A Social Proxy for Distributed Tasks: Design and Evaluation of a Working Prototype

Thomas Erickson[1], Wei Huang[1,2], Catalina Danis[1] and Wendy A. Kellogg[1]
[1]IBM T. J. Watson Research Center
P.O. Box 704, Yorktown Heights, NY 10598, USA
[2]University of Michigan
701 Tappan St, Ann Arbor, MI 48109-1234, USA
snowfall@us.ibm.com, huangwei@umich.edu, {danis|wkellogg}@us.ibm.com

ABSTRACT
This paper describes an approach to managing tasks and processes that are distributed across a large number of people. The basic idea is to use a social visualization called a task proxy to create a shared awareness amongst the participants in a task or process. The process awareness provided by the task proxy enables its users to monitor the task state, the states of participants, and to communicate with those in particular states. We describe the concept, a first prototype, its evaluation, and discuss future directions.

Author Keywords
Social proxy; social computing; visualization; awareness; process awareness; design; workflow; CSCW; task support.

ACM Classification Keywords
H.5.3 [Information Interfaces and Presentation]: Group and Organization Interfaces – Computer supported cooperative work, evaluation/methodology, asynchronous interaction, synchronous interaction

INTRODUCTION

We are interested in supporting communication and coordination among members of distributed groups. In this paper we describe the design of a widget intended to support the coordination of relatively simple tasks that are spread across large numbers of participants. In this section we begin with a real example of the problem we intend to address, and discuss how it exemplifies a general class of problem. In the remainder of the paper we discuss related work, introduce the concept of a task proxy, describe its embodiment in a working prototype, and its evaluation via a user study. We conclude with a discussion of future work.

CHI 2004, ***April 24–29, 2004, Vienna, Austria.***

Widely Distributed Tasks

In June of 2003, a worm appeared on our organization's internal network. The IT department sent a broadcast email to the organization, detailing the measures to be taken (installing a patch, updating anti-virus definitions, and scanning all machines), and stressing the need for prompt compliance. Figure 1 provides a look at the management and monitoring of the task by one manager, as seen through her email in-box. Callouts 1 and 2 indicate the original request from IT and the manager's subsequent broadcast message to her seven direct reports, in which she requested that they complete the task and acknowledge its completion via email. Upon receiving the request, one of several things happened: some did the task and reported back promptly (call out 3); others did the task but forgot to reply; still others deferred the task; and one vacationing employee didn't get the message right away. Over all, five days passed (callout 4 shows a gap of about 48 hours and 100 messages) before the final message (callout 5) verifying task completion came in.

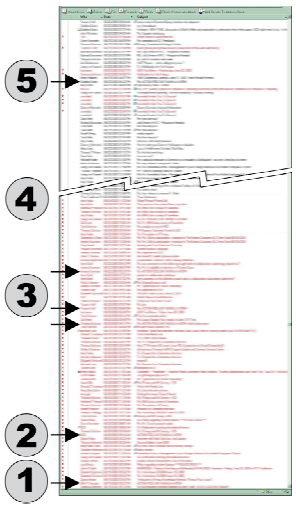


Figure 1. A 'stretched' window reveals how email relevant to one task is scattered through the queue.

Although the task itself is simple, managing it required a disproportionate amount of time and effort by the manager (and, in the general case, by the organization). There are two problems. First, responses (including questions about how to proceed in special cases) are scattered through the email queue, requiring extra effort by the manager to locate responses. Second, responses are usually embedded in the email, and not readily apprehensible without having to open each message. One employee altered the subject line to show his response, but this exacerbated the first problem

because it meant that his response did not cluster with the others when sorted by subject. Finally, because this task is just one of many with which employees and managers must cope, the multiple instances of such tasks contribute to information overload and attention management problems.

In our organization this sort of complete-and-acknowledge task occurs frequently. Other examples include certifying that employees have attended diversity training sessions; verifying that business guidelines have been read; completing inventories of equipment; submitting individual plans for self education; submitting end-of-the-year reports and acknowledging their review; and so on. Note that in tasks of this sort the *acknowledgement* of completion is as important as actually completing the task. The organization needs to be able to demonstrate that the task has been completed, and as a consequence needs to assure accountability on the part of its divisions, departments, groups and so on.

