Seeing Further: Extending Visualization as a Basis for Usable Security

Jennifer Rode, Carolina Johansson[†], Paul DiGioia, Roberto Silva Filho, Kari Nies, David H. Nguyen, Jie Ren, Paul Dourish, and David Redmiles

Institute for Software Research
University of California, Irvine
Irvine, CA 92697-3425

[†]Department of Information Technology
Uppsala University
Box 337, 751 05 Uppsala, Sweden

{jen, cjohanss, pdigioia, rsilvafi, kari, dhn, jie, jpd, redmiles}@ics.uci.edu

ABSTRACT
The focus of our approach to the usability considerations of privacy and security has been on providing people with information they can use to understand the implications of their interactions with a system, as well as, to assess whether or not a system is secure enough for their immediate needs. To this end, we have been exploring two design principles for secure interaction: visualizing system activity and integrating configuration and action. Here we discuss the results of a user study designed as a broad formative examination of the successes and failures of an initial prototype based around these principles. Our response to the results of this study has been twofold. First, we have fixed a number of implementation and usability problems. Second, we have extended our visualizations to incorporate new considerations regarding the temporal and structural organization of interactions.

Categories and Subject Descriptors
H.5.1 Information interfaces and presentation (e.g., HCI): General – Evaluation/methodology; K.4.4 General: Computers and Society – Security

General Terms
Design, Experimentation, Security, Human Factors

Keywords
Effective security, theoretical security, usable security, user study, dynamic visualizations, configuration in action, peer-to-peer file sharing, history, user and media characterization

1. INTRODUCTION
Although interest has been growing in the usability of privacy and security, there is still considerable debate over what topics are actually of concern here. One approach (what we call the "strict usability" approach) applies traditional usability measures to individual security components that people might employ in the course of regular computer usage (e.g. passwords and other mechanisms for authentication, encryption technologies, virtual private networks, communication tools, etc.) A second approach (what we call the "everyday use" approach) argues that privacy and security cannot be held to absolute measures, but rather need to be negotiated in everyday use just as social scientists have argued for interpersonal privacy [3,4]. Visualization technologies have been a particularly appealing mechanism in this approach.

Advocates of this everyday use approach, including ourselves, have argued that the critical concern for "usable security" is not that applications or software components demonstrate measurable effectiveness upon some abstract scale, but rather that people must, in the course of their activity, be able to make informed decisions about their actions. Ironically, this involves an inversion of traditional approaches to usability. Where "usability" has often been associated with a distancing of users from the details of system implementation, visualization approaches argue that, in fact, aspects of a system's behavior need to become visible or manifest to people as part-and-parcel of their interaction with technology.

It is often suggested that this approach is problematic because exposing people to the details of system operation might be confusing and overwhelming. Two analogies may help to express our position. One is the analogy of driving a car. Most drivers, consciously or unconsciously, monitor and respond to aspects of the car's internal behavior that become apparent to them in the course of driving – such as the sound of the engine, the feel of the steering and the clutch, etc. This does not require that they have a detailed understanding of the car's mechanical and control systems, but merely that their activity is coupled to the car's actions in ways that allow for fine-grained control. Our second analogy is to other aspects of system behavior. Not all components of the user interface manifest themselves graphically. One key source of information about the behavior of a system is its temporal response – what things are quick, what things are slow, how responsive the system might be. These are cues around which user activity is organized. Hence, when we suggest that a goal for usable security is to make aspects of system behavior visible so that people can make informed decisions, we neither suggest a dependency on complex models of system structure, nor extensive graphical displays. Rather, we want to make system behavior apparent in ways similar to those that support the detailed temporal organization of activity and the reflexive self-monitoring of a driver.

We have been exploring this approach to usable security in the Swirl project. Early work from this project was published in the

SOUPS conference last year [6,7]. In this paper, we wish to explore a number of further issues addressing unresolved questions from our earlier work, including questions that arose in discussions following our paper presentation last year, particularly more extensive evaluation. In addition, we want to show how we have been extending the initial techniques in order to incorporate new considerations, most particularly those of the temporal and structural organization of interactions.

2. The Impromptu Testbed

In a paper for this conference last year [6], we discussed three design principles that we are exploring in order to further understand our theoretical approach: visualization mechanisms, integration of configuration and action, and the use of event-based architectures. The Impromptu prototype, an application for ad-hoc peer-to-peer file sharing, was developed as a testbed to explore both the concept of integrating action and configuration and the concept of dynamic visualization of activity. That is, Impromptu was not designed to be the best file sharing interface, but simply an application with which we could explore our design principles in a scenario that would be comprehensible to test subjects. A journal publication delves more deeply into the theoretical design and related literature [5].

