who have a different perspective and perhaps a different vocabulary for describing their ideas.” In other words, they need to establish a common ground and a shared understanding (see Table 2).

Building such a community of interest would allow—at least to some extent—a joint development without overly neglecting specific use contexts as users from different contexts can be included in the participatory design effort; this meets the challenge created through the greater dissemination of the CommSy software that was initially developed for a well-defined and focused community of practice.

In doing so, however, new needs for participation emerged. Existing PD methodology needed to be extended as it traditionally focuses on a single use context and the working practices in this context in order to fit the software to the task. Balancing the needs of users from different contexts is inherently more difficult as shared work practices are harder to identify and may not even exist. Thus, it became necessary to create participation on both the concrete use level and an abstract design level that can be shared by different groups of participants in the design process.

<i>User communities</i>	<i>Problems and artifacts</i>	<i>Example tasks and topics in the case study</i>	<i>Community members in the case study</i>
<i>CoPs</i>	Different tasks in the same domain and a uniform representation	Introduction of the community platform for e-learning in different schools and subjects Project management for virtual companies, use conventions in virtual teams	Users from an individual group sharing common practices, e.g., users from different secondary schools
<i>CoIs</i>	Multiple domains and different representational schemes	Software support Document management	All participating CommSy users

Table 2: Comparison of Communities of Practice and Interest (cf., Fischer 2001, p. 9).

To this end, two new techniques were introduced in the design process: inter-contextual user workshops and commented case studies. Inter-contextual user workshops bring users from different contexts and developers together to reflect on the usage and design of the software and its future development. Commented case studies make this type of face-to-face interaction persistent – providing a written documentation of use experiences and design decisions.

3.1 Inter-contextual user workshops

Starting in 2003, the CommSy development team has been organizing inter-contextual user workshops regularly about once a year. Their goal is bringing users from different contexts together in face-to-face interaction to discuss experiences and requirements, thus possibly finding a minimal consensus regarding future developments.

Participants

Usually, about ten to fifteen users and three to five developers or members of current research teams attend the one-day workshops. Over the past years, more than 50 users took part in the workshops.

All moderators of CommSy workspaces—typically teachers or leaders of project groups – receive individual invitations. Workshops are also announced publicly, e.g., on the CommSy website, to reach a wider audience.

Participants are typically ‘heavy users’, representing a certain institution or group, who will communicate the workshop results back to their colleagues. Their motivation is to partake in the future development of CommSy and to benefit from the experience of others’ ‘lessons learned’.

Setting

Generally, the workshops are framed as one-day events, starting in the late morning. When planning the workshops, different working conditions, time schedules and time constraints (e.g., term breaks, holidays, freelancers' high workload phases) need to be considered. Also, workshop locations (in the past, workshops have been hosted by universities and schools) are changed in a rotational system to allow people from different regions to participate, as the development team cannot compensate for travel expenses.

Workshops are prepared and moderated by two or three members of the interdisciplinary development and research team, including software developers and evaluation staff. Classical moderation and visualization (Metaplan) techniques are used to facilitate discussion. A detailed record is kept and sent to all participants afterwards together with the photominutes.

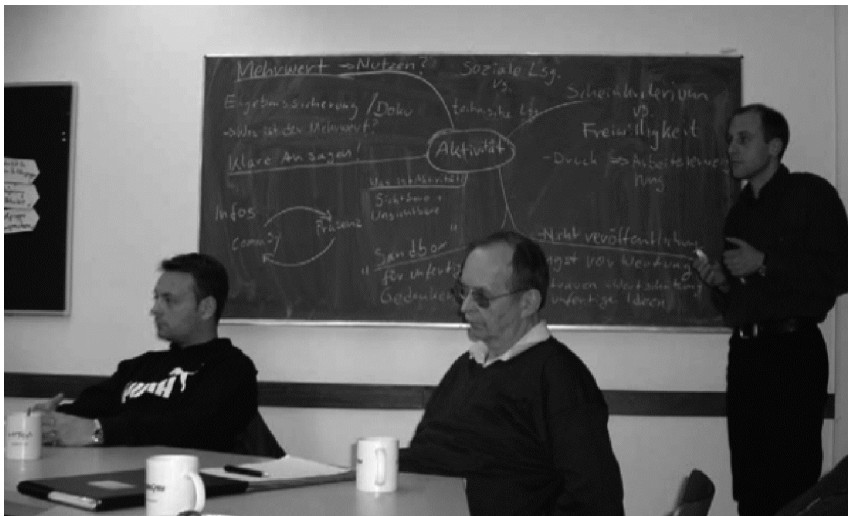


Figure 2: Workshop participants discussing mindmaps for future development.

Typically, the workshops start with an extensive round of introductions to help participants to get to know each other and their respective contexts of use. Then, developers report about upcoming developments. This is usually the starting point for an extensive discussion of use experiences, problems, and requirements. The moderators visualize the contributions for clustering later on. Sometimes the participants wish to work on different issues emerging from the discussion in smaller groups. Topics often address use problems or phenomena that are shared by many participants, e.g., how to increase active participation in online communication. A feedback round concludes the workshop.

Since these workshops are not primarily research activities and need to provide an informal and productive working atmosphere, they are not audio- or videotaped for detailed analysis. The workshops are evaluated using the records and documentation (posters, etc.). A lot of evaluation and documentation is actually carried out during the workshops, as participants, e.g., cluster

and prioritize requirements. The results are discussed within the development team afterwards and consolidated with other evaluation measures such as questionnaires (as described before).

Example

In the following section, a concrete example of an inter-contextual user workshop is described.