In addition to these sorts of tasks, which are characteristic of large organizations with highly developed bureaucracies, analogs of this type of task occur across or outside of organizations. One apropos example is that of managing paper reviews carried out by a distributed set of reviewers. As with the worm task, the review process is distributed across a number of people, and it is important that all (or at least most) complete the task in a timely fashion. In addition, knowing which tasks have been completed (e.g., the reviews for a particular paper) is helpful in suggesting how to proceed: if all a paper's reviews are complete, it is an opportune time to read them and try to integrate their findings; if none of a paper's reviews are complete, it may be a good time for the review manager to panic.

In summary, the sorts of tasks with which we are concerned have three characteristics: they are distributed across more than a few people, it is desirable that most or all of the participants complete their bit of the task; and it is helpful for some (or all) participants to be aware of the task's state.

RELATED WORK

An interest in supporting distributed work has a long history. While a thorough review is beyond the scope of this paper, we lay out the basic approaches to supporting distributed tasks and position our work relative to them.

To do so, we rely on the common notion that systems for supporting distributed work can be arrayed along a continuum of structure. At one end we have applications such as email and instant messaging which, whilst allowing people to coordinate their work, 'know' nothing about the work being done. As our worm example illustrates, and as is more systematically demonstrated by others (e.g., [11, 16]), the use of email to manage tasks has limitations and can result in email overload. At the other end of the continuum are workflow systems such as The Coordinator [12], and those described by [2] and [6]. While such systems have engendered considerable debate having to do with their reification of formal models of work and the resulting inflexibility (e.g., [2, 15, 17]), it is also true that workflow systems are now a common means of managing expenses, inventories and other organizational processes.

In the last decade many researchers have turned their attention to the middle of the continuum. From the structured end, researchers have explored ways of increasing the flexibility of workflow. For example, the Freeflow system [5] enables users to step out of the workflow model, even as the system notes and tracks the "constraint violation" to ensure that it is not forgotten. Similarly, investigators have explored ways to add limited amounts of structure to systems at the unstructured end: for example, the Taskmaster system [1] embeds resources for managing tasks directly in email clients.

Our approach, too, represents a foray into the middle part of the continuum. However, rather than attempting to incorporate information about the structure of the task into the system, our approach is to provide participants with a shared awareness of each others' states *vis a vis* the task or process in which they are engaged. In doing so, we draw from three other bodies of work, integrating elements of each into our approach.

First, there is a large class of systems that successfully support collaboration without using explicit task models. These include configuration management systems, bug tracking systems, help desks, and review management systems such as that used for CHI. By and large such systems do not use visualizations of the state of the task components (except insofar as one considers sortable lists to be visualizations). Nor are they generally designed explicitly to support shared awareness, although that may be one of their benefits. Grinter, for example, in describing the use of a configuration management system, notes that it provided an overview of who had checked out which code modules that enabled developers to "reorganize their work as ... their view changed" [6, p. 201].

Second, there is a large body of work by the CSCW community on awareness interfaces and on the value of shared awareness. For example, in an ethnographic study of a securities trading house, Heath et al. [8] describe ways in which dealers monitor one another's activities to coordinate their interactions. Similarly, in empirical studies Gutwin et al. [7] showed that widgets providing shared awareness of a workspace facilitated the performance of various tasks. Overall, as noted by Dourish and Bellotti, shared awareness "provides a context for individual activities and thus facilitates group progress" [4, p. 113].

Third, our approach to providing participants in a task with shared awareness relies on visualization. In this we build on work from the HCI, CSCW and Information Visualization communities, most particularly upon a body of design-oriented work in social visualization. Social visualization focuses on portraying characteristics of (and relationships among) large groups of people. Examples include Visual

Who [3], Donath's mapping of people according to mailing lists they participate in, Sack's visualization of newsgroup posters and content [13], and Smith et als' visualizations of persistent conversations [14].

THE TASK PROXY

The Original Concept

The task proxy originated as a design sketch, accompanied by scenarios of how it could support various complete-and-acknowledge tasks such as the worm and paper review management tasks. The basic idea was that a small visualization could serve as a 'proxy' for the task, providing an overview of each participant's status vis a vis the task.

The initial sketch envisioned the task proxy as a packed set of hexagons, each representing a person, with its color reflecting the task status (e.g., not started; in progress; done). Borders around groups of hexagons showed work groups and other levels of organizational structure (see figure 2 for the implemented version). Hexagons were a mostly arbitrary choice, although we did feel that the visualization's resemblance to a honeycomb was an apt way of portraying workers in an organization.

