From Sandbox to "Fundbox": Weaving participatory design into the fabric of a busy non-profit

Randall H. Trigg
The Global Fund for Women
425 Sherman Avenue
Palo Alto, CA 94306 USA
+1-650-853-8305
trigg@workpractice.com

ABSTRACT

This paper documents the experience of doing participatory design in a small non-profit staffed by an empowered workforce. The first eight months of database design and development included close collaboration and an effective division of labor with a member of staff, a collaboratively conducted training workshop, and prototyping in an on-site "sandbox." The sandbox was continually available for each staff member to explore and tinker with, both in one-on-one sessions with me and on their own. In addition to describing several successful cooperative designs, the paper explores ongoing efforts to fit the work of participatory design into the staff's busy schedule. Their time constraints led me occasionally to postpone group planning and decision-making, and more generally, to confront the tradeoffs between short-term orientations of participatory design to the needs of particular individuals and longer-term orientations to group design and strategy sessions.

Keywords

Participatory design in non-profits, database redesign and development, participatory design in busy workplaces.

INTRODUCTION

Participatory design teaches us to listen, stay flexible, participate and encourage participation whenever appropriate and possible, be respectful and accountable, report back to all concerned parties, and encourage mutual learning. A worthy set of principles indeed. But what demands do they place on our busy co-participants in a workplace? How can we weave the work of participatory design of new technology into the schedules of workers who are invested and engaged with the project's goals, but all but swamped with their own work?

For the last year, my site for participatory design has been a non-profit called The Global Fund for Women (GFW) with a small empowered staff.[1] During the first eight months, we conducted a project of database redesign and development, which we agreed from the start to base on a participatory design approach. Because GFW's staff places a premium on mutual respect across and within organizational boundaries, the fit seemed to be a natural one. This paper is the story of the start of that project, the work of which is ongoing.

In *PDC 2000 Proceedings of the Participatory Design Conference.* T. Cherkasky, J. Greenbaum, P. Mambrey, J. K. Pors (Eds.) New York, NY, USA, 28 November - 1 December 2000. CPSR, P.O. Box 717, Palo Alto, CA 94302 cpsr@cpsr.org ISBN 0-9667818-1-3

In one of the few published papers on participatory design in non-profits, Margaret Benston writes:

> Non-profit groups typically operate with little money and with already enormous demands on staff and volunteers - one of the major problems in attempting to work with them in a participatory way lies in their members finding the time for an approach that at times simply seems another burden. It's hard for over-stretched people to be future-oriented enough to recognize that the time put in during early stages will be repaid later. My experience has been that there is a resistance to the need to learn enough to even begin the design process. There is a strong tendency to want the expert to do it for them. (Benston, 1990, p. 107)

My experience confirms the conditions Benston describes, but not her conclusion. At GFW, a lack of time does not indicate resistance to the participatory design process. On the contrary, my co-participants are committed to learning how to improve their technology infrastructure and full of suggestions for how to do it. Our joint task has been to conduct the process in a way that fits into their taxing daily schedules and seasonal work cycles.

After placing this project in the context of earlier participatory design work, I describe the setting and outline the chronological course of my involvement. To ground the closing discussion, I discuss four instances of participatory design: revising one database field's value list, semi-automating the production of outgoing correspondence, automating the calculation of status codes, and maintaining database integrity. I then return to the problem of conducting participatory design in the face of staff's busy schedules, and outline the approaches we've explored. Our experiences raise

[1] During the first six months of this project, I was a volunteer at GFW. In March 2000, I joined staff in the position of Research Analyst.

a variety of questions: When is it worth calling a meeting to prioritize a set of tasks? When is it practical and effective to work with and address the needs of individuals rather than the group as a whole? Can a participatory design process make headway in the absence of, in advance of, or in the midst of longer-term group decision-making processes? Finally, what is the role of food in all this?

RELATED WORK

My work with the staff of GFW fits into the classic framework of participatory design by engaging members early and regularly in the design process, by recognizing and valuing differing forms of expertise, and by supporting mutual learning. At the same time, this project occupies several less populated corners of the space of published participatory design projects.

- *Which work sites?* Participatory design projects have traditionally been conducted in governmental institutions and commercial organizations. Only rarely do we venture into the growing "third sector," the world of non-profits and NGOs. Two notable exceptions are found in the work of McPhail, Costantino, Bruckmann, Barclay, & Clement (1998) and Benston (1990).
- *Where do the designers call home?* Most projects are conducted by means of visits to the work site where interviews, workshops, prototyping sessions, and the like are held. Seldom are the designers themselves part of the workforce at the site. Or what is perhaps more likely, such in-house participatory designers rarely write about their experienes for academic conferences and journals.
- *Where are the prototypes housed?* By the same token, most prototypes described in the literature are housed off-site between cooperative prototyping sessions, rather than being continually available for exploration and experimentation by the workers.