The workshop took place from 10.30 A.M. to 5 P.M. at a secondary school. It was organized by two schoolteachers who had introduced CommSy to their school and provided support for their colleagues regarding CommSy use. They had already attended earlier workshops and had offered to host the workshop when a location was needed.

Ten users and three members of the development team attended the workshop. Five users were also secondary school teachers, three were university lecturers and two were from an Employers' Liability Insurance Association which used CommSy as an intranet. Two of the participants hosted their own CommSy servers while the others made use of the application services provided by the CommSy development team. Figure 3 reproduces an excerpt from the workshop record.

Reflection

So far, the development team received thoroughly positive feedback by the workshop participants regarding the benefits and usefulness of inter-contextual exchange. Being confronted with perspectives originating in different contexts makes users reflect their own usage; this is possible from an angle not present within their own community of practice. Challenged by other participants to explain why certain features are important to them, and contrasting this with experiences from other backgrounds, they start to think through and sometimes question their use routines. An example: The demand for highly differentiated access rights, which was voiced by several users, turned out to be grounded in the transfer of practices participants knew from other software products. In the workshop, this boiled down to a discussion of trust, hierarchies, and authority and the value of equality that is inherent in the design of CommSy.

Nevertheless, establishing a dialogue between users from different 'cultures' (in the sense of different working contexts, disciplines, values), speaking different 'languages' (conveying very different experiences and beliefs) is not always easy and requires time. For example, at one occasion entrepreneurs, teachers and university lecturers all spoke of 'projects'—while addressing completely different settings. A lengthy argument arose on the correct usage of the term, which in the end had to be settled by the moderators.

Other conflicts arise when more 'powerful' user groups (who are able and willing to pay for specific developments, for example) threaten to dominate the development of the software. Inter-contextual user workshops help to set these conflicts at rest, as a compromise is worked out in direct interaction with other communities of practice. However, sometimes the compromise does not work out in practice and the problem surfaces again.

Furthermore, developers use the workshops to validate their own design decisions and ideas: If features are approved by users from different contexts and with different backgrounds, they are more likely to be helpful for the user community as a whole.

Excerpt From Workshop Record

1. Introduction

The workshop started with the participants introducing themselves, their involvement with CommSy, their motivation for participation and their expectancies for the workshop. Four clusters emerged for discussion: Exchange of experiences; problems and queries; feature requests; and participation in the development process.

Afterwards, the developers gave a short overview of the latest software developments and possibilities for application service providing.

2. Exchange of experiences

The discussion generally circled around the issue of lurking and motivating users (especially students) to participate actively within the workspace, respectively. The workshop participants shared experiences and “best practices”, e.g. methods to encourage online discussions and feedback or useful netiquette rules. Even though different approaches were regarded as appropriate in the different use contexts (e.g. when dealing with pupils rather than adults) the participants expressed that they had learned from each other. One participant, who had produced short videos as “guided tours”, offered to make them available for the other participants as well.

3. Problems and queries

Afterwards, the participants discussed difficulties they encountered when using CommSy. e.g. problems with access rights, copyright concerns when distributing digital material, data import/export, or breaking usage rules. A lengthy discussion arose on the topic of individualization, as participants missed opportunities to configure their workspaces according to their respective use contexts (e.g. by advocating different terms for workspace features). So far, the CommSy team refrained from offering individualization possibilities because of the increased complexity and also possibly inconsistencies that might arise from it, which might in turn impair usability.

Again, even though the participants faced different challenges in their respective contexts of use, they were able to share some useful advice. For example, one participant working for an Employers’ Liability Insurance Association offered to ask his legal department for standard form contracts which might be used by the other participants as well. The discussion on different terms and their meaning in diverse contexts raised the participants’ consciousness to watch out for different viewpoints and understandings within their contexts as well.

4. Feature requests

Besides the issue of individualization, especially school teachers asked for the possibility to trace individual use paths (e.g. to check whether certain pupils read their assignments). On the opposite, participants from other contexts pointed out that several users might share a computer and work cooperatively, with only one person actually logged in. Others argued that students might download a document but not necessarily read it. They suggested to call for feedback explicitly, for example by requesting written responses/comments to material posted in the workspace. After the discussion all participants agreed that individual read access information should not be provided by the system.

Two feature requests were brought forward congruently by all user groups: Workspace templates to save standard configurations and also content for reuse, and easy integration of other software tools (e.g. tools for online assessment).

5. User participation in the development process

The participants discussed further possibilities for participation in the development process. They suggested using blogs and a newsletter in addition to the existing websites and workspaces.

Figure 3: Excerpt from example workshop record.

Inter-contextual user workshops: a novel technique?

In contrast to more traditional PD workshops, inter-contextual user workshops focus on reflecting and enhancing use practices within the current system design rather than designing new or revised features. Therefore, traditional PD techniques such as paper prototyping or developing scenarios together with users have not been applied. While traditional PD workshops focus on support for the respective work context, it's not the primary purpose of inter-contextual workshops to analyze all of the participants' different backgrounds in detail. Instead, participants refer to the shared knowledge about an existing piece of software to highlight how its design relates to their tasks. This reflection will always be partial or anecdotal and cannot cover all aspects of neither the software design nor the respective use context. To some extent, this is a drawback. However, with users from backgrounds as diverse as in our case study, the immediate challenge was to enable basic communication. Compared to design discussions related to work tasks, as in traditional PD workshops, design discussions in inter-contextual workshops address a more abstract level (cf., section 4).

Inter-contextual user workshops are similar to focus groups – moderated group discussions (e.g., Krueger and Casey 2000)—as they also make use of discussions among participants. However, focus groups are mainly a method for data collection, used in qualitative or market research to measure attitudes and experiences. They are typically not oriented towards compromise, consensus and mutual learning, concepts central to the inter-contextual user workshops described here. Furthermore, group discussions are only one method incorporated in the workshops. If appropriate, workshop moderators also make use of a variety of other moderation techniques; these can be more focused on an outcome as, e.g., some specific design decision and its trade-offs need to be decided.