The task proxy as envisioned here enables two sorts of things. First, it permits the overall status of a task to be visualized (e.g., Figure 2 shows the task proxy filling in as the group collectively completes a task). This affords two possibilities: most obviously, it enables a manager to exercise oversight over a task for which he or she is responsible; but also, if the proxy is visible to everyone, it opens the possibility for non-centralized social phenomena such as imitation (e.g., 'Pat did it—I'd best do it too') and peer pressure (e.g., 'almost everyone is done except me!'). Second, the task proxy provides a contextualized means of communication that is tied to the task and its state: for example, the manager of a task might want to send email to only those who have not yet started it.

As we considered how the design could support various scenarios, it was apparent that privacy was an issue. We developed the idea that task proxies would have visibility policies that would govern who could see what about whom. Visibility policies could range from 'transparent' (all users can see everyone's task statuses) to 'translucent' (users can see some but not all task statuses) to 'opaque' (only a task's manger can see others' status information). Thus, if the task were to organize a potluck, with 'status' indicating the type of dish each person was bringing, a transparent policy would be appropriate; if the task were more sensitive, a more opaque policy would be better.

The process of creating, reflecting on, and discussing the task proxy concept raised a number of issues, such as: Who would use it? Is there a real need for it? Is the basic concept understandable? How would it be implemented? How would task statuses be entered and updated? Would users be able to understand the concept of visibility policies? Would the proxy be a new application, a new aspect of an email or calendaring system, or a component that lives on a web page? While the design sketch raised these and other questions, it did little to answer them. It seemed clear that it would be valuable to proceed by building a working prototype and collecting feedback from its users.

The Task Proxy Space: A Working Prototype

The Task Proxy Space supports a wide range of tasks and users. It enables users to view task proxies in which they are participants, to manipulate their state information, and to create and manage new task proxies. It also enables managers of a task to email participants in a particular state (e.g., 'not done'), and allows participants to set reminders.

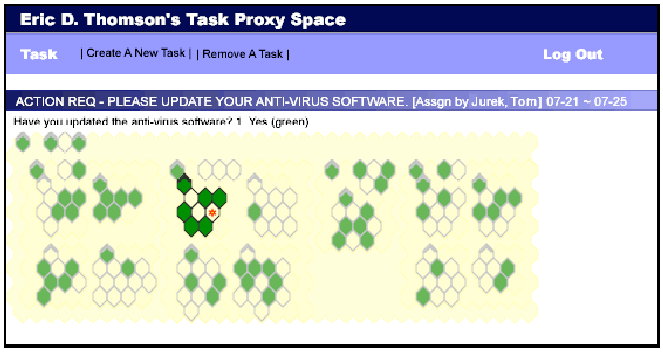
The Task Proxy Space was developed as a web-based client that used Scalable Vector Graphics (SVG™, Adobe® SVG Viewer 3.0) to produce the visualization of the task proxy and its user interface. It uses a recursive algorithm to lay out the hexagon visualization so that it can dynamically support changes in organizational structure. The task proxy data is stored in a DB2™ database; Java™ Server Pages and Java Servlets are used for database interactions.

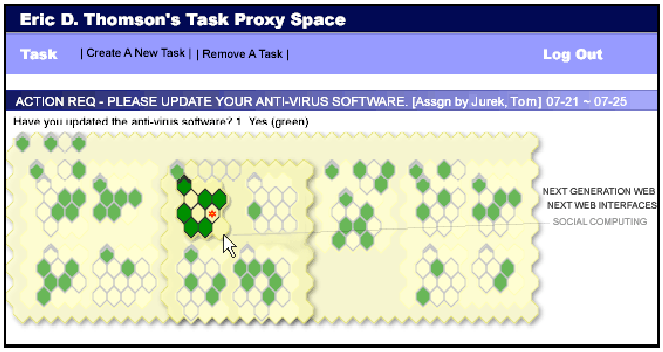
We will begin by focusing on a single task proxy visualization, describing its static appearance, and then the interactivity built into it. Then we describe the task proxy as a whole, and the creation of new task proxies.