My project at GFW aligns with an ongoing effort in the participatory design community to move from participatory design as research project toward participatory design as an insider's practice of choice, namely "on-site" participatory design. According to Finn Kensing and his colleagues this shift is underway: "In working with managers and IT professionals during most of the 10 projects contributing to the development of our method [MUST], we have experienced an increasing awareness of the pitfalls in the predominant practice as well as a willingness to experiment with alternatives" (Kensing, Simonsen, & Bødker, 1998a, p 267). Like them, I am exploring what difference it might make for an organization's IT staff, volunteers, and consultants to adopt participatory design.

Brenda McPhail and her colleagues describe a participatory design project with a non-profit called CAVEAT located in Toronto, Canada (McPhail, et al., 1998). Like our project, the CAVEAT team was engaged in the redesign of a Filemaker-based database system and confronted many of the same technical problems. In addition, the staff of CAVEAT shares some of the values of GFW, relating as peers and "working towards a common goal" (McPhail, et al., 1998, p. 224).

GFW has fewer volunteers as a fraction of the total number of database users than CAVEAT. Partly for this reason, everyone I worked with was intimately familiar with the content of the database. Similarly, the diversity of computer backgrounds at GFW was not as extreme. Most of the staff were quite adept at browsing, searching, and modifying data in the database, and many were also accomplished at composing and modifying layouts for correspondence and reports. This difference may be due to the intervening decade of growing technological sophistication among non-profit staffs.

Like McPhail et al, we collaborated especially closely with one staff member during the project. The CAVEAT project team's "insider" from the non-profit went on to coordinate installation of the prototype as CAVEAT's working system (McPhail, et al., 1998, p. 232). In contrast, my primary collaborator at GFW felt free to resume her full time grantmaking duties once the redesign project was complete, in part because of my continuing presence.

Toni Robertson describes participatory design in small companies that are similar to GFW-like non-profits (Robertson, 1998). Her companies also have small staff size, no IT department, and make use of off-the-shelf software that must be tailored in house. Robertson identifies the reduction of system design to shopping as a characteristic phenomenon in such companies. In my case, the shopping phase, at least for software infrastructure, was complete. Having bought a database system, Filemaker Pro, and mastered its basic features, GFW was now looking for help in learning about Filemaker's customization tools, rethinking the architecture (or thinking about architecture for the first time), and in general, better aligning database design with day-to-day work practices.

Robertson argues on the one hand that small companies are characterized by separation between situations of use and design because they use off-the-shelf software and lack resources to hire IT staff. On the other hand, she argues that small company employees tailoring off-the-shelf software are at the same time users and designers. Their capacity to modify the software is constrained by the available tailoring tools, their programming skills, and the lack of resources to gain more. This was just the situation at GFW when I arrived. I sensed an eagerness to learn more about the tailoring tools of Filemaker and explore the possible improvements they could enable. Seemingly, GFW was an ideal environment for participatory design.

A BUSY NON-PROFIT

The Global Fund for Women as described in its mission statement is "a global network of women and men committed to a world of equality and social justice" that "supports a wide range of initiatives that protect and defend the human rights of women around the world." As a foundation GFW provides grants of up to $15,000 each to women's groups outside the United States. Since its founding in 1987, the Global Fund has given grants totalling some $16 million. In the last fiscal year alone, GFW gave away more than $3 million. (More up to date information is available at the GFW web site at www.globalfundforwomen.org.)

The computer infrastructure at GFW's headquarters in Palo Alto, California consists of Apple Macintoshes for most of the 15-20 staff members, plus two PCs running QuickBooks for the finance group. The Macintoshes are all running Microsoft Office and Netscape along with Filemaker Pro. There were several databases in place at the start of this project; the three in regular use by most of the staff cover grantmaking, foundations and corporate donors, and individual donors. Of these, the grantmaking database was the primary focus of my initial work. In the rest of the paper, "the database" refers to the grantmaking database, and "the staff" to the grantmaking team.[2]

A couple of years prior to my arrival, database consultants were hired to address specific problems. Although the facilities they built were workable at least for a time, there was little follow-through. Indeed, the database contains several of the consultants' scripts and fields whose rationale remains a mystery to me and to the current staff, most of whom were hired since the time of the consulting work. Staff members expressed dissatisfaction at not learning enough about the consultants' tools to be able to incorporate them in daily practice. As a result, they resolved that at least one member of the group would always have an understanding of any new tools added to the database. This is an important sense in which the GFW workforce is empowered around technology.

PROJECT HISTORY

In the fall of 1999, I started working at GFW as a volunteer 15-20 hours a week. I was initially assigned to the then Africa and Middle East program officer, Jennifer Astone, who had just joined the staff and whose vision of a more effective database was crucial to getting this project off the ground. This section outlines my work with Jennifer and the rest of staff to better understand GFW's workplace and work practices, while exploring the technological possibilities she and others were contemplating.