Visualizing system activity gives users a means of understanding and assessing the consequences of their action. By providing dynamic feedback on relevant but hidden aspects of system activity, our goal is to provide people with a means to understand the relationship between their actions and the technology configuration through which they are performed.

Conventional interfaces separate configuration and action in both space and time. System activity is usually separated from configuration, through the use of a separate control panel in which preferences are set. This presents a dual problem: not only does it separate two coextensive forms of activity (the act of "sharing being distributed across the preference window and the system window), but it also separates the expression of preferences for the occasion or situation in which those preferences are to be invoked. Our design approach seeks to make configuration and action part of the same interactional space.

2.1 Design

Figure 1 depicts the Impromptu client interface. The primary interface feature is the circular "pie" corresponding to the shared workspace as a whole in which each "slice" corresponds to a single user's area of the shared workspace. These areas expand and contract as users arrive and leave. Files, represented by labeled dots, are placed in and around the circular region. Each area is tagged, on the pie's perimeter, with a unique color assigned for each user. This color is also associated with a user's files, and with indicators of that user's activity.

The pie in turn is separated into multiple concentric regions; the basic metaphor is that the closer the files are to the center, the "more shared" they are. Various degrees of sharing might be implemented. The particular mappings we have been using are that files outside the circle are not shared at all, but available to the local user only; files in the outer region are visible but not readable or writable to others; files in the next region are readable but not writable; in the next, readable and writable; and in the center, readable, writable, and available persistently. Persistent access means that the file remains accessible even after the owner leaves the session; by default, files are non-persistent, meaning that when the user leaves the session, their files will disappear from others' interfaces.

The dynamics of the interface reflects its concern with the visualization of internal actions. Individual activities are reflected quickly to the group as a whole, for two reasons – first, this ensures that everyone can see potentially consequential actions, and second, it provides individuals with direct visual feedback on the ways in which their own actions are seen by others. This is an important consideration in developing an understanding of the consequences of action. Furthermore, the dots that represent files also represent activities over those files. For example, remote file accesses to local files cause the icons for the files to blink in colors that indicate the identity of the user accessing them. This dynamic visual display draws attention to current activity and allows for a quick overview of access patterns.