3.2 Commented case studies: inter-contextual communication made persistent

Participants of user workshops often wished to find a more permanent form for the workshop results and to establish an ongoing exchange between users from different contexts. To facilitate this, the CommSy development team set up an online discussion forum and invited users to participate. Yet this platform never came to life; participation was almost nonexistent due to the lesser immediate value compared to workshops.

Therefore, a new, more compact and more moderated form of documenting and distributing user experiences was established, called commented case studies. Commented case studies are similar to a small book or journal. They describe the use of CommSy in different contexts in original user voices, as users from different contexts are invited to reflect and report on their individual appropriation of CommSy and share their practices with the user community. Just like inter-contextual user workshops, they aim at communicating design decisions among a larger group of people involved in the development process and at enabling interaction between users of different communities of practice and also between users and developers.

Participants

The experiences reported here are based on two editions of commented case studies from 2004 and 2006, containing 23 case studies from 30 authors (Finck et al. 2004b, Finck and Janneck 2006).

To produce a collection of case studies, the editors—two or three members of the development team—contact moderators of CommSy workspaces and ask them to contribute. Naturally, a higher motivation is necessary for contributing a written text than for spending some hours in a workshop. Typically, contributors have made personal contact with the development team before (e.g., as workshop participants) and have experienced participation in the development process as fruitful.

The authors are typically ‘power users’, representing a certain institution or group. Their motivation is to influence the future development of CommSy by giving a voice to their respective user group and to benefit from the experiences of other users. The second edition of case collections was initiated and encouraged by users themselves, who wanted to read more about different use contexts.

Setting

Commented case studies require some time for completion as they loosely follow the publishing process of a scientific journal.

Invitations for contributions were made by e-mail and included a description of the current CommSy version that the reports would refer to. Also a common format and structure was proposed to make it easier to both comprehend and compare the user reports. As a side effect, the development team used the thematic structure to moderate the writing process and guide attention to topics they needed information on, much as in the workshop format. In the first edition, the structure included section titles such as initial configuration, introduction into use context, use conventions and so forth. This structure was not directly imposed on the authors, but the template document that was sent out included the section subtitles. As a result, the structure was accepted and used by almost all authors (some omitted sections they felt were not relevant for their report).

Contributed reports were subjected to a review cycle that was mainly focused on understandability and spelling, while care was taken not to influence the content of the article. These reports formed the main body of the resulting case study volume. To preserve the value of the contributions over time, a software description was prepended to the report collection.

The development team then grouped the different reports and created several classifications of the practices and problems described in the case studies. An excerpt of the classifications used in the case study are listed in Table 3: Apart from descriptive and quantitative data characterizing the individual use case, the reports were analyzed regarding similarities with other use cases and typical activities such as configuration of the workspace, documentation and support given to users, etc. This offered pointers to similarities and differences, providing the basis for developing a common language and the negotiation of requirements that also takes place in the workshops.

<i>Characterization of CoP</i>	Form of organization (e.g. research group, company) Reasons for using the software
<i>Quantitative data on use</i>	Number of Users Number of Instances of the system in use
<i>Configuration</i>	Focus of preparatory administrative work
<i>Documentation</i>	Forms of existing documentation used Additional user-created documentation
<i>Support</i>	Different support channels Roles and models for support
<i>Focus of use description</i>	Main topic the case reports deal with (e.g. information management, appropriation etc.)
<i>Features used</i>	Software features that were used in the respective context

Table 3: Examples for how the case studies were classified.

Finally, the last section of a volume consists of comments by the development team, discussing the different use experiences reported in the case studies and the feature requests resulting from them, and explaining software features and design decisions on the background of these use experiences. Also, an outlook on the upcoming version of CommSy was added to the volume to indicate how the development would react to the requirements voiced by the different user communities.

Example

In the following section, one edition of commented case studies is described in detail.

In the 2006 edition, contributions were made by users from *VIRKON*, a research project on virtual networks, *Mikropolis*, an information management research group, *Branta*, *JUMP* and *Consulting Netzwerk*, three networks of independent consultants, *HR-Verband*, a human resources association and *CI WPS*, an IT-consulting firm. All these organizations use CommSy to support their internal communication and documentation needs. Furthermore, *MQ21*, a European management development network, reported on their use of CommSy for organizing a conference, and a member of the *Branta* network described his use of CommSy as a communication platform for a single customer-related project. Finally, the *eCampus* report detailed how CommSy was introduced and supported in a University department.

Three different categories were created to group the case studies: preparation of use, appropriation of use, and facilitation of use. Furthermore, the case studies were categorized according to the functionalities of the software that were used and described (Figure 4). Based on these classifications, the case studies were indexed to enable readers to find case descriptions that match their use context and practices.

	VIRKON	Mikropolis	Branta	JUMP	HR-Verband	Consulting Netzwerk	MQ21	i-Portal Project	eCampus	C1 WPS
Description of use										
Mapping of organizational structures	•			•	•	•	•	•	•	•
Organization of documents	•						•	•	•	
Types of documents			•		•					•
Organization of use			•	•			•			•
Focus on features in use										
News										
Appointments										
Materials										
Discussions										
People										
Tasks										
Themes										
Groups										
Form of Organization										
Research Group		•	•							
Network				•	•	•	•	•		
Project									•	•
SME										•
CommSy workspaces used										
Shared Spaces	1		1	1	1	1	1	1		1
Project Spaces	9	1	5	4	3	3	1	1	6	1
Group Size										
< 10					•			•	•	
10 to 25			•	•						
25 to 50		•				•				•
> 50							•			•
Reasons for using CommSy										
Organizational requirement	•		•	•						
Mouth-to-mouth							•	•	•	
Previous experience			•						•	•
Direct contact	•	•	•	•						•

Figure 4: Different classifications for indexing the case studies

The case descriptions are about three to four pages in length. In this volume, they are typically structured along the following paragraphs: Description of use context (name of group/organization using CommSy, number of members, reason for using CommSy), types of usage (objectives and purpose of use, frequency and participation etc.), moderation of use (e.g., netiquettes and rules), and evaluation and conclusion (e.g., problems, lessons learned, feature requests etc.). Figure 5 shows an excerpt of an actual case description.