Participatory analysis

My first weeks were spent talking with staff about their work with and without the database, analyzing the current architecture, and learning Filemaker. In return for the valuable time I was getting with staff, I passed along tips I learned for more effective use of Filemaker, answered general questions about their computer environments, and offered advice on technology strategies. Meanwhile, I was learning how impressive the staff's own use of Filemaker was, in particular their competence in designing layouts for letters, memos, labels, and reports.

Over the first month or so of meetings and interviews with the members of grantmaking staff, I learned about several problems with the database.

- On the one hand, there were too few access points. A single "master" layout was used for search, entry, browsing, and records processing. Its large size required continual scrolling.
- On another front, there were too many access points. There were over a hundred layouts for reporting and correspondence, many of which were redundant. Because the layouts were only accessible via a single pull-down menu, they were difficult to navigate.
- Staff sometimes needed to go "offline" into Word to complete correspondence started in Filemaker. The resulting letter text was thus not available for subsequent Filemaker searches.
- Extra work was required to generate personalized correspondence from standard template layouts. Each change to the template had to be undone so that the next user continued to see the standard layout.
- The staff felt a need for more sophisticated reporting, including sub-totalling.

As often happens, GFW had grown beyond a simple "flat file" database architecture in which all fields are forced into the records of one large table (McPhail, et al., 1998). This meant, for example, that a client organization's contact information was repeated in each proposal record from that organization. In fact, there was no reliable way to count the number of different organizations in the database given that locations, names, and contact people change over the years. Also the geography embedded in the table was too rigid – it was difficult to restructure the country/region divisions to reflect structural changes in the GFW workplace. Both of these problems pointed to the need for a relational architecture, including separate tables for proposals and organizations and for geography.

Prototyping in an on-site sandbox

Approximately one month into the project, we started a "sandbox" for experiments with new database ideas. Like other case-based prototypes (Blomberg, Suchman, & Trigg, 1996), the sandbox contained some of their own real data, but configured in new ways. Unlike many prototypes however, it was located in their space, i.e. continually available from their workstations' networked "sharing folders." The staff explored the sandbox in parallel with their continued intense use of the existing database.

We seeded the sandbox with a sample of 172 records out of some 8500 that comprised the grantmaking database at

[2] At the time of this project, the Grantmaking team consisted of three program officers, one program associate, and one program assistant.

that time. We separated the fields of these records into distinct tables for proposals, organizations, and geographic divisions like countries, regions, and program office areas. The new geographic tables enabled the periodic shuffling and reassignment of countries to regions and regions to program officers that was necessary to reflect GFW's changing organization structure and evolving world geography and politics.

The sandbox also included numerous interface changes similar to those described by McPhail, et al (1998), primarily the use of buttons to move between layouts, and the use of layout "modules" corresponding to the major steps of the grantmaking process: enter, decline, pursue, evaluate, award, and close. The layouts comprising each of these modules included a replicated banner displaying fields that characterize the proposal. We also assigned buttons with understandable names to commonly used correspondence and report layouts. This helped staff avoid the unsorted and almost unmanageable menu of more than a hundred layouts. Finally, we adopted a few conventions around color and button shape, although the database cannot be said to have a consistent look and feel.

During this time, a division of labor emerged between Jennifer and me. We wanted to increase the alignment between database and grantmaking work practices, and decrease the inefficiencies caused by double work and a lack of integration. I was focusing on questions of missing functionality and unused Filemaker features, while Jennifer was especially oriented to the need for readable layouts and ease of access to commonly used layouts and fields.

We co-led a workshop some two months into the project in a way that reflected this division of labor. Our goal was to inform the grantmaking team of our activities, obtain feedback, and begin the process of ongoing training. Jennifer directed a discussion of user interface issues using examples she had created in the sandbox. I led a discussion of architecture including a grounded overview of the concepts of relational tables, and an introductory look at computation in the form of scripts and calculated fields.

For several months following the workshop, we realized more of our design ideas in the sandbox. I continued to answer questions and offer help with the current database, but as often as possible, I employed the sandbox in one-on-one sessions. Our goal was to help staff evaluate the potential benefits of switching over, that is, of installing the architecture and interfaces from the sandbox.

Launching the new database

The staff members were aware of the tradeoff between the benefits of new features and interfaces on the one hand, and the disruption of work and the need for training that the changes would entail, on the other. Over the course of the winter, the staff as a whole felt that the benefits outweighed the disadvantages, and, picking a relative trough in the wave pattern of their work, we planned a launch of the new architecture into the grantmaking team's daily practice. Not only was staff in favor of the launch, but among some members there was a sense of urgency, wanting to use features in their everyday work that they had explored in the sandbox.

In order to ease the transition to the new database, which Jennifer had dubbed the "Fundbox," we held individual interview/training sessions with most of the staff. The launch took place over several days with bug-fixing continuing over the course of the next few weeks. Eventually, I began adding new features, as staff gradually familiarized themselves with the new database.[3] We retained the old layouts so that staff could move into the new interfaces at their own pace. We also made the old database itself accessible on the server with read-only access. This allowed staff to review information in its former state, browsable using the old layouts.

