Integrating User Advocacy into Participatory Design: The Designers' Perspective

Peter Mambrey, Gloria Mark, Uta Pankoke-Babatz
GMD-FIT
German National Research Center for Information Technology
Institute for Applied Information Technology
CSCW Research Group
Schloß Birlinghoven
53754 Sankt Augustin, Germany
+49 2241 14 {2710, 2416, 2707}
E-mail: {Mambrey,Gloria.Mark, Pankoke-Babatz}@gmd.de

ABSTRACT
We report on participatory design activities within the POLITeam project, a large project which introduces groupware into the German government. Working with a representative small group of users in different worksites, an existing system was adapted to user and organizational needs, with the plan to improve and expand the system to a large scale. We integrated new approaches of user advocacy and osmosis with an evolutionary cycling process. User advocates and osmosis were techniques used to explore the users' needs during actual system use and were incorporated into the system development. In this paper, we present experiences with this approach and reflect on its impact on the design process from the designers' point of view.

Keywords
System design, participatory design, user advocacy, evolutionary system design

1. INTRODUCTION: THE RESEARCH AND DEVELOPMENT CONTEXT

POLITeam - telecooperation within and among ministries

In 1991, the German parliament voted to move the capital from Bonn to Berlin. This vote has substantial consequences for German society, since one-third of the government agencies will move to Berlin, but this decision especially has significance for the potential of telecooperation. Within the larger context of the POLIKOM research project, launched by the Ministry of Education, the POLITeam project, begun in 1994, concentrates on providing telecommunication support for group coordination, task management, work-flow and archiving, between the distributed governments of Bonn and Berlin (Hoschka et al., 1993). Project partners are the Federal Ministry of Family, Senior Citizens, Woman, and Youth, and several ministries of the state of Western Pommerania.

The introduction, adoption, and development of groupware in the ministries should meet several system development goals: daily use in working life instead of laboratory research, adoption of a groupware system to the needs of users and the organization, introduction of new concepts of groupwork into the ministries, and step-by-step integration to a larger user group (Klöckner et al., 1995). It was planned in the initial proposal to begin with a system and to further develop it according to the user needs. An existing groupware system was carefully introduced into the work practice of a small unit of a ministry consisting of 12 employees and typists of the central writing pool (Sohlenkamp et al., 1996). The field trial is currently expanding and will lead to new versions. In this paper, we would like to introduce our approach of involving user participation in the development process of our system and will focus on the designers' reflection of this design approach.

Designing design: multiple actors define research and development

The project formally began with a detailed written proposal planning the different research activities. To those who do external monitoring, i.e. supervisors from academia and a controlling board, the project's highly structured nature evokes transparency and progress. But the ideal of a structured, rational system development following prescribed plans is perceived differently by those working as system designers within the project. For them, the guiding visions and goals occur as a mixture of complex individual and organizational goals, and actual technical, social, or organizational constraints, that are embedded in dynamic political and technical discussions.

Our story is a story of reactions to a dynamic groupware adoption in a ministry based on some coarse ideas on how to integrate the users in the adoption and redesign of a

In *PDC'96 Proceedings of the Participatory Design Conference.* J. Blomberg, F. Kensing, and E.A. Dykstra-Erickson (Eds.). Cambridge, MA USA, 13-15 November 1996. Computer Professionals for Social Responsibility, P.O. Box 717, Palo Alto CA 94302-0717 USA, cpsr@cpsr.org.

system. In papers, a logical structure leads from research questions to conceptualizations, approaches, tools, methods, results, and finally conclusions. In real working life, such a logical thread is not always appropriate. Our goal was to realize user participation, by cooperatively developing a communication structure between designers and users. We began with a set of options, based on our combined experiences and capabilities, and tried to determine which procedure would work best in establishing this user-designer communication. The decision making process was made difficult because the external conditions were so dynamic; our goal was often like a moving target. Our objectives were often not precisely planned in advance but rather evolved during our work.