The case descriptions are followed by an extensive comment (about 10 pages) by the developers, reflecting their past design decisions, answering questions and commenting on feature requests raised in the case descriptions. For example, they describe how the CommSy design principles were slightly altered to meet the requirements of heterogenous use contexts better, and if and how (or why not, respectively) specific feature requests would be addressed in the future. Furthermore, they make suggestions how specific use problems might be targeted within the current software design.

opportunity to download files, they will arguably continue to use files posted in the project rooms although the course is officially over.

The instructor found sending e-mails (for individual students as well as for all members of the [redacted] through [redacted], an efficient way to communicate with students. Information pertaining to the course as well as to the English Department was conveyed in this manner on a regular basis. Although two courses were grouped together in the [redacted], two corresponding groups were not set up in "Groups" as the number of students in the [redacted] was relatively small compared to previous grammar [redacted] which have contained up to 75 students. Furthermore, the summaries, files, and e-mails were relevant for both groups as the weekly schedule was identical. Based on the instructor's observations, students also used the [redacted] to find the e-mail addresses of fellow students. Students' profiles, however, only included their names, user names, and e-mail addresses; personal information was not added.

Although students experienced few difficulties using [redacted] the instructor noted a few common user errors while troubleshooting. When typing the [redacted] address in the browser window for the first time, students often added "www." to the [redacted] URL and were frustrated when they could not find the start site. In an attempt to register for the correct [redacted], a few students tried to enter the instructor's older [redacted] at [http://\[redacted\]](http://[redacted]) or the instructor's other [redacted] at [http://\[redacted\]](http://[redacted]). Throughout the semester, students occasionally forgot their user names and passwords. Students often claimed that older entries had "disappeared" without realizing they were older than 7 days and, thus, no longer displayed on the "Home" page.

Most students (approximately 20) registered for [redacted] within the first two weeks of the semester. Students who failed to register did not receive e-mails that the instructor

Figure 5: Example of a case description

Reflection

To sum up, commented case studies consist of indexed and annotated typical descriptions of use written by real users, and a reflection and explanation of design decisions on the basis of these case studies. They follow an informally proposed structure, including a short description of the use context and purpose, the participants, the way the software was introduced and adopted in the respective setting, and an extensive report of use experiences and 'lessons learned'. An extensive introduction written by the editors gives an overview and classification of the cases presented and helps readers to select the ones that are most relevant or interesting for their purposes (cf., Finck et al. 2004b, Finck and Janneck 2006).

Commented case studies satisfy several possible uses:

For developers, they bundle authentic, unedited reports from different use contexts in a comparable way to help them develop a more thorough understanding of their requirements beyond the observable feature requests. Furthermore, the commented case studies serve as a collection of use cases for evaluation or documentation purposes.

Users are given access to the experiences of other users. If desired, they can get in contact with them to exchange their experiences. Furthermore, by commenting the case descriptions, developers document design decisions in a transparent way, enabling users to gain insights about how and why CommSy is developed in the way it is, or—possibly—why design proposals users made were changed or could not be taken into account. This also makes it possible to

anticipate future design decisions, especially when they do not primarily address the respective use context.

One major drawback of commented case studies is that the production of a volume is quite time-consuming. Thus, by the time it is published the case descriptions might already be outdated, especially regarding the software version that they correspond to. However, the underlying use scenarios described in the case studies seem to be subject to somewhat lesser changes.

Commented case studies: a novel technique?

Commented case studies can be contrasted to use cases or scenario techniques employed in software engineering (cf., Hertzum 2003). While use cases are highly formalized and detailed descriptions relating to the way concrete implementations are planned or carried out, commented case studies describe existing experiences of use in a more anecdotal, less formalized way. Scenarios, on the other hand, which are similar to commented case studies in their narrative descriptions of tasks and ways of use, are meant to capture typical, representative descriptions of use, while commented case studies are highly individual.

4 Reflections on inter-contextual participatory design

To summarize experiences, we found that traditional PD methods are well adapted to work within a singular context of use, or community of practice. By developing methods for inter-contextual PD to be applied in project settings where stakeholders are physically and, even more important, organizationally distributed, we were able to establish a diverse community of interest. The methods we presented here reflect on system usage and design both on a concrete level of use practices and tasks within specific communities and an abstract level of design philosophy and underlying viewpoints and values within the different contexts. By addressing values on an abstract level, users from different contexts succeed in clarifying their requirements without having to share details of their daily work routines and practices. Furthermore, both methods aim at establishing a deepened understanding between users and developers, especially when some of the developers are almost exclusively involved in just one context. Ideally, users and developers succeed in speaking a common language.