Figure 2 shows a task proxy (extracted from the rest of its user interface) for a single work group at four points in time. Each hexagon represents an individual: the user's hexagon is marked with an asterisk, and hexagons representing managers have what users came to refer to as 'hats' (upper left corner of the group). A hexagon's color represents its user's state with respect to the task: for the proxy shown, white means that no state has been entered, yellow [light gray] means "in progress," green [dark gray] means "completed." Thus, the progression in Figure 2 depicts the gradual completion of a group's task over time.

Figure 2. A task proxy for a single group at 4 points in time.

Figure 3 shows the entire Task Proxy Space; in this case the proxy shows an organizational division that contains four departments, each of which contains work groups (the group in Figure 2 can be seen in the upper middle portion of the task proxy). Figure 3a shows the static view that permits users to see the general state of an organization wide task; it also shows the organizational hierarchy through subtle variations in background color [not visible here]. When a user mouses over an element of the task proxy, it dynamically pops up borders, shadows, and (to the right) labels that show the location of the person or group in the organizational hierarchy, as shown in Figure 3b.





Figures 3a and 3b. A task proxy (a) in its static state and (b) responding to mouse over events.

When a user mouses over his or her own hexagon, the user's name and task status is displayed in the lower margin. Depending on the visibility policy in effect, users may or may not be able to see the names and statuses of coworkers in their group, department or division. Clicking on a hexagon pops up a dialog that reveals more information about the user and task status (again, per the visibility policy in effect), and allows users to change their own task statuses. Each proxy also has a title bar and legend with task related information; clicking on a proxy's title bar pops up a window containing a full description of its task.

Finally, the Task Proxy Space menu bar allows users to create and manage task proxies. Creating a new task proxy is a matter of filling in the name, description, deadlines, etc., for the task. Users specify the number and names of states a hexagon can reflect, the set of people who will be represented in the task proxy (by selecting individuals, groups, etc., from the organizational hierarchy), and the visibility policy that governs who can see what about whom. As a consequence, a wide range of task proxies can be specified. In the prototype, all of an individual's task proxies are consolidated on a single page.

EVALUATION

Our goals were to obtain feedback on the user interface and interaction techniques, assess the extent to which users understood the basic concepts, solicit comments on the perceived value of the idea, and explore other issues such as privacy. Thus the study was designed to elicit qualitative feedback via dialog between the experimenters and users, rather than to collect quantitative performance metrics.

User Study Design and Process

We recruited 12 participants by posting a request for volunteers on our division's mailing list. All had spent at least two years in the company and had expertise in computer science. Three were (or had been) managers; participants were equally divided between genders.

After an initial explanation of the task proxy concept, participants were asked to perform three tasks, each consisting of a series of steps. The tasks used in the study were familiar to these participants. The task proxy used in the first two tasks was populated with an array of tasks statuses (as in Figure 3). The task proxy contained the appropriate portion of the organizational hierarchy (140 people), so users were able to log in as themselves and see their coworkers and managers located appropriately.

The aim of the first task (the worm scenario described earlier) was to get users to explore the proxy. We asked them to log on, find themselves in the proxy, find their group, find a different group, and update their task status. For each step, the user to attempted the action on his or her own; if a minute or so passed without success, we provided a hint; if that failed, we guided the user through the step.

The aim of the second task was to allow the user to experience the same task proxy (for the worm task) from the perspective of a different user who—because of the visibility policy in effect—would see different information. This would allow us to gauge the extent to which users understood visibility policies. Thus, we asked the participant to log on as his or her second level manager (in the case of managers, we had them log on as a non-manager). Then users proceeded through a series of steps as before (e.g., find yourself; find your group's task status; find a group that's mostly done; find task information, etc.).

In the third task we asked the participant to create a new task proxy using a scenario involving soliciting quarterly highlights for a department status report. The aim of this task was to get feedback about proxy creation in general, and the construction of visibility policies in particular. As with the other tasks, this task was divided into steps.

Following the tasks, a questionnaire with a five point Likert scale was used to collect quantitative data, and a series of open-ended questions were used to get qualitative feedback. All sessions were videotaped and had two experimenters present: one running the study, another taking notes.

Results

Performance on the Tasks

The majority of users completed all tasks without help; most *did* spend time exploring and experimenting, especially during the first task. Only two users needed explicit help, and then only for one or two steps of the task.