Why is this the case? It is because a dynamic introduction and adoption process of groupware within a working organization is comparable with a decision-making process where different actors, roles, goals, solutions, etc. negotiate in different arenas within different sets of constraints. We cannot name all overt or hidden goals but a short overview makes actors, goals, and constraints more explicit. This complex mixture set the stage for our approach. It is like a worm in an apple: the way is self-determined, but the container is given. In this container we identify different factors:

- experiences, attitudes, and career expectations
- different research and development orientations
- working contexts
- goals, tasks, methods, and tools
- political and economical contexts

The factors are active on different levels: as paradigms, as goals, as tasks, and are dynamic. The result of working within this mixed container was the development of a user advocacy approach which was integrated with a more conventional participatory design approach and evolutionary cycling. After two years of experience with this approach, and facing yet two more years of project work, we initiated a self-reflection process among the designers about their perspectives on the applied methods and tools used in design. We regard this not as a final evaluation but as a snapshot on designers' perspectives of participatory design in a current project. The purpose of this paper is to report these perspectives as a way of understanding designers' reactions to an approach in which they are faced with a totally new method of design. In the next section we describe the design process.

2. THE PROJECT TEAM AND DESIGN APPROACH

The development team had several integrated divisions, dealing with different but inherent problems of system implementation ranging from more technical to more social science-based perspectives. The players were:

- nine designers, with training in computer science and with a scientific orientation, some working toward their PhDs. For most designers, it was their first experience with designing and implementing an actual system outside the laboratory. The designers are conceptually-oriented and had large ambitions for the project.
- two user advocates, with training in computer science, and practical experience in system design and in user training and services
- two social scientists, with experience in information technology

The design approach

We designed a cooperative development process with mutual learning between designers and users to exchange experiences of design and work practice, and to enhance people's cognitive access to the changes. The evaluation criteria of the system are: the usage in existing organizations, how it assists users in their tasks, and how it helps users to meet their requirements for organizational goals and working life. Integrating autonomy and flexibility into the groupware system is an explicit goal of the project. To achieve this, we focus in particular on the users who use the system in a real world setting: their workplaces (Bowers, 1994). It was our intent that the integrated design approach that we developed should give us the chance to react flexibly if conditions and requirements would change (Mambrey, Oppermann, Tepper, 1987). In the following sections we describe the three components integrated into the design approach.

Evolutionary cycling

We applied an evolutionary approach because it enables stepwise modification and development (see figure 1). The approach, planned as an open process, takes into account the dynamics of the organization such as the organizational goals, organization of group work, career paths, fluctuation of personnel, cognitive access to the system, qualification processes, the state of the art of technology, etc. (Floyd and Keil, 1982).

The definition of the requirements for the system components is done in such a way that an existing system with a sufficient basic functionality is installed at the users' workplace, and the requirements are determined by means of the practical experiences and theoretical knowledge gained by using the system. With this approach, users have a tangible artifact with which they can evaluate their needs (Ballay, 1994).

In order to incorporate the user feedback into the system development process, as the evolutionary cycling required, we had to find appropriate ways. We set several requirements for ourselves. First, we wanted to benefit the design process as much as possible by having each project member retain competence in their own profession, i.e., the users should remain specialists in their work, and the, designers specialists in the system (Kyng, 1994). Second the users should benefit from the system use without overburdening them too much with the design process. Because we are designing a groupware system, we must be aware that unlike single-user systems, the appropriateness of groupware can only be evaluated in real practice during group work (e.g. Suchman, 1987). Thus, we did not want to distract users from their work by bringing them into the design team. Instead, they should be free to concentrate on their work and rather it should be the design team's task to elicit the user feedback and requirements. Therefore, the design team worked on establishing communication channels to them. This process was not unlike a Darwinian selection; those procedures which survived and proved to be the most successful in establishing user-designer communication channels were osmosis and user advocacy.

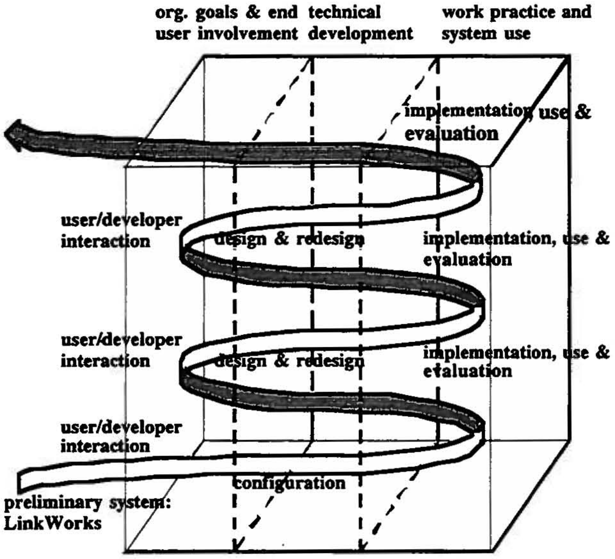
org. goals & end user involvement
technical development
work practice and system use
implementation, use & evaluation
user/developer interaction
design & redesign
implementation, use & evaluation
user/developer interaction
design & redesign
implementation, use & evaluation
user/developer interaction
configuration
preliminary system: LinkWorks