Applying PD to different communities of practice with their special needs presents a number of challenges that have not been central to PD research so far. Traditional PD workshops and future workshops (Greenbaum and Kyng 1991) are useful to criticize and develop existing work practice in a single community of practice with similar daily work practices and tasks. Even when different parties participated, they were involved in one context as stakeholders with different interests (e.g., O'Day et al. 1996; Henderson and Kyng 1991). The same is true for other design techniques, such as probes (see, e.g., Kanstrup and Christiansen 2006 for an example of a postcard probe or Lindquist et al. 2007 for examples of cultural communication and technology probes), design games (e.g., Brandt et al. 2008) or various forms of prototyping: They focus on the design task to support users and designers in finding an optimal solution for their concrete everyday tasks—“designing for skill and work practices in context” (Binder et al. 2008, p. 2).

Some of these methods have also been applied in distributed contexts before (cf., Lindquist et al. 2007; Björgvinsson 2008). However, while dealing with e.g., physical distribution as in the case of Lindquist et al., who explored design activities with distributed family members, they were still designing for shared practices and tasks, such as communication between family members. Björgvinsson (2008), who emphasizes the vital interaction between co-workers rather than between workers and designers, still explicitly states that this was carried out within a single community of practice.

For multiple distributed contexts, it is necessary to first establish a common ground for experiences. When DPD aims to build a community with shared interests, it is in our experience very useful to consequently develop sensitivity for different perspectives among the users. We introduced techniques to foster a shared understanding among users from different contexts and gain insights into each others' perspectives, values, and norms: Both the inter-contextual user workshops and the commented case studies were successfully employed to foster a community of interest (although the software developers might have lost some prospective customers who did not share the common values). While other design techniques might also be adapted to work towards this goal, this has yet to be investigated in future research.

In contrast to communities of practice, a community of interest requires no shared repertoire of activities and little mutual engagement between its members. It still needs a shared repertoire of symbols—a common language—and of artifacts—software being a central one for participatory design. The community of interest thus has no need for codified knowledge, instead focusing on shared understanding and trying to “make all voices heard” (cf., Fischer 2004). This matches well with the personal motivation of the users – naturally, not all members of all communities of practice were interested in learning about software use in unrelated contexts, but the common language used for inter-contextual community building revealed shared interests, and seemingly unrelated activities successfully served as stimuli for self-reflection. Users explicitly stated that they were surprised how differently CommSy was used in other contexts, but nevertheless they learned from the practices employed in these “exotic” settings and adopted and generated new ideas for their own software use.

The experiences from this case study can be generalized based on the distinction between the different levels of involvement that coexisted during the development process (see Figure 6): communication with individual user representatives, with members from a single community of practice, and with representatives from several different communities of practice that should form a community of interest.

- *Single users / user representatives:* A permanent form of participation existed through the one-to-one link of key users and individual members of the development team. The nature of this participation can be described as feature-based – design requests often concerned single features and were based on immediate tasks.
- *Community of practice:* Development workshops with members of a single community of practice were scheduled to involve stakeholders without immediate contact with the development team in the process, to broaden the view on the application context and to provide a forum for reflection and redesign of organizational practices. While key users and development team members already cooperating with each other typically initiate the discussion, active participation of other end users —and developers—is encouraged

in this format. The participatory design process touches both concrete features/tasks and use practices/organizational structures.

- *Community of interest*: Inter-contextual user workshops and commented case studies draw upon experience from different communities of practice. User-user and user-developer interaction is expanded to include different contexts. This distributed form of participatory design covers anecdotal aspects of the software where interests meet. To find a common language, the more abstract discussion is centered on values and their application within different contexts to form a use vision appropriate to individual communities of practice. Shared values allow formulating a common product vision and building a community of interest. Both the interested users and all members of the development team can improve their understanding of the different contexts the software tries to serve and thus better understand the direction that design will be taking.

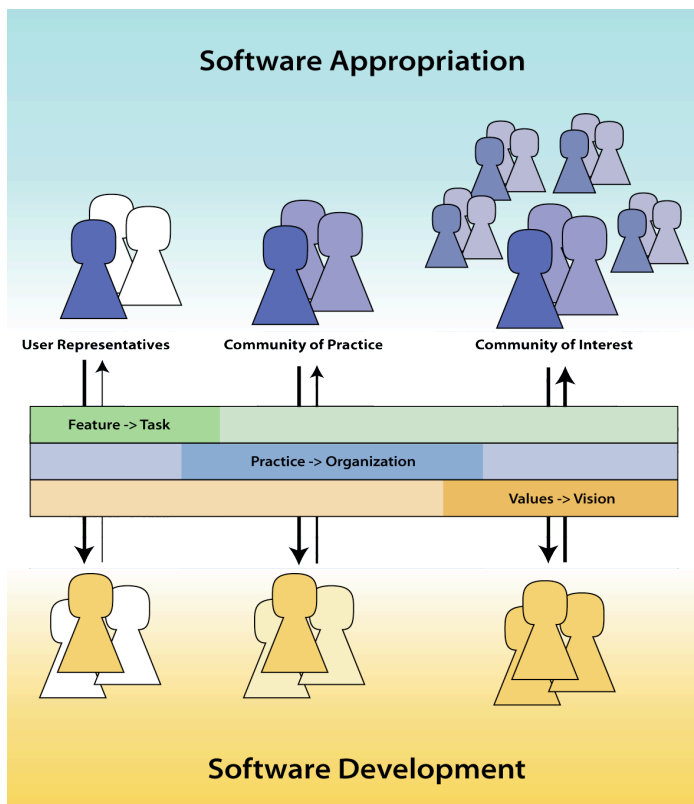


Figure 6: Extension of participation scope in user and developer domain and increasingly abstract level of discussion.