Around this time, I joined staff part time at 25 hours/week. My personal "ownership" of GFW's causes has increased as a result, while my work style remains as participatory as possible. Effectively, I've gone from a volunteer helping them to realize envisioned changes to being their first staff person officially charged with IT responsibilities.

EXAMPLES

The following four examples of on-site participatory development raise questions concerning the role of group decision-making in the participatory design process, the process of automating various common practices, and the problems of ensuring database integrity. The examples provide the basis for the discussion that follows.

"GFW Issues"

An important topic for any participatory design project is how decisions are made. The decision-making process is especially relevant when the changes in work practice are fundamental, affect all members of a group, or imply a shift in the general flow of work. Interestingly, technological changes required for radically different work practices may entail little in the way of design or implementation work. (At other times, the opposite is true – changes that are easily agreed upon pose subtle design and integration requirements.)

A case in point is the "GFW Issues," a list of terms used to characterize a proposal under consideration. In the old database, each proposal could be coded with any subset of the issues, simply by selecting from a panel of checkboxes. Technically, this coding interface worked well. However, some on staff were questioning the utility and effectiveness of the issues themselves. Was there redundancy? Did some issues subsume others? There were also doubts as to their appropriateness. Did the list continue to reflect the content

[3] One downside of leaving the sandbox is that it became harder to make rapid changes. The FileMaker Pro server prohibits defining new fields without interrupting service of the database. I began saving lists of field changes for the next evening or weekend when I could bring down the database with minimum disruption to the staff.

of submitted proposals? Were the issues useful to staff members at GFW outside of grantmaking? Finally, could two proposals coded with the same issue be comparable if the first proposal was also coded with 10 other issues, while the second was coded with just the one?[4]

In order to revise the list of legal values for this field in the database, I would only need to write a simple script mapping old values to new ones to be run as part of the launch process. But the changes required significant mobilizing within the group. Jennifer took the lead, organizing several meetings to discuss the meanings of the issues, their uses, and their interrelationships. Several staff members divided up the list of the old and new issues and wrote up definitions and justifications for each subset. In addition to finalizing a new issues list, they restricted the maximum number of issues chosen for any proposal to three in an effort to increase the relative value of each coding. Technically, changing the interface to allow only three issues was not hard, although the change in the way the group had been working was significant. Eventually, the group agreed on a new list of issues, which we later incorporated into the launch script.

This example underscores an old lesson from participatory design, that technological questions are rarely at the centers of the worlds of our co-participants. As we'll see, I had ample opportunity to relearn this lesson over the course of the project.

Outgoing correspondence

The staff revised the GFW issues through a series of group meetings. The next example involves a different process, one which iteratively changed technology and work practices in the absence of a decision process by the group as a whole.

The example involved building a tailorable facility for semi-standardized correspondence, where by "semi" I mean that each generated letter or memo can be personalized before printing. Our experience building correspondence templates in Filemaker bears a resemblance to the participatory design of WordPerfect letter templates in the AT project conducted at Aarhus University (Trigg & Bødker, 1994). There, the management of the institution took an active interest in the proceedings; indeed their interest in the tailorability of WordPerfect seems to have been partly motivated by the chance to increase the degree of standardization. At GFW, the staff came up with the idea of semi-automating certain forms of correspondence in the hopes of reducing their burgeoning writing workload.

The staff sends many forms of correspondence over the course of handling and evaluating a proposal: an immediate acknowledgement, a "question letter" asking for more information from the submitter, an endorsement letter soliciting appraisals from selected members of a world-wide council of advisors, a decline letter conveying regrets and pointing to other sources of funding, a contract letter congratulating the organization and outlining the conditions for the award, and a letter acknowledging receipt of the final report, among others. Originally, each form of correspondence was based on a form letter template that was automatically filled in with address, organization name, proposal number, and the like. This scheme had two problems. First, staff members often needed to personalize the letters for the case at hand, perhaps reflecting a personal relationship with the group's contact person, or a history of previous grants to that organization. The staff member had to modify the template itself, and then undo the change so the next instance of that type of correspondence for a different proposal would continue to start from the generic template.

The second problem was that correspondence was not stored in the database. In most cases, the staff saved only the hardcopy, which was often out of reach when someone from the organization suddenly telephoned from halfway around the world. Some staff members copied the letter templates into Microsoft Word and filled them in manually. They saved them only on their own hard disks, again making the documents difficult to retrieve in a hurry.

Around the time of the launch, we realized that Filemaker could support template-based correspondence that was personalizable and where the resulting letter could be stored in the database. We also envisioned an historical record of all outgoing correspondence regarding a particular proposal. Eventually, we did develop just such a facility, currently operational. But how did we get there and how were design decisions made?