Fig. 1. An illustration of the evolutionary cyclic approach.

Osmosis

We use the term "osmosis" to refer to that multi-level information that a designer receives by visiting the workplace and having contact with users. This gives a rich picture of the users' working life which cannot be reproduced in any other form, such as paper, word-of-mouth, etc. This contact encompassed interviews, user workshops, active user services, and simply being present at the workplace. Designers reported that they felt like a sponge soaking up information. Osmosis took place at the beginning of the project and preceded user advocacy.

User advocacy

The idea of user advocates originated from several sources. It came from the past experience of team members who worked before on large-scale development projects. In large scale projects, for practical reasons there is a necessity for division of labor. Also, team members felt that the roles of system designer and user evaluator could not effectively be performed by the same individuals, due to conflicting goals, such as having to defend one's own design decision. In addition, because the potential interaction with users could be extremely high in such a large-scale project, it was decided to "channel" the access to users. Lastly, the idea of a consultant with expertise working together with a client group has theoretical origins in models of process consulting (Schein, 1988) which focus on developing close relationships to work out joint solutions.

In our project, the goal of using user advocates was to augment interaction between users and designers. First, the user advocates did active user services, watching and advising the users while they were using the system in their normal working situation. Second, as user representatives, they reported the users' proposals and requirements during design workshops and made design decisions jointly with the designers. Lastly, the evaluations of prototypes were done by the advocates, and explored by using examples from the user practices.

This role required the following qualifications:

- Having experience with user services in the past, they could easily understand the user's point of view: in gathering design requirements, and in explaining technical design decisions.
- Trained as computer scientists, they could understand technical issues of the system enabling them to propose technical changes and additional functionalities and to hold sophisticated technical design discussions with the designers. In these discussions, the user advocate could draw on their expertise, and challenge the technical reasons behind the decision.

The benefits envisioned in using user advocates were

- the establishment of regular relationships between a few individuals and users as opposed to a large group of designers. It was expected that few individuals with contact over time could develop a close relationship with trust, continuity, and reliability, which could serve to reduce the users' fears in reporting problems. The more information that one can get from the users, as a result of regular visits, the more comprehensive is the picture of the users. The two user advocates always combined their impressions in order to build a complete model of the application field.
- the introduction of "virtual" non-existent future users to the design team by applying their knowledge of the workplaces and processes and abstracting from the actual users. This was a means to overcome conflicting goals of the users and to take an overall view of the application field, making technical visions and future user requirements discussable within the design team.

In sum, the user advocates took the "offensive". Not waiting for problems to occur, they travelled to the users' worksites on a regular basis to find out how the system was being used. They advised the users not only about the functionality, but also in how cooperation could be done with the system. The user advocates learned about user needs from worksite visits and user workshops, where they also gave users feedback and moderated debates among the users, such as in establishing conventions for using the system. The designers' presence at the workshops was valuable in that the users' feedback confirmed the picture that the user advocates presented.

3. THE DESIGN TEAM PERSPECTIVE

When introducing a new approach to design, the designer is faced with integrating a new perspective with his or her current one. Often a new approach, as in our case, is different from theoretical academic approaches. We discovered that the redistribution of work on the design team, by creating a communication channel to the users via the user advocates, was a totally new way of working for the designers, who were used to either direct communication with users, or none at all. And lastly, the nature of the user advocate and designer roles had inherently different goals, which in some cases created conflicts within the design team. For these reasons, we felt that it was essential that we examine the effect of introducing a new design process within the design team. We felt that the process had the greatest impact on the designers, especially in terms of their motivation in understanding the users' needs. Therefore, we focus on reporting their perspectives.