The reported shift from targeting a single community of practice to trying to support heterogeneous, distributed contexts changed the aims for participatory design: development could no longer pursue optimal software quality for a well-defined user audience or cater for a coherent, finite number of tasks and redesign work practice within a single organization. Rather, compromises had to be formed. On a concrete level, limited resources and conflicting requirements had to be moderated: “what is (already) possible to do with the software?”—with appropriation (cf., Pipek 2005) taking a more prominent role. On an abstract level, the nature of these compromises needs to be reflected. A discourse about the values inherent in the software design and their application and applicability to different contexts allows an indirect and abstract dialogue about concrete features. While discussions with users are often centered on feature requests and new ways of tailoring the software to existing processes, awareness of how the same ‘tool’ is used in completely unconnected contexts made it possible for the discussion to shift to a more abstract level. Both users and developers are encouraged to take a step back from their respective standpoints and their daily routines to gain an understanding of other forms of usage.

Thus, the development of values and a design philosophy for the software acquired a central role. Values were used to communicate the reason for taking one design direction (and consciously ignoring other options) to users (e.g., for maintaining non-hierarchical access rights in spite of CommSy use in more hierarchical contexts). The design philosophy acquired a communication function, and changes in the direction of design became visible as the values underlying the design discussions shifted. Features proposed or demanded by users could be assessed not only by asking what consequences they would have in respect to certain work practices, but also how they would affect central design principles.

Creating a shared language for discussing design and understanding what the software is about enabled users from dissimilar contexts to begin an exchange about their perspectives on *how* and *why* to use this particular groupware system. In our view, this is another facet of ‘user empowerment’: In addition to the ability to influence organizational structures, users acquire skills for technology selection, adoption and use. However, sometimes the level of abstraction lead to misunderstandings, e.g., when the users propagated values of their work that were actually not so visible in their daily routines (cf., Greenbaum et al. 1994).

Inter-contextual user workshops were a useful tool for fostering a community of interest as they focus on commonalities between different use contexts in their dual discussion of specific software aspects and abstract software values. As differences between the communities of practice mainly reside on the organizational level that is skipped in communal discussion and only surfaces in personal reflection, workshop communication successfully manages to focus on similarities. Lifting the discussion to the abstract level of values allows an indirect exchange regarding work practices and organizational structure that would not be possible otherwise.

The commented case studies add to this a persistent form and greater depth. Authors are given more room to detail their experiences and to reflect on their personal or organizational work practices. While this is associated with a slower feedback cycle, we see the written format as a very useful complement to user support, traditional workshops and inter-contextual use workshops, and as an important link to participatory use documentation. As with all written documentation, however, the audience must be considered carefully. While the format used here may be well suited for users with an academic background, target groups who are less accustomed to abstract, long-winded discussions may prefer other media.

<i>Users</i>	<i>Development Team</i>
Increased acceptance Empowerment Use vision	Legitimation for design Domain understanding Software vision

Table 4: Benefits of inter-contextual community building.

Based on the different levels of abstraction that we find in both the workshops and the case studies, we distinguish three levels where the software development process benefited from the methods described here (Table 4): on the first level, users' acceptance of development decisions is increased as they learn about the rationale behind design. Reciprocally, developers obtain legitimation as their underlying values are being confirmed by users. On the second level, empowerment of users comes along with a deepened understanding of the respective domain on the developers' side. Building upon this, on the third level, users are enabled to integrate the software into their work practice, while developers can form a consistent vision spanning the different use contexts.

5 Summary and future work

In this paper, we analyzed changes in focus and aims of participatory design based on the ComSy case study: In increasingly distributed contexts, the focus shifted from custom software development to empowering users in assessing their own practice and technology use. While discussions between stakeholders and developers remained detailed concerning the application of certain features, they became more abstract in communicating values and design philosophy instead of analyzing daily work practice. The resulting participation was essential for this project to foster acceptance for design decisions, enriching user experience, and building an inter-contextual community across several distributed use contexts.

With distributed participatory design across several contexts, the repertoire of PD techniques was adapted and changed. New methods for DPD were developed—inter-contextual user workshops and, as a persistent alternative, commented case studies – and integrated into the software development process. While the decision to extend PD methodology was at the time due to practical necessity rather than theoretical reflection, the changes in the user-developer-interaction can be interpreted as emergence of a new level of abstraction. Introducing values into discussions with users provided a common language for different communities of practice as a basis for collective design.

Furthermore, this type of inter-contextual interaction turned out to actually *enrich* user experience by encouraging users to experiment with new, creative, and unthought-of ways of use and also to rethink the very foundations of their work practices.

There is, however, further need to expand the repertoire of distributed participatory design methods. Especially the issue of the up-to-dateness of the information documented in the

commented case studies might need to be addressed. As sustainability was a main goal for the research project during which the work described here was performed, its end presents a promising test case for measuring success. Finally, we seek to explore even further use contexts for CommSy development and will continue to put inter-contextual PD to the test.

In this paper, we described an alternative PD approach for distributed software development, where new use contexts were sought that share not the concrete work practices, but common use values—a community of interest (cf., Fischer 2004) formed by distributed participatory design. As a discipline, PD will make a decision whether the challenge of building software that spans individual communities of use is a worthwhile goal. Current PD methodology focuses on optimizing support for existing and future work practices for different groups within a single context. Compromises will have to be made if that focus is extended: commercial products often successfully apply an approach that modularizes functionality and provides different perspectives to different target groups.

6 Acknowledgement

First and foremost, we would like to thank all our users, especially the workshop participants and the case study authors. We have all learned much from you, and continue to do so. This work would not have been possible without our colleagues in the WissPro project and at the University of Hamburg. We would like to thank Christiane Floyd and Horst Oberquelle for laying the roots for this work, and for continuous support during the last decade. We thank Susanne Bødker for her comments on an earlier draft, and for helping us to find our own words for the issues described in this paper. Finally, we would like to thank the anonymous reviewers for their patient and helpful insistence on numerous improvements to this paper.