One school of thought held that staff needed to meet to discuss such a template facility, both around its design, and as an occasion to standardize templates for the organization. On the other hand, some staff members needed certain forms of correspondence automated right away and prefered not to wait for completion of a series of meetings.

In the interests of moving forward, I advocated an incremental approach. When an individual who was the primary author of certain forms of correspondence designed a template and identified the dimensions along which its text varied (e.g. the recipients' native language, whether a grant was a renewal or was multi-year, whether the amount awarded was less than that requested, etc.) then I built support for automating, storing, and personalizing that particular type of correspondence. By separately orienting to and then generalizing from several individuals' needs, I made the design tailorable enough so that the template facility's "point person" (GFW's then program associate, Annie Hillar) could modify templates, create new translations of optional texts, and the like, without having to modify Filemaker scripts or create new fields.

[4] Trigg, Blomberg, & Suchman (1999) describe the design of a system for coding scanned documents with topic keywords. There, the coding facility was being installed for the first time albeit by means of several iterations of the design. In contrast, the issue codes revision described here exemplifies a later stage of a long-term redesign process (Braa, Bratteteig, & Øgrim, 1992).

Each successive instance demonstrated anew the value of the facility for others in the group, while still allowing correspondence to be drafted in the old style. By starting small, but doing so in a generalizable, tailorable way, the facility got off the ground in parallel with discussions of its adoption. At the time of this writing, most of the dozen or so forms of correspondence are supported under the new scheme.

On a more cautionary note, though tailorable, the facility was the most complex I'd yet implemented in Filemaker. It concerns me that the underlying scripts are inscrutable to staff. This argues for further training, but again, who has the time?

Status codes

We explored a somewhat different form of automation in the next example. GFW's "status codes" identify the stages through which a proposal passes. These codes are numbers from 1 to 9 together with a few words of explanation. For example, new proposals under review for which correspondence has yet to be sent are assigned the code "1 - UR/under review or awaiting translation." "5 - UR/likely to be rec." indicates a proposal that will be recommended to the board for approval. "7.12 – G/paid (1 of 2 payments)" signals an awarded multi-year proposal one of whose two payments has been made. "9.3 – award withdrawn" indicates a proposal whose award was withdrawn for some reason. Originally, staff shifted one code to the next manually as the proposal moved through the various stages of processing. But problems of consistency arose between codes and implicated fields. Suppose a proposal shows status 7, but its Date Paid field is empty. Which field is to be trusted?

This problem of inconsistent indicators led to the idea of automating the status code, making it what is called in Filemaker a "calculated field," one whose value depends on the values of other determinant fields. For example, whenever the DateDeclined field was filled in, the proposal's status code would instantly switch to "0 – declined."

The potential benefits seemed clear; no longer would staff have to remember to reset the status field when they filled in a date, and there would hopefully be an end to one source of inconsistency in the database. The costs were less clear at the start, but emerged over time. For example, the group needed to decide for each status code which fields would be its determinants. In some cases, new fields had to be created as indicators of a given status, and occasionally those fields might interact. Suppose a proposal has a Reason for Decline filled in, but no Date Declined? Or an Amount Awarded, but no Date Awarded? Over time, discussions resolved these and other similar dilemmas. For example, the former leads to status "Likely to be declined" and the latter indicates a proposal whose Amount Awarded could appear in a financial report generated prior to final approval.

In addition, the design needed to address "legacy" proposals. For example, how should we fill in a date field for proposals that were completed before that field had even been devised? (One approach: build a script that picks a date in the appropriate time period, and records the act of "guessing" in the Comments field.)

Finally, we learned that staff occasionally used the flexibility of the status code field to indicate status changes that occur without the normal accompanying actions. For example, in an exceptional case, a proposal might be moved to status "5 - UR/likely to be rec." in the absence of the usual correspondence. However, unlike the correspondence template facility, there is little "elbow room" in a calculated status code. The only way to affect the value of a calculated field is to fill in or change the fields on which it is based. Thus, to attain something like the flexibility of manual manipulation, the team needed to learn not only the semantics of the status codes, but the workings of the calculation.

Most of these discussions and deliberations occurred while exploring the facility in the sandbox. As in the case of the GFW issues, the new status code facility required the group's approval, as have the small changes that occurred since launch. I met separately or in groups with the staff members responsible for the different stages of the proposal's processing to vet the status code calculations and any new fields required.

Database integrity

The last example is in fact a much larger topic, an elephant in the middle of the room looming over the database work at GFW. We've been calling this elephant "database integrity." The examples of inconsistent proposal status mentioned above are cases in point. But the problems can get bigger and much more difficult to resolve.

A long history of working in a flat-file database with no time for quality assurance (e.g. periodic checks for missing or duplicate information) led to a database that was usable, but whose statistics generated by counting and summarizing records were not always reliable. The new architecture made consistency possible, but couldn't address extant problems like proposals with missing organizations, duplicate records with the same proposal number, artificial country names, inconsistent organization contact information, and the like.