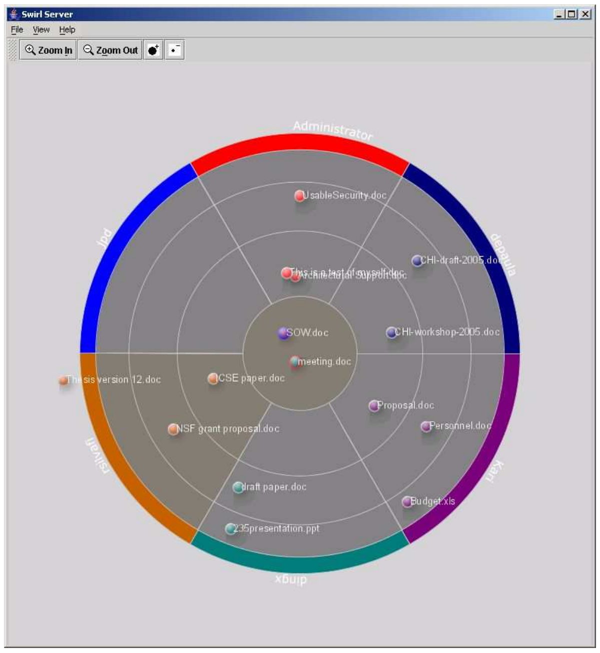

Figure 1: Impromptu Client Interface

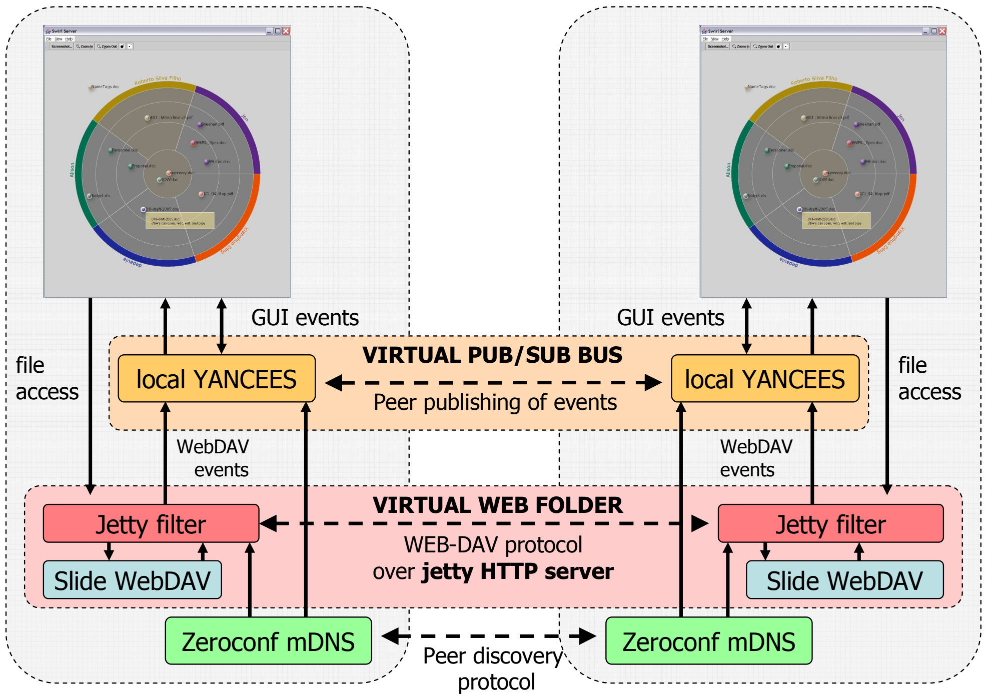

Figure 2: Impromptu Architecture

2.2 Implementation

The goals of the Impromptu testbed imply four significant constraints on software design and implementation. First, setting up a collaborative file space should require essentially zero configuration; the overhead must be negligible, or close to it, in order for the application to be effective. Second, since sharing is ad hoc, it should require no prior registration of relevant parties. Third, ideally, the system should be operable with no fixed infrastructure; it should not require, for example, connection to the public Internet. The fourth is that it should operate on a wide number of platforms. Counter-intuitively, strict security is *not* a requirement, for two reasons: we see security as a relative matter for user determination, and our goal is to make both secure and insecure states visibly manifest.

The implementation of Impromptu was improved since our previous SOUPS paper based on issues arising in the evaluation described in the next section.

Primarily, changes were made in order to improve the performance and the integration of the application with the operating system. An HTTP filter was used instead of a Servlet, in order to implement our virtual global repository, increasing performance. The access to the local WebDAV repository is now made through the operating system (currently Windows, Mac OS X), allowing files of all types to be shared and manipulated by the application.

The current Impromptu architecture is illustrated in Figure 2. Each client's files are stored in a WebDAV repository. WebDAV [9] is an IETF standard that extends the HTTP protocol for distributed authoring and versioning. WebDAV provides us with a standard interface to access file and control access permissions. Each client runs a local instance of Jetty [1] a Java HTTP server containing a Slide [2] WebDAV servlet. A Jetty filter stitches these separate servers/repositories together and creates, on each client, a unified virtual shared space. For example, when reading a file in this virtual folder, the filter will redirect any request for a remote file to the appropriate peer where it is being shared; on the same token, a request for a listing of all files will return an aggregated list of the contents of all Impromptu peers' repositories. The choice for WebDAV was motivated by its easy integration with current operating systems, being broadly accessible across platforms both through Web interfaces and also through native file system interfaces on a range of systems including Windows, MacOS X, and Linux.

In our unified repository model, there is no central server; the system operates entirely as a peer-to-peer architecture in which each "client" is, essentially, also a server and in which no server has a uniquely distinguished role. Shared files, then, are distributed across the set of clients that make up a session, and so when a user leaves, their files disappear from the workspace. When users leave the system, all their persistently shared files (those in the center of the "pie") are automatically moved to another machine. In this way, a session persists through multiple arrivals and departures until, finally, there is no Impromptu client running.

One particular challenge in a peer-to-peer workspace implementation is the identification and management of peers that are constantly arriving and departing from the network. We accomplish this using an implementation of the IETF Zeroconf protocols [14]. Zeroconf is a set of protocols that implement peer discovery, address allocation, name resolution, and related services over the TCP/IP protocols. This allows Impromptu peers to find each other automatically with no previous configuration or user intervention. Whenever someone runs Impromptu, it automatically finds and joins other Impromptu peers on the same network.

Impromptu is implemented over an event-based architecture, which allows the GUI component to monitor and register access events from both local and remote repositories, as well as to present a unified (WYSIWIS) view of the application session. In other words, in Impromptu, events are used to both visualize dynamic activity and to ensure view consistency. This is accomplished by a virtual event bus that connects local and remote repositories with local and remote interface components.

The event-bus was implemented using YANCEES (Yet Another Configurable and Extensible Event