7 References

- Amaury, D., Charlier, B., France, H. and Esnault, L., “Participatory Design for Developing Instruments for and with Communities of Practice: a Case Study,” 2nd Workshop on Distributed Participatory Design, CHI 2008, Italy, Florence (2008).
- D’Andrea, V., Baskin, A., and Reinke, R., “A Distributed Software Development Team Meets a Distributed Community of Practice: Participatory Meetings,” 2nd Workshop on Distributed Participatory Design, CHI 2008, Italy, Florence (2008).
- Arias, E., Eden, H., Fischer, G., Gorman, A. and Scharff, E., “Transcending the Individual Human Mind. Creating Shared Understanding through Collaborative Design,” *ACM Transaction on Computer-Human Interaction*, (7), 2000, pp. 84-113.
- Barcellini, F., Détienne, F., and Burkhardt, J.-M., “Requirements for design participation in Open Source Software communities,” 2nd Workshop on Distributed Participatory Design, CHI 2008, Italy, Florence (2008).
- Binder, T., Brandt, E., and Gregory, J., “Design Participation(-s),” *CoDesign*, (4: 1), 2008, 1-3.

- Braa, K., "Priority Workshops: Springboard for User Participation in Redesign Activities," in *Proceedings of COOCS 1995*, ACM Press, 1995, pp. 246-255.
- Braa, K. and Vidgen, R., "Action Case: Exploring The Middle Kingdom in IS Research Methods," in *Proceedings of Third Decennial Conference Computers in Context: Joining Forces in Design*, 1995, Aarhus, Denmark, pp. 50-60.
- Brandt, E., Messeter, J. and Binder, T., "Formatting design dialogues – games and participation," *CoDesign*, (4: 1), 2008, pp. 51-64.
- Bjerkness G. and Bratteteig, T., "A strategy for work life democracy?" in *Proceedings of PDC 1994*, CPSR, 1994, pp. 3-12.
- Björgvinsson, E. B., "Open-ended participatory design as prototypical practice," *CoDesign*, (4: 2), 2008, pp. 85-99.
- Bødker, S., Grønbæk, K. and Kyng, M., "Cooperative Design: Techniques and Experiences from the Scandinavian Scene," In *Human-computer interaction: toward the year 2000*, R. M. Baecker, J. Grudin, W. A. Buxton and S. Greenberg (eds.), Morgan Kaufmann, 1995, pp. 215 - 224.
- Checkland, P., "From Framework through Experience to Learning: the Essential Nature of Action Research," in *Information Systems Research: Contemporary Approaches and Emergent Traditions*, H.-E. Nissen, H. K. Klein and R. Hirschheim (eds), North Holland, Amsterdam, 1991, pp. 397-403.
- Cole, J., Foster, H., Using Moodle, O'Reilly, 2007.
- Czyzewski, P., Johnson, J. and Roberts, E., "Introduction – Purpose of PDC '90," in *Proceedings of PDC 1990*, CPSR, 1990, pp. i-ii.
- O'Day, V.L., Bobrow, D.G., Hughes, B., Bobrow, K.B., Saraswat, V.A., Talazus, J., Walters, J. and Welbes, C., "Community designers," in *Proceedings of PDC'96*, CPSR, 1996, pp. 3-13.
- DePaula, R., "Lost in translation: a critical analysis of actors, artifacts, agendas, and arenas in participatory design," in *Proceedings of PDC 2004*, ACM, 2004, pp. 162-172.
- Elovaara, P., Igira, F. T., and Mörtberg, C., "Whose participation? whose knowledge? exploring PD in Tanzania-Zanzibar and Sweden," in *Proceedings of PDC 2006*, ACM, 2006, pp. 105-114.
- Erickson, T., Halverson, C., Kellogg, W.A., Laff, M. and Wolf, T., "Social translucence: designing social infrastructures that make collective activity visible," *Communications of the ACM*, (45: 4), 2002, pp. 40-44.
- Floyd, C., Reisin, F.-M. and Schmidt, G., "STEPS to software development with users," in *Proceedings of ESEC 1989*, Springer, 1989, pp. 48 - 64.
- Finck, M., Gumm, D. and Pape, B., "Using Groupware for Mediated Feedback," in *Proceedings of PDC 2004*, ACM, 2004a, pp. 45-48.
- Finck, M. and Janneck, M., "Hospitality in Hosting Web-Based Communities: Two Case Studies," in *Proceedings of WBC 2005*, IADIS, 2005, pp. 327 - 330.
- Finck, M., Obendorf, H. and Pape, B. (eds.), *Fallbeispiele der CommSy-Nutzung – Eine Sammlung von Nutzungsberichten* [Case Studies of Using CommSy], University of Hamburg, Department of Informatics, 2004b.