Perhaps the most pressing example was the problem of duplicate organizations. Recall that an advantage of the new architecture was the ability to separate proposals and organizations into distinct relational tables. One organization was now linked to all its proposals rather than being replicated (more or less) for each new proposal. Let's assume for the moment that each new proposal that arrives from an organization that already has a record in the database, can be easily mapped to that record. This leaves us with the case of "legacy" data. Some proposals are assigned to different organization records, but in fact are from the same organization. How can we determine that two organization records refer to the same organization given that an organization can change name, address, contact person, even country, say, when regional wars force it to take up temporary residence across the border? And just as important from the perspective of participatory design, how important/urgent is this decision? Perhaps a few duplicates are permissible among older proposals or among those whose proposals were declined. My natural tendency as a system developer is to place a premium on data integrity. But together, we had to balance this against the human effort required in each case to distinguish duplicates. For older proposals, this means retrieving folders from boxes of archived files. How much integrity is required *and* practical in an everyday database?

For a second example, the database, which is over ten years old, includes records with missing information ranging from missing proposal dates to missing country names. The latter seem particularly ominous, given that the new achitecture includes a set of hierarchically nested geographic relational tables – or put another way, a series of one-to-many relationships:

> Program office -> GFW region -> country -> organization -> proposal

An organization with no country is left out of this linkage. Such organizations usually date from the earliest days of the database, and their proposals had often been declined.

As participatory designers we have to suspend natural urges for consistent and complete records and recognize that the human effort required to do the clean up may not be worth prioritizing relative to other pending work. That determination ought to be made in joint discussion with the workers. As yet, however, staff has not had the time to address the question of database integrity as a group. In the meantime, I've implemented a script that staff can use to merge duplicate organizations and reassign countries. A few members of staff run these scripts as they encounter problematic records.

DISCUSSION: WHO HAS TIME TO PARTICIPATE?

My overarching concern as a participatory designer at GFW is keeping *my* work connected to the ever-shifting demands and requirements of *their* work. The connections are crucial for several reasons.

- I need deeper understandings of the flow of their work in part so I can become better aware of which topics matter most and when.
- I need to hear their ideas for changes to the technological environment and to their work practices, and in turn, I need to bounce my ideas off of them. Unfortunately, responses to their concerns and ideas often occur to me days later, when the topic is no longer as fresh in their minds.
- I often wish for discussions with the group as a whole, sometimes around high-level technology planning and strategizing, sometimes in order to prioritize a collection of pending development tasks.
- I look for chances to hold sessions with individual staff members around the database partly to see what's working and not, but also to engage in opportunistic training.

These desires shouldn't come as a surprise. Increasing the communication and input I get from staff should lead to more effective and responsive technology development on my part. But technology development is not the focus of their work; their primary goal is to move money from donors and foundations into the hands of women's groups around the world. GFW's technological environment is but one instrument in that project. The staff balances time spent on technology redesign against their other pressing demands. We might as well recognize this situation as a fact of life for participatory designers. There will rarely be time for the degree of participation we feel is needed.

So, how do we manage with less? How can we make the most of the opportunities for interaction and input that arise? How do we make progress "in the meantime," when opportunities to check in with staff are hard to come by? The suggestions that follow are based on my work at GFW along with prior participatory design experience. They come with no guarantee of generality, but I have found them to be useful in GFW-like small, busy, empowered workplaces.

Be there regularly

Let's start with perhaps the most obvious approach: being there. For me, being on-site part-time, but with regular hours helped staff feel secure enough to explore and then adopt significant technological changes. Also, doing the work in place, in the sandbox for all to see and explore even when I wasn't present, made it easier for staff to foresee the results of planned changes, and begin to gain familiarity with the new database before launch.

Being there also addresses the problem of time constraints. At the site, I could take advantage of opportunities to hear about Filemaker problems as they arise. When I wasn't there, the sandbox acted as a kind of proxy for our design ideas, letting staff explore possible designs without needing to first set up a meeting with me.

Interestingly, we originally expected to create a new sandbox after launch in which we'd explore the next set of changes. Instead, we're working within the new architecture now, implementing new features and modifications within the running system without (so far) adversely affecting their work patterns. For example, we built the correspondence facility described in the previous section after launching the new database, although the idea came up in earlier design meetings. By ensuring uninterrupted access to the old facility, we slowly worked in the new scheme for those staff members and volunteers who understood how to use it and how it could help them.

Being there also enabled learning about their work practices and which issues were most important to each staff member. In keeping with recent moves to integrate ethnography and participatory design (Blomberg, 1995; Blomberg, et al., 1996; Kensing, Simonsen, & Bødker, 1998b; Mogensen & Shapiro, 1998), my regular presence at GFW with a job to do was an ideal context for participant observation.