To learn the perspectives of the design team, we conducted interviews with the designers and user advocates. The interviews were conducted individually, and ranged from 1 _ to 3 hours each, covering discussion over: each one's personal experience with each design aspect, what each personally learned about design from the experience, how closely they felt the design approach was able to match requirements to user needs, the consequences of having different roles and different perspectives on the design team, advantages and disadvantages, timing, and success and satisfaction with the design approach. The project members were very enthusiastic about the interviews, and many commented that it was valuable since it enabled them to think about the design experience in a different light. In many cases, common themes emerged, which may have developed due to common discussion in the design meetings. However, in some cases, some common themes appear to have originated independently.

As we will discuss shortly, it was difficult for the designers perceptually to separate the combined approach into the three separate design components. We begin by presenting the designers' perspectives on osmosis and evolutionary cycling, and then focus on the impact that user advocacy had for them as designers.

The perception of an integrated process

Although in theory osmosis, user advocacy, and evolutionary cycling were distinct, in reality these design aspects became so interconnected through working with the uses that the designers' perceptions of them were as one holistic approach. This was especially true of the designers' perceptions of the osmosis and user advocacy aspects; they were not so much distinct communication channels, but two components of the same wide channel. Both involved users, and they saw it as a process which began with direct contact of many designers with users, naturally flowing to a stage with two user advocates having more direct contact with the users. To some designers, the relationships with users changed from "semi-regular" (with the designers) to "regular" (with the user advocates).

The designers reported that the osmosis experience was valuable because it helped confirm for them the later impressions introduced through the user advocates. Overall, the designers rated the design approach a success since they felt that this progression was reasonable not only from the user point of view (by moving toward permanence in a relationship) but also from the designer point of view, since the user advocates "freed" the designers with their time, so they could begin devoting themselves to working on the design. Lastly, the evolutionary cycling gave the designers a chance to get user feedback and assess how the system worked in actual use. The designers' contact with the users continued throughout the project, moving in one designer's expression from "planned" to "unplanned" osmosis, such as when designers visited the worksite to install the system or solve problems. It was the earlier emphasis on the "planned" osmosis that made it possible for the designers to have comfortable, even at times, joking, intimate relationships with the users later.

The ambiguity of the evolutionary approach

While working, the designers were able to explore the use of the system through the evolutionary cycling, enabling them to adapt the system. To the designers, the users' needs and resulting products were a function over time: what they implemented in the system over time became modified, in some cases eliminated. In *theory*, a Top-Down approach assumes unchanging conditions, and in *reality*, in actual work conditions where the context changes over time, evolutionary cycling came closer to meeting actual users' needs. As one designer expressed:

> It has to do with evolutionary design—there's always constant changes. With a Top-Down design approach, the users accept the design model, what's given to them at the beginning. With our approach, the (initial) model is closer to the users. Therefore we must give support constantly. The users are so used to getting support. Therefore, they ask for it continually. With a Top-Down approach, you give the model once, and it's finished. With our approach, we have a chance to think more about the system and find out more about how the users are using it. Because users are used to asking for requirements, their fear is reduced and they can and will ask continually. With a Top-Down approach, it is more of an abstract level; with ours, it is more concrete.

We found through experience that it is fictitious to think that there is an end-point with the introduction of a system, as some Top-Down approaches might lead people to believe. The actual experience of the ministry showed that external influences, such as organizational restructuring (combining two ministries) changed the context of work for the users, which of course changed the users' needs.

Climbing out of the costumes of computer scientist and user

We learned from this experience that pure roles in the project did not exist. The roles in fact, enlarged for the designers, so that in some sense they also became teachers during workshops, interviewers, or even "ethnologists" via the osmosis aspect. The designers reported

> When one has different roles themselves, for example being in a workshop, I had a different role, then one can easier see different perspectives.
>
> It's fun to do tasks that a computer scientist doesn't normally do, such as learning new fields. I learned what other professionals learn in other fields, for example, social scientists—what they do. The designer is closer connected to the users.
>
> [Describing the process of interviewing]: I found that I did a "predesign" in my brain that happened during the interview when I heard the answers. I imagined how the design would be implemented in the system...With a "predesign" in one's head, you can ask further questions on how to make the design implementation. You can use your design experience to react quickly to ask deeper questions.