While working, users voiced a number of confusions and encountered various problems, which we discuss below. All participants completed the three tasks in 30 minutes or less.

Ratings in Response to the Questionnaire

Table 1 summarizes the responses to the questionnaire. For readability, the statements have been recast (i.e., originally half the statements were negative: e.g. "I had difficulty understanding…") so that agreement signifies positive ratings of the proxy. Note that since the questionnaire was administered at the experiment's end it reflects participants' retrospective views rather than their on-going experience.

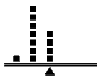
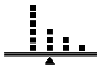
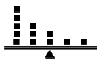
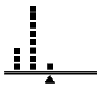
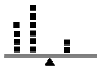
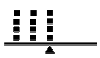
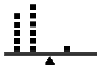
Statement	Ratings
Ease of use and understanding	+ …N… -
1. Over all, the user interface was easy to use	
2. My interaction with the task proxy was clear and understandable	
3. I had no difficulty understanding the basic concept of the task proxy	
Suspected problem areas	+ …N… -
4. I could easily find my group in the task proxy	
5. The 3-level representation of group hierarchy is easy to understand	
6. The use of color to represent task status is easy to understand	
7. I could understand why I could access the responses of some groups but not others	
Perceptions of usefulness and value	+ …N… -
8. The awareness of my own group's task status is helpful	
9. I think different access [visibility] policies are useful for different situations	
10. I think the task proxy would be a useful tool for managing tasks	

Table 1. Results from questionnaire—ratings range from strongly agree (left) to neutral to strongly disagree (right)

Overall, users' ratings of the task proxy were quite positive, as can be seen by the predominance of ratings on the left side of the scale (the triangle indicates the 'neutral' point). While positive answers must be interpreted cautiously, given that participants are often predisposed to respond in line with their expectations about the experimenters' desires, we note that their questionnaire responses were mirrored by their performance. We will consider the details of the ratings in the discussion.

Qualitative Responses

Participants were encouraged to talk aloud throughout the experiment. We paused after each task to talk, and occasionally prompted them in the midst of a task if they had been silent for a while or seemed confused. Participants also talked as they filled out the questionnaire, and, of course, during the final period with open-ended questions. Participants showed no hesitation in voicing confusions, criticisms and suggestions. We will consider the qualitative responses in the discussion session.

DISCUSSION

Usability Issues

Overall, the generally smooth performance on the tasks, the responses to the questionnaire, and the participants' comments all suggest that the basic concept of the task proxy was understandable, and that its use—in the large—was not difficult. As seen in Table 1, majorities gave positive responses to the first three statements, with the least positive response (3 negatives, 3 neutrals) coming in response to the statement 2, which asked about the interaction with the task proxy. We believe that this reflects various usability problems that users encountered while doing the tasks. These included problems in knowing what to click (e.g., to expose more details about a task), difficulties in relating the area of proxy being moused over to its label (shown off to the side), a cumbersome syntax for specifying the mapping between task states and hexagon colors, and problems with labels and instructions. These problems, now that they are known, have obvious fixes. We discuss two more significant problems in the next sections.

More generally, it is clear from participants' comments that having a recognizable organizational hierarchy embodied in the task proxy was a clear aid in their coming to understand and use it. That is, their ability to find themselves, identify co-workers, and see where various members of their management chain appeared in the visualization helped them to quickly make sense of it. Because such familiarity would be present for most of those using a task proxy for real, we do not see this as an unrealistic advantage.

Design of the Task Proxy Visualization

We anticipated problems with the visualization, and included statements 4, 5 and 6 (see Table 1) in the questionnaire to probe for suspected problems. To our surprise users' ratings were quite positive (29 positive, 3 neutral, and 4 negative ratings, overall). Nevertheless, we

did observe users engaging in quite a bit of exploration and trial and error while performing the study tasks. From their activities and comments, it was clear that the visualization design has a number of problems that stem from its complex and dynamic nature. That is, the visualization aims to show the state and names of individuals, work groups, departments (second level), and divisions (third level). Thus, as a user moves the pointer across the visualization, labels, lines and shadings that depict group and departmental boundaries highlight as appropriate, producing a complex and mutable display. While users *did* understand the basics of the visualization, its dynamic nature made a number of tasks (e.g., navigating to a particular person) more challenging than they might be.