Align design with the organization's work cycles

Part of participatory design is knowing when *not* to participate, when to stay out of the way, when to recognize that we and our technology initiatives are not at the center of *their* universe. At GFW, our collaborative activities went in waves, in phase with their deadlines and crises. Sometimes there was room for a strategy meeting or brainstorming session, sometimes only for one-on-ones with individuals, and sometimes it was best to simply stay out of the way. When staff was particularly swamped, I stayed available in case my help was needed, and otherwise took the opportunity to work on my own, say, on tasks that required learning new developer tools or scripting languages. In addition to unanticipated crises and deadlines, my participatory design work needed to align with GFW's seasonal work cycle organized around dockets, board meetings, and fiscal year transitions.

A good example is the problem of prioritizing technology development activities. As participatory design practitioners we want to keep affected parties informed of the whole set of pending tasks as well as the ones at the top of the stack, inviting reprioritization as needed and as a matter of course. In the course of this project, I've learned to see reprioritizing, though sometimes disappointing especially if my pet idea is downgraded, as a *positive* event, a sign of user/workplace engagement.

Researchers with the University of Oslo's Functional Integration through Redesign (FIRE) project propose what they call priority workshops as a way of keeping IT work in an organization aligned with user needs (Braa, et al., 1992). They describe how to conduct such meetings and who should be invited. As they admit, however, they offer no advice on *scheduling* priority workshops. As we saw in the case of standardizing correspondence templates, it is difficult at an overworked place like the GFW to mobilize regular meetings around technology. Instead, I suggest saving questions and issues until a calmer period in the cycle of activity, and in the meantime, make progress on tasks that are doable with minimum staff input and discussion.

Consider first addressing individual needs

Group meetings that forge consensus or that initiate standardization processes can be valuable as we saw in the case of revising the GFW Issues. But in the absence of such meetings, usually because of time constraints, it can be effective to orient to the voiced needs of one or a few of the workers as we saw in the case of semi-automating correspondence.

Here are some questions that can help reach a decision. Is the group ready and able to have a constructive discussion? Do they have time to meet and do the "homework" necessary to adopt a joint strategy? Is it reasonable to address an individual's problem to get the ball rolling and give future discussions something concrete to work from? As a designer, can I keep the design general and tailorable enough to adjust easily as more members of the group begin to try it out? As ever, the goal is to stay flexible and *listen*.

Identify a point person for important tasks

In this project, it was important to identify a point person for certain tasks like the correspondence facility. As I spent more time at GFW, I learned about the various attitudes and opinions in the grantmaking group, and who might be a point person with whom to try out the new facility. At the same time, I reported on my activities to the entire group and whenever possible, engaged the group in informal discussions of the tradeoffs.

"Should we fix this database problem together, or do you want me to do it behind your back?"

The cooperative prototyping session (Bødker, Grønbæk, & Kyng, 1993, pp. 170-171) is one tool in a participatory designer's bag, extremely valuable for engaging workers and showing that the wizardry lurking behind the curtain is less arcane than they think. But time constraints sometimes make a cooperative prototyping session impractical or inadvisable. I only made the offer when I thought we could actually complete the necessary design work in one session, say, in changing the behavior of a single status code calculation. Sometimes the response was "I want us to do it together, but let's put it off til later." Such cases served as exercises in learning to give up control: in order to encourage a more active form of participation, I had to put off development work until a time that suited them.

Incorporate training as an ongoing part of participatory design

Following the tenets of participatory design as listed at the start of this paper, I should be acting as catalyst for their learning, rather than simply as problem solver. Ideally, every new feature that turns out to be of value in their daily work should eventually be understood by at least one person on staff. By "understood" I mean that the staff member can motivate the feature, explain its workings, and modify it slightly to meet changes in the work practices and technological environment (MacLean, Carter, Lovstrand, & Moran, 1990).

Understanding the more complex features of the new database presumes familiarity with Filemaker's advanced features like sub-totalling layouts, calculated fields, and scripts. In fact, a few members of staff are already exploring these topics. I am concerned however, that the inner workings of some of the more complex designs (e.g. the correspondence facility) are currently inaccessible to staff. We have talked about holding an in-house workshop on advanced Filemaker topics, grounded in examples from their work. But again, this requires more of their time. In the meantime, we press on with low-road or "in-line" training, opportunistically initiated in the cracks of their work. In fact, as Bonnie Nardi and Jim Miller point out, this kind of as-needed training can be a particularly effective way to learn programming techniques (Nardi & Miller, 1991).

Live with inconsistency

As we saw in the example of database integrity, cleaning up a rich database with a long history requires time and effort. Scripts can help, but human experts are required, for example, to recognize that two organizations are the same when their names and/or contact information are different. I've been learning to live with inconsistencies, in part by ensuring that new facilities aren't overly dependent on data purity. In the longer term, we'll need a slate of tools that search for incomplete or inconsistent records, especially as a way of reminding staff of tasks that are pending.[5]

Don't always say yes

Adding features or effecting radical redesign must always be weighed against increased complexity. In the day-to-day work of participatory design, wanting to respond to each next request, I sometimes lost track of the larger picture. More up-front technology planning could have helped by setting the boundaries of redesign work. Ideally, such plans would shift and re-form organically to meet changing organizational conditions, say, through periodic strategy meetings that monitor and revise overall project directions. But again we butt up against the time constraints of an already swamped workplace. In the absence of periodic strategy meetings, it was incumbent on me to resist "feature creep."