- Finck, M. and Janneck, M. (eds.), *Fallbeispiele der CommSy-Nutzung im Kontext virtueller Organisationen und Netzwerke - Eine Sammlung von Nutzungsberichten*. [Case Studies of Using CommSy in Virtual Organizations], University of Hamburg, Department of Informatics, 2006.
- Fischer, G., "Communities of interest: Learning through the interaction of multiple knowledge systems," in *Proceedings of the 24th Annual Information Systems Research Seminar in Scandinavia*, Ulvik, Norway, 2001, pp. 1-14.
- Fischer, G., "Social creativity: turning barriers into opportunities for collaborative design," in *Proceedings of PDC 2004*, ACM, 2004, pp. 152-161.
- Franssila, H. and Pehkonen, M., "Feasibility of iterative design as a challenge in distributed participatory design lifecycle," Nordichi workshop on distributed participatory design, Oslo, Norway, 2006.
- Greenbaum, J. and Kyng, M. (eds.), *Design at Work: Cooperative Design of Computers Systems*, Lawrence Erlbaum Ass, New Jersey, 1991.
- Greenbaum, J., Snelling, L., Jolly, C., and On', J., "The limits of PD? Contingent jobs and work reorganization," in *Proceedings of PDC 1994*, CPSR, 1994, pp. 173-174.
- Gumm, D.C., Janneck, M. and Finck, M., "*Distributed participatory design - a case study*," Nordichi workshop on distributed participatory design, Oslo, Norway, 2006.
- Gumm, D. C., "Distribution Dimensions in Software Development Projects: A Taxonomy," in *IEEE Software*, September/October 2006, pp. 45-51.
- Henderson A., Kyng M. There's no place like home: Continuing design in use. In: J. Greenbaum and M. Kyng (eds) *Design at work: Cooperative Design of Computer Systems*. Hillsdale, NJ, Lawrence Erlbaum Ass. (1991), 219 - 240.
- Herrmann, T., Kunau, G., Loser, K. and Menold, N., "Socio-technical walkthrough: designing technology along work processes," in *Proceedings of PDC 2004*, ACM, 2004, pp. 132-141.
- Hertzum, M., "Making use of scenarios: a field study of conceptual design," *Int. J. Hum.-Comput. Stud.*, (58: 2), 2003, pp. 215-239.
- Irestig, M., Eriksson, H., and Timpka, T., "The impact of participation in information system design: a comparison of contextual placements," in *Proceedings of PDC 2004*, ACM, 2004, pp. 102-111.
- Janneck, M. and Finck, M., "Appropriation and Mediation of Technology Use in Stable Self-organized Online Communities," in *Proceedings of WBC 2006*, IADIS Press, 2006, pp. 149-156.
- Janneck, M., Finck, M. and Obendorf, H., "Participatory Design: An Issue for Web-Based Community Development?!" in *Proceedings of WBC 2006*, IADIS Press, 2006, pp. 274-277.
- Kanstrup, A. M. and Christiansen, E., "Selecting and evoking innovators: combining democracy and creativity," in *Proceedings of NordiCHI 2006*, ACM, 2006, pp. 321-330.
- Karasti, H. and Syrjänen, A., "Artful infrastructuring in two cases of community PD," in *Proceedings of PDC 2004*, ACM, 2004, pp. 20-30.
- Kensing, F., "Participatory design in commercial settings: A conceptual framework," in *Proceedings of PDC 2000*, CPSR, 2000, pp. 116-126.

- Korpela, M., Soriyan, H.A., Olufokunbi, K.C., Onayade, A.A., Davies-Adetugbo, A. and Adesanmi, D., "Community Participation in Health Informatics in Africa: An Experiment in Tripartite Partnership in Ile-Ife, Nigeria," *Computer Supported Cooperative Work* (7), 1998, pp. 339-358.
- Krueger, R. A. and Casey, M. A., *Focus groups: A practical guide for applied research*. Sage Publications, Newbury Park, CA, 2000.
- Kyng, M., "Creating Contexts for Design," in *Scenario-Based Design*, J. Carroll (ed.), John Wiley, 1995, pp. 1-24.
- Lindquist, S., Westerlund, B., Sundblad, Y., Tobiasson, H., Beaudouin-Lafon, M., and Mackay, W., "Co-designing Communication Technology with and for Families – Methods, Experience, Results and Impact," in *The Disappearing Computer, Interaction Design, System Infrastructures and Applications for Smart Environments*, N. Streitz, A. Kameas, I. Mavrommati (eds.), Springer, Berlin, 2007, pp. 99-119.
- Mathiassen, L., "Reflective Systems Development," *Scandinavian Journal of Information Systems* (10: 1 & 2), 1998, pp. 67-118.
- Merkel, C. B., Xiao, L., Farooq, U., Ganoe, C. H., Lee, R., Carroll, J. M., and Rosson, M. B., "Participatory design in community computing contexts: tales from the field," in *Proceedings of PDC 2004*, ACM Press, 2004, pp. 1-10.
- Pape, B., Bleek, W.-G., Jackewitz, I. and Janneck, M., "Software Requirements for Project-Based Learning – CommSy as an Exemplary Approach," in *Proceedings of HICSS 2002*, IEEE, 2002.
- Pipek, V., *From tailoring to appropriation support: Negotiating groupware usage*, PhD Thesis, Faculty of Science, Department of Information Processing Science (ACTA UNIVERSITATIS OULUENSIS A 430), University of Oulu, Oulu, Finland, 2005.
- Rosson, M. B. and Carroll, J., *Usability Engineering: Scenario-based Development of Human-Computer Interaction*, Morgan-Kaufman, 2002.
- Schuler, D. and Namioka, A., *Participatory Design: Perspectives of Systems Design*, Lawrence Erlbaum Assoc., 1993.
- Schümmer, T. and Lukosch, S., *Patterns for Computer-Mediated Interaction*, Wiley & Sons, 2007.
- Stalder, F. and Hirsh, J., "Open Source Intelligence," *First Monday*, (7: 6), 2002. http://firstmonday.org/issues/issue7_6/stalder/index.html
- Snyder, C., *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*, The Morgan Kaufmann Series in Interactive Technologies, 2003.
- Stefanov, S., Rogers, J. and Lothar, M., *Building Online Communities with Phpbb 2*, Packt Pub, 2005.
- Vidgen, R. and Braa, K., „Balancing Interpretation and Intervention in Information System Research: The Action Case Approach," in *Information Systems and Qualitative Research*, A.S. Lee, J. Liebenau and J.I. DeGross (eds.), London, 1997, pp. 524 - 541.
- Wenger, E., *Communities of Practice. Learning, Meaning, and Identity*, Cambridge University Press, 1998.