We see two approaches to addressing these problems. First, it is possible to build support for particular tasks into the visualization. For example, when asked in one of the study tasks to count the number of people in a group who had finished it, several participants commented that the computer should just give them the numbers directly. This is obviously a good point, and would be easy to support. However, we think this approach is premature, because we don't yet really know how people will want to use the proxy. Note that the only reason users needed to count was because we asked them to, and the reason for *that* was because it was a simple way to get them to interact with the proxy. Thus, we intend to defer providing functionality tailored to particular interactions until we are able to observe a more naturalistic use of the Task Proxy Space.

A second approach to simplifying the visualization is, of course, to redesign the visualization itself. One approach is to retain the basic concept, but to make it somewhat less dynamic, by designing the visualization so that the organizational structure is obvious without having to mouse over it (i.e., eliminate the dynamic darkening of boundaries, popping up of sub-areas, and so forth). This approach is suggested by the relatively positive ratings of the proxy's understandability, as well as the fact that many users, when asked to name positive aspects of the system, said they liked the compact nature of the visualization. However, we also believe that it may be worthwhile to explore some different approaches to designing the visualization, perhaps along the line of Treemaps [10].

One other interesting issue arose from the user study. Several study participants argued for an egocentric visualization. That is, their position was that 'I am most concerned with myself and my group,' and therefore my group and I ought to be in the most prominent location in (our view) of the visualization. While this is an understandable position, it stands in tension with another position: To serve as a resource for larger groups (e.g., imagine a meeting of third level managers all looking at proxies for tasks the organization is trying to complete), it is desirable for everyone to have (and to be familiar with) the same view of the organization. That is, a task proxy serves as both an individual tool and also as a form of common ground, and these two entwined purposes pull the design of the visualization in different directions. Thus, in redesigning the visualization, we aim to explore a number of techniques for ameliorating the tension between personal and collective use of the proxy (e.g., keep the structure of the proxy constant, but to morph or otherwise highlight the user-relevant area of the task proxy).

Visibility Policies

The most troublesome area of the task proxy has to do with the notion of visibility policies. This is reflected in the mixed responses to statement 7 (Table 1) which asked users whether they understood why they could see information about some groups but not others. Given that this is a retrospective rating—that is, it was generated after they'd had time to figure things out—this clearly indicates a problem. And, indeed, the response to statement 7 is mirrored by comments and confusion during and after the second task (in which users took on a different identity and therefore saw different information): users often didn't understand why they saw different information. Some of this may be an artifact: the worm task was not generally viewed as one where privacy was an issue, and thus having a restrictive visibility policy (as was the case in task 2) didn't make sense to some users. But, on the other hand, when users were asked to construct their own proxy in task 3, they often expressed surprise at the proxy that they generated. Since users also claimed to understand the need for visibility policies (statement 9, Table 1), and were adept at giving examples of situations in which one would want different policies, our working hypothesis is that we need to focus more on how to portray the effects of visibility policies (e.g., by providing previews of task proxies as users are in the process of defining them).

To our surprise, users did not express much concern with privacy issues. They understood the need for different visibility policies for different situations, noting, for example, that the issue of whether a person had completed their anti-worm task was not as sensitive as whether their year-end evaluation had been accepted. Perhaps the lack of concern is due to the fact that the tasks themselves are familiar, and those that are sensitive have strong (and widely understood) privacy policies already associated with them, and it was assumed that these would remain in effect. While participants noted the possibility of phenomena like peer pressure, they did not see it as a problem. First, they noted that peer pressure could help them complete tasks they needed to finish. Second, if they could see that few others had finished a task, this would provide a rationale for deferring the task until there was more pressure (which, while not perhaps to Management's liking, is nevertheless a reasonable rationale for managing an overbooked day).

PERCEPTIONS OF VALUE AND USEFULNESS

One of the primary goals of the study was to get participants to reflect on the value of the task proxy, and on

the possible uses they saw for it beyond those discussed in the context of the study.