By climbing out of the costumes of their respective roles (e.g. Ruhleder, 1995), it enabled a mutual learning process to take place between designers and users. For example, we see this when the users took the initiative to raise design issues, such as discussing access rights and security with respect to signing documents. As a result of this process, it

- increased the understanding of the users and designers (see e.g. Briefs, Ciborra, and Schneider, 1983)
- created a meta-understanding of the design
- deepened learning about the context of the system (Gärtner and Wagner, 1994)
- strengthened members' functions as specific information providers
- helped "market" the project to different audiences

4. EXPERIENCES WITH USER ADVOCACY

All the designers agreed that the user advocate approach was effective, and different points were emphasized. Some designers felt that by having people who had permanent relationships with the users, they did a better job of eliciting requirements than the designers would have. Many of the designers commented that they developed better intuitions about the users' perspectives by hearing them through the user advocates. The different points can best be expressed directly from the designers'comments:

> The user advocates had a serious impact. The users would take them more seriously [than the designers]. They were able to have a trusting relationship with the users. I think they [the users] would be more careful with me. They [users] might even laugh at the designers if we would bring up certain requirements. The ideas of the functionality were evaluated higher from the user advocates, as they would have been with the designers.
>
> I would not have enough patience to make sure that the user was satisfied.
>
> I would be too eager too please, because of my own personality, and I would let other issues slide.
>
> The whole concept of the user advocate is that they're integrated into the group. The problem stays "hot"—it's reported right away. It's solved faster, as opposed to spending time in writing it up. It's a faster and shorter way. Also, the quality of reporting the problem is higher.
>
> There is the danger that users would want what they are used to. For example, the circulation folder. They [users] are used to that, and they of course wanted that. For a totally new way, they would be reluctant to want it. It's more difficult for the designers to bring up new ideas about functionality [than for the user advocates].
>
> The user advocates had more positive attitudes and patience with the users than the developers did. The developer is confident with the system. They would not want to hear criticism.
>
> The user advocates wanted to prevent traps and misuse of the system. It made the design more flexible.

A new division of labor

User advocacy introduced a new division of labor to the design team. The result of this redistribution of work had several impacts.

1) *It saved the designers work.* By having the user advocates gather design requirements and in turn present the design decisions to the users, it enabled the designers to concentrate on their tasks of implementing the design.

> The user advocates—we have people who are really dedicated. They do work that's bothersome to designers. For example, you can imagine if for every problem, a designer would have to go. If the users know, for example, that a certain designer implemented X, then they don't call him directly, but go to the user advocates.
>
> With the user advocates, they took over tasks that I would have to do. They did the first analysis, and summarized the users' viewpoints. This saved me work.
>
> It's easier to delegate things. One user called me with a problem once, and I answered that the user advocate will solve the problem on the next visit.

2) *The roles of the user advocates expanded.* Since the division of labor enabled user advocates and designers to devote more time to their roles, they became more specialized. By being able to concentrate solely on developing relationships with the users, the user advocates' roles as "user representatives" deepened and expanded from what was initially envisioned. For example, the user advocates are often used as an information resource for designers on how processes are going on in the ministry or what is planned for the future. One designer explained that to know how something really works in the ministry, they go to one of the user advocates. Rather than deal directly with trying to understand the complex work methods themselves, the designers report that they can go to the user advocates to find out how these rules really operate. Still another role was that of becoming "marketing people", by having to "sell" the designers' ideas to the users.

3) *The responsibility for actions was not clear.* To the designers, it was not always clear where an idea originated, or where responsibility for design decisions should lie.

> If there's a problem, then the users know that the user advocates are not responsible.

This comment by one designer raises the larger issue, of namely, when working with user advocates on a design team, who is responsible for what? The designers reported that by having user advocates, the responsibility to the users was at times alleviated because they did not have direct contact with the users. It is our speculation that this division of labor may also diffuse the responsibility for design issues among the team.

The relationship of user advocates and designers: an understandable separation

The reality of combining different roles was that this sometimes led to a polarization of the group. The tension involved with participatory design is that by its inherent nature, it takes two sides into account: on the one hand, enhancing the potential of the technology, and on the other hand, satisfying the everyday needs of the user. These two aspects of the project became manifest in the roles of the team: the goal of the user advocates was to fulfill the user needs while the goal of the designers was to implement innovative design concepts. This tension is expressed in one designers' criticism

> It's not so easy to reach a decision because the user advocates have a strong way of pushing requirements through—they're doing their job too well. There's a saying—it [a design implementation] works well until someone tries to use it.