Try chocolate

Last but not least, I've learned the value of delicious food to overworked staff members. Offering sweets at a technology priorities meeting, for example, can counteract the sometimes ponderous atmosphere of an impending technical discussion.

CONCLUSION

The staff of GFW model in their work with grantees, donors, and each other the kinds of respectful communication and service that participatory design presumes. Maintaining these respectful relationships is time and resource intensive. Although improved technology tools can ease some of the strains of day-to-day work, those time pressures also constrain their ability to engage in design deliberations. In this paper, I've recounted my experience of this dilemma and offered words of advice for others in similar situations. In the end, our best hope is to follow the fundamental principle of participatory design, to base our methods, practices, and communications on a deep respect for the skills and humanity of those with whom we collaborate.

ACKNOWLEDGMENTS

My heartfelt gratitude goes to the staff of The Global Fund for Women, who have been a joy to work with and so welcoming and patient with the participatory designer in their midst. I am continually learning from these women's adept, caring, and respectful peer relationships both within their office and across the cultural and geographical distances their work spans.

REFERENCES

Benston, M. (1990). Participatory design by non-profit groups. In *Proceedings of the Participatory Design Conference.* Palo Alto, CA: Computer Professionals for Social Responsibility, 107-113.

Blomberg, J., Suchman, L., & Trigg, R. (1996). Reflections on a Work-Oriented Design Project. *Human-Computer Interaction, 11*(3), 237-265.

Bødker, S., Grønbæk, K., & Kyng, M. (1993). Cooperative design: Techniques and experiences from the Scandinavian scene. In D. Schuler & A. Namioka (Eds.), *Participatory Design: Perspectives on Systems Design.* Hillsdale, New Jersey: Lawrence Erlbaum Associates, 157-175.

Braa, K., Bratteteig, T., & Øgrim, L. (1992). Organising the redesign process in system development. FIRE report No. 6. Department of Informatics, University of Oslo.

Clement, A. (1994). Computing at Work: Empowering Action by 'Low-Level' Users. *Communications of the ACM, 37*(1), 52-63.

Hochheiser, H. (1998). Workplace database systems: Difficulties of data collection and presentation. In B. Friedman (Ed.), *Human Values and the Design of Computer Technology (Csli Lecture Notes, No. 72).* Cambridge, England: Cambridge University Press, 239-258.

Kensing, F., Simonsen, J., & Bødker, K. (1998a). MUST: A Method for Participatory Design. *Human-Computer Interaction, 13*(2), 167-198.

Kensing, F., Simonsen, J., & Bødker, K. (1998b). Participatory design at a radio station. *Computer Supported Cooperative Work, 7*(3-4), 243-271.

McPhail, B., Costantino, T., Bruckmann, D., Barclay, R., & Clement, A. (1998). CAVEAT Exemplar: Participatory design in a non-profit volunteer organisation. *Computer Supported Cooperative Work, 7*(3-4), 223-241.

Mogensen, P. H., & Shapiro, D. (1998). When survival is an issue: PD in support of landscape architecture. *Computer Supported Cooperative Work, 7*(3-4), 187-203.

Nardi, B., & Miller, J. (1991). Twinkling lights and

[5] The potential for abuse is always present when a database is used to record work activity even at an empowered workplace like the GFW. As Harry Hochheiser explains, we should ensure that record-keeping and workflow monitoring tools are designed to meet the voiced needs of those whose work they record (Hochheiser, 1998). A certain level of trust within the organization is necessary for workers to feel comfortable that others with interests in these records will not use them to threaten or inappropriately measure worker productivity (Clement, 1994). At GFW, I feel confident that such trust is present.

nested loops: Distributed problem solving and spreadsheet development. *International Journal of Man-Machine Studies, 34*, 161-184.

Robertson, T. (1998). Shoppers and tailors: Participative practices in small Australian design companies. *Computer Supported Cooperative Work, 7*(3-4), 205-221.

Trigg, R. H., Blomberg, J., & Suchman, L. (1999). Moving document collections online: The evolution of a shared repository. In S. Bødker, M. Kyng, & K. Schmidt (Eds.), *ECSCW'99: Proceedings of the Sixth European Conference on Computer Supported Cooperative Work (12-16 September 1999, Copenhagen, Denmark)*. Dordrecht, The Netherlands: Kluwer Academic Publishers, 331-350.

Trigg, R. H., & Bødker, S. (1994). From implementation to design: Tailoring and the emergence of systematization in CSCW. In R. Furuta & C. Neuwirth (Eds.), *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW'94)*. New York: ACM Press, 45-54.