The large majority of study participants saw value in the task proxy concept. Among the 35 responses to the last three statements in Table 1, 2 were negative, 5 were neutral, and 28 agreed that being able to see the status of a task distributed across the organization was valuable. Users also commented that being aware of their peers' state was not only valuable as a form of 'peer pressure,' but that it would also enable them to see which of their colleagues had completed the task, and thus whom they might go to for help or counsel (e.g., 'ah ha, Pat's finished—I'll find out if installing the patch caused problems before I do it!'). All except one user saw the usefulness of differing visibility policies. And all except two users (one neutral, one negative) thought that the proxy would be useful for managing tasks. The sole dissenter argued that the task proxy would not actually support *managing* tasks unless it provided some means of communication or control. In fact, the proxy does allow email to be sent to everyone in a particular state (e.g., 'not done'), but the feature was turned off for the user study to prevent broadcasting multiple rounds of email to the entire division. Other participants agreed on the importance of communication, many commenting that it would be useful to be able to email or instant message with those in the task proxy.

One reaction that surprised us was that participants very much liked being able to have a coherent and compact view of their organizational hierarchy. They liked being able to easily browse their division, seeing who else, for example, was in the same group as someone they knew. (This is currently possible using the organization's online directory, however it is a slow and cumbersome process that involves traversing the hierarchy branch by branch.)

Other indications of value come from the suggestions for other uses of the task proxy. These included:

- **Non-organizational task management**: A number of users commented that the task proxy was useful for more than official, organizational tasks. That is, many less formal collaborations could benefit from the shared awareness—e.g., supporting the activity of a reading group by showing who had yet to do their readings).
- **Organizational Task Tracking**. Some users liked the idea of having a single place to track the complete-and-acknowledge tasks in which they were involved, versus having them scattered across email queues and web sites.
- **Task component management**. Rather than having a cluster of hexagons represent people involved in a distributed task, several users commented that hexagons could represent different elements of a task. Thus a task proxy could represent everything from a personal to do list (where each hexagon represents an item, and clusters represent types of items), to a way of signing up for a 'pot luck' dinner (where each hexagon represents a type of dish), to a collaborative task in which the completion of one component depends on another (with the visual representation expressing the dependency relationship.
- **Deadline-oriented task management**. Some users suggested integrating task deadlines into the display so that (in one person's vision) one could watch task proxies gradually drifting towards a tangible deadline.

FUTURE WORK

While our results provide reason to be encouraged about the understandability and usefulness of the task proxy concept, there is more to do. Our user study suggests that more work should be done on refining the visualization and on making visibility policies easier to construct and understand. We also need to explore ways of managing the trade off between tailoring a task proxy to the needs of its particular user, and making sure that it is useful as a shared resource for the group. Advances on these fronts seem likely to provide value for developing other sorts of task proxies (e.g., such as those laid out in the last section).

In addition, the work we've described in this paper has a number of limitations that we hope to address in the future. First, because this was the first working prototype, we opted not to deploy it to our division of a 140+ people; instead, we choose to do lab-based user testing. As a consequence, there were a number of issues that we did not get direct feedback on. Chief among these was the question of whether the task proxy's visualization would be successful in supporting group-based phenomena such as imitation and peer pressure. Although our users assumed such phenomena would occur when talking during the study, direct evidence that bears on this will only come through an actual deployment of the system.

Second, the working prototype, as it stands, is not well integrated with users' digital worlds. While this seems a natural consequence of iterative development—the basic functionality of the prototype is implemented and tested, and only after that seems sound does one pursue integration with other systems—nevertheless it results in a number of problems that will need to be addressed as we move forward. Perhaps the most obvious is that task proxies rely on their users to manually update their states. While this is acceptable for the complete-and-acknowledge tasks we initially focused on—these are organizationally mandated tasks, and neither their performance nor their separate acknowledgement is optional—it is well known that systems that rely on users to update state are prone to failure. Fortunately, developing proxies that automatically update their states in response to events sent by other programs seems possible.

Another integration problem has to do with how task proxies are organized and viewed by their users. In the prototype, all of a user's task proxies were displayed in a single, web-based task proxy space. While this is an acceptable solution for the case of the complete-and-acknowledge tasks—indeed, providing a single place for tracking the status of all institutionally mandated tasks

would be a great advance over current practice—it seems clear that were they used for a wider range of tasks, it would be important for users to have flexibility over where and when task proxies were visible. Given that people vary widely in how the organize their work, it would be valuable if task proxies could be displayed in different places in users' systems—for example, associated with a calendar, and/or an email client, and/or on the desktop, and/or a PDA.