The persistence of the user advocates with design requirements was interpreted by one designer as creating an artificial separation between the user advocates and designers. This separation became manifest in two ways: 1) prior beliefs developed on both sides about how the design proposals would be reacted to, and 2) misunderstandings occurred about reasons for why people were against ideas. Some designers expressed this as:

> The user advocates thought that if others were against an item, then the designers were not taking them seriously. More often the designers would propose an idea and the user advocates would be against it. It was also the other case. The user advocates would propose something—it was often on a high level—and the designers would say that it's not technically possible. It's good though to have theoretical proposals, because it works against tunnel vision.

> Sometimes the user advocates would ask three designers about something. If the first two would say no, they can't implement it, but the third person would find some way around it, and find a solution, then the user advocates would think that the first two were just blocking the solution.

The different roles and the different perspectives sometimes led to the problem that a core issue for the user advocates played a trivial or even irrelevant role for the designers or vice versa. The user advocates, for example, were strongly convinced that the users have to rely on the software manual. After one year and several changes of special software features they asked for a revised handbook which should be in accordance with the actual system functionalities. The designers at first did not want to invest time in the revision because for them manuals had a low priority. The user advocates finally convinced them about the importance of a manual for those who are not so experienced with computers.

In reflection, these comments are actually an indication to us that these roles were being taken seriously. Both ends of the design goal spectrum were considered. In the second quote, the user advocates see this as showing persistence on their part. They were also applying their technical expertise here; from their own experience they found that a technical problem sometimes is difficult for one person to solve, but that the chances are higher that in a group, someone might be found to know the solution. They simply kept asking designers thinking that they might find someone for whom the technical solution would be clear.

Hindering innovations and missing the goal

In the first phase of the project, an emphasis was placed on refining the system. Based on the user needs reported by the user advocates, a new system version was designed and implemented even before its scheduled introduction date in the project plan. The designers rated personal satisfaction with the system as very high. However, since the system version was user-driven, a number of small and minor design issues were brought up to smooth the existing tools, rather than raising needs for new concepts and tools. Related to the previous idea discussed, we see the conflicting goals of the designers and user advocates surfacing here. The most common problem identified by the designers with respect to the combined design approach was exactly this phenomenon: too many small problems needed to be solved. This had four consequences for the designers:

1) *Small problems led to an expense of time and energy for the designers.* By being so intimately involved with the design process, the users raised many small issues concerning the design. From the user perspective, being able to bring up even the smallest problems about the system would be read as an advantage. But as one designer said, "low-level technological problems are bothersome—no one wants to do them." The designers often could not understand why the small problems should be solved. To a designer, they would not be problems—there is always a way around to find a solution. It was very often because of the persistence of the user advocates that the small problems would get attention and get worked on. The user advocates would explain the users' side, and defend the importance of fixing this problem for the user.

2) *Solving the small problems took time away from more conceptual, or scientifically interesting work for the designers.* The designers on the project were actually faced with two goals, which conflicted. On the one hand, they wanted to fulfill the project goal; on the other hand, their

individual goals were to make a scientific contribution. The small problems escalated this conflict for the designers. Most designers complained that they felt they were doing support work instead of research and serious development. One designer commented that the project fell short of its scientific goals because of the design approach. Another would have preferred to "expand concepts", but couldn't because other priorities existed because of the users. Designers blamed "the hindering of innovations", and not being able to "concentrate on my work" due to the small problems.

3) *Often the common goal of the project was suppressed.* Concentrating on solving the small details often resulted in the design team "missing the forest for the trees". One designer commented that one often feels they are doing things for others on the design team, and forgets about the larger goal. In this sense, specific task goals of team members sometimes took priority over the project goals. Another designer explained that priorities in the project changed on short notice, such as when problems in the ministry had to be solved immediately. These events all contributed to distracting the designers from focusing attention on the larger, common goal of the project.

4) *Contradictory requirements from the users occurred.* By creating an atmosphere for the users where they felt free to bring up even small problems, ironically, the large collection of problems sometimes resulted in contradictory requirements from the designers' point of view. One designer attributed this problem to the team being too "democratic"; requirements may solve one user's problems, but may create other problems for other users. To this designer, the process was event-driven. He felt that a Top-Down approach would have avoided this, because final decisions would be made instead of an open-ended on-going negotiation and bargaining process.

User advocates as subjective filters

By hearing the users' problems reported at design meetings via the user advocates, and not hearing the problems direct from the source, the designers reported that they often failed to understand the roots of the users' problems. There was also the suspicion raised by the designers that many of the problems were ideas that came from the user advocates, and not from the users. Others claimed that the user advocates were "filtering" the ideas of the users, rather than reporting them directly. The idea of filtering information emphasizes the idea that user advocates are individuals, with individual personalities, and will always be "subjective interpreters" of the users' problems.

> I have the impression that some ideas came from the user advocates and not from the users; therefore I became resistant to some ideas.

> A disadvantage was that the user advocates selected the problems of the users. They were a filter. The user advocates reinterpreted the user requirements.

> The user advocates had certain preconceived ideas about the design requirements, and this was reflected in their presentation to the designers. The requirements were more representative of their ideas than the users. Of course this was unintentional, but it was really noticable. They were not the pure ideas of the users.

> With user advocates—we don't see the roots of the [system] problems with the users.

> When you see many problems with the functionality, that you yourself don't see as problems, then you imagine they were not problems of the users, but of the user advocates. With this long list of user requirements, the priorities that were given were personal impressions [of the user advocates].

> I had the impression that some requirements were not necessary. Designers are used to the system and can get around it. Designers can survive with many problems.

Ironically, the idea of "filtering" ideas of the users was part of the original reasoning of using the user advocate approach. Because of their knowledge of computer science, the user advocates intended to better represent the users' wishes to the designers by applying their expertise. The same problem of subjectiveness occurs in discussing guiding visions, future solutions, and planned innovations with the user advocates.

User advocacy: situation-driven?

In sum, despite the problems associated with the design approach, all the designers overwhelmingly rated the design approach a success, and rated personal satisfaction high. They agreed that the advantages outweighed the disadvantages. In fact, to emphasize this point, one designer claimed that all the disadvantages could also be viewed as advantages, depending on one's perspective.

The designers expressed a latitude of opinions as to whether using this approach changed their thinking or philosophy about design, ranging from not at all to being a "born again" participatory designer. A common opinion expressed toward the osmosis aspect was discovering that "the world of the user was totally different than the world of the designer". Many designers expressed that they developed a strong tendency to think about what the user needs and how the user uses the system. For one newly committed convert, he would not do a project again without involving users, and in fact, now assigns his students to involve users in the design.

It should be pointed out that many of the designers' perceptions about the user advocates are not unique to this project, nor to that of an information technology context. Although user advocates have additional functions to mediators, similar perceptions about mediators' roles have been reported in other contexts, referred to as negotiator cognition (Bazerman, 1990). The point of considering these events in a larger context is that many of the problems that occurred we interpret as situation-driven, and not only unique to the personalities or project context, although personalities certainly play a role. That is, by placing people in a context where roles have conflicting goals, it is not unusual to expect that some of these interactions would occur. One suggestion that a designer gave as a method of overcoming such problems, is to rotate roles within the project. This is an intriguing idea of interchanging designers, user advocates, and social scientists, and holds merit since many of the designers reported the value to them of learning about others' perspectives.

5. SUMMARY AND OUTLOOK

Participatory system design has been discussed since several years (for an overview, see Bjerknes, Bratteteig 1994); case descriptions, theories of different aspects, and metastudies exist (van den Besselaar, Clement 1993). Although widely discussed, participatory design with its different facets and niches is evolving, continually raising new questions, and requires solutions due to changing time and context, e.g. the changing workforce structures, concepts of design like rewarding work instead of salary or job security, or driving forces in (self-) organizations (Mambrey, Paetau, Prinz, Wulf, 1996).

It is too early for conclusions about the benefits for the users, organization, product, and so on, because the development of the procedures of the design process is still on-going. The processes of osmosis, user advocacy, and evolutionary cycling are expanding and still forming for us. However, one indication of success of this approach is that an intermediary version of the system was introduced in February 1996, which was not originally planned in the proposal. In this version, a long list of user modifications were implemented.