Finally, the potential value of task proxies needs further exploration. Task proxies represent a middle ground in various genres of task management, from locally initiated 'grassroots' tasks, to formal, enterprise-wide workflows. One interesting challenge would be to extend task proxies to more complex and abstract tasks, such as executing corporate-wide business strategies. How might a proxy representation help organizations self-organize to meet strategic objectives? How might progress towards such objectives be tracked through the constant reality of organizational change? And to what extent might task proxies mitigate the needs for top-down organization and control? Supporting such tasks remains for future work.

ACKNOWLEDGMENTS

Thanks to Christine Halverson for assistance in designing and setting up the user study, to Jason Ellis, Jeremy Sussman, and Tracee Wolf for advice on the design and implementation, to Paul Matchen for assistance with SVG, and to the participants in our user study. Thanks as well to Judy Olson and several anonymous reviewers for helpful comments on the paper.

REFERENCES

1. Bellotti, B., Ducheneaut, N. Howard, M. and Smith, I. Taking Email to Task: The Design and Evaluation of a Task Management Centered Email Tool. *Proc. Conference on Human Factors in Computing Systems* (CHI '03), ACM Press (2003), 345-352.
2. Bowers, J., Button, G. and Sharrock, W. Workflow from Within and Without. *Proc. European Conference on Computer-Supported Cooperative Work* (ECSCW '95), Kluwer (1995), 51-66.
3. Donath, J.S. Visual Who: Animating the Affinities and Activities of an Electronic Community. ACM Multimedia 1995, pp. 99-107.
4. Dourish, P. and Bellotti, V. Awarenss and Coordination in Shared Workspaces. *Proc. Conference on Computer Supported Cooperative Work* (CSCW '92), ACM Press, 107-114.
5. Dourish, P., Holmes, J., MacLean, A., Marqvardsen, P.A. and Zbyslaw, A. Freeflow: Mediating between representation and action in workflow systems. *Proc of the Conference on Computer-Supported Cooperative Work* (CSCW '96), ACM Press (1996), 190-198.
6. Grinter, R. E. (2000) Workflow Systems: Occasions for Success and Failure. *Proc. Computer Supported Cooperative Work* (CSCW '00), ACM Press (2000), 189-214.
7. Gutwin, C., Roseman, M. and Greenberg, S. (1996). A Usability Study of Awareness Widgets in a Shared Workspace Groupware System. Proc *Conference on Computer Supported Cooperative Work* (CSCW '96), ACM Press (1996) 258-267.
8. Heath, C., Jirotka, M., Luff, P., and Hindmarsh, J. Unpacking Collaboration: the Interactional Organisation of Trading in a City Dealing Room. *Computer Supported Cooperative Work*, Vol. 3, 1995, 147-165.
9. Hughes, J.A., Randall, D. and Shapiro, D. Faltering from ethnography to design. *Proc. Conference on Computer Supported Cooperative Work* (CSCW '92), ACM Press (1992), 115-122.
10. Johnson, B., and Shneiderman, B. Treemaps: A space-filling approach to the visualization of hierarchical information structures. *Proc. of the 2nd International IEEE Visualization Conference,* IEEE (1991), 284-291.
11. MacKay, W. E. More than just a communication system: Diversity in the use of electronic mail. *Proc. Conference on Computer Supported Cooperative Work* (CSCW '88), ACM Press (1988), 344-353.
12. Medina-Mora, R., Winograd, T., Flores, R., Flores, F. The action workflow approach to workflow management technology. *Proc. Conference on Computer Supported Cooperative Work* (CSCW '92), ACM Press (1992), 281-288.
13. Sack. W. Discourse Diagrams: Interface Design for Very Large Scale Conversations" in the *Proc. Hawaii International Conference on System Sciences* (HICSS '00), IEEE Computer Society, (2000).
14. Smith, M.A. and Fiore, A. T. Visualization Components for Persistent Conversations. *Proc. Conference on Human Factors in Computing Systems* (CHI 2001), ACM Press, 136-143.
15. Suchman, L. Do categories have politics? The Language/Action Perspective Reconsidered. *Computer Supported Cooperative Work*, 1994, 2:3 177-190.
16. Whittaker, S. and Sidner, C. Email overload: Exploring personal information management of email. *Proc. Conference on Human Factors in Computing Systems* (CHI 96), ACM Press (1996), 276-283.
17. Winograd, T. Cateogries, disciplines, and social coordination. *Computer Supported Cooperative Work*, 1994, 2:3, 191-197.