In this paper, we tried to stay close to the empirical level. Our idea was to report on applied ideas for participatory design from the designers' perspectives because we believe that the division of labor in large scale development projects and the current interactions of multiple perspectives in design teams are one important clue to understand the dynamics of system design. Our future plans are to continue to evaluate the design process including the user perspective as well, once they have the opportunity to experience the new version over time.

ACKNOWLEDGEMENTS

We thank the design team for their participation in these interviews. They expressed their experiences with a deep commitment to the project. We also thank the project team for their valuable comments.

REFERENCES

Ballay, J. M. (1994). Designing Workspace: An Interdisciplinary Experience. In B. Adelson, S. Dumais, and J. Ohlson, Human Factors in Computing Systems, CHI 1994, New York: ACM Press, pp. 10-15.

Bazerman, M. (1986). Judgement in Managerial Decision-Making. New York: Wiley.

Bjerknes, G., Bratteteig, T., (1994) User participation: A strategy for work life democracy? In: Proceedings of the Participatory Design Conference '94, Chapel Hill, North Carolina, USA 27-28 October, 1994, pp. 3-12.

Bowers, J., (1994) The Work to Make a Network Work: Studying CSCW in Action. In: Proceedings ACM 1994 Conference on Computer Supported Cooperative Work October 22-26 Chapel Hill, North Carolina, USA, pp. 287-298.

Briefs, U., Ciborra, C., Schneider, L. (eds.) (1983) Systems design for, with, and by the users. North Holland Amsterdam 1983.

Clement, A., van den Besselaar, P., (1993) Participatory Design Projects: A Retrospective Look, Communications of the ACM, Vol. 36, #6, 1993, pp. 29-37.

Floyd, C., Keil, R., (1982) Softwaretechnik und Betroffenenbeteiligung. In. Mambrey, P., Oppermann, R., (eds.) Beteiligung von Betroffenen bei der Entwicklung von Informationssystemen. Campus Verlag Frankfurt and New York 1982, S. 137-164.

Gärtner, J., Wagner, I., (1994) Systems as Intermediaries: Political Framework of Design & Participation, In: Proceedings of the Participatory Design Conference '94, Chapel Hill, North Carolina, USA 27-28 October, 1994, pp. 37-46.

Hoschka, P., Butcher, B., Streitz, N., (1993) Telecooperation and Telepresence. Technical challenges of a government distributed between Bonn and Berlin. In: Information and the Public Sector, 1993, 2(4), pp. 269-299.

Klöckner, K., Mambrey, P., Sohlenkamp, M., Prinz, M., Fuchs, L., Kolvenbach, S., Pankoke-Babatz, U., and Syri, A., (1995) POLITeam - Bridging the Gap between Bonn and Berlin for and with the Users. In: Proceedings of the Fourth European Conference on Computer-Supported Work - ECSCW '95, Stockholm, Sweden, Sept. 1995, Kluwer Academic Publishers, Dordrecht, 1995, pp. 17-31.

Kyng, M. (1994). Experience with Participative Application Development. In K. Brunnstein and E. Raubold (Eds.) Applications and Impacts: Information Processing '94, Amsterdam: North-Holland.

Mambrey, P., Oppermann, R., Tepper, A., (1987) Experiences in Participative System Design. In: Docherty, P., Fuchs-Kittowski, K., Kolm, P., Mathiassen, L., (eds.) System design for human development and productivity: participation and beyond. North-Holland Amsterdam et al. 1987, pp. 345-358.

Mambrey, P., Paetau, M., Prinz, W., Wulf, V., (1996) Self-Organization of Social Systems - A new Challenge for Organization Sciences and System Design. In: SIGOIS Bulletin, VOL. 17, No. 1 (April 1996), pp. 4-6.

Ruhleder, K., (1995) Involving the "Users": Blurred Roles and Co-Design Over Time. In: SIGOIS Bulletin Vol. 16, No. 2 (December 1995), pp. 14-15.

Schein, E. H. (1988). Process Consulation: It's Role in Organization Development. Reading, MA: Addison-Wesley.

Sohlenkamp, M., Mambrey, P., Prinz, M., Fuchs, L., Syri, A., Pankoke-Babatz, U., Klöckner, K., Kolvenbach, S., (1996) Supporting the Distributed German Government with POLITeam. 1996. Forthcoming in International Journal of Multimedia Tools and Applications-Special Issue on Multimedia Projects in Government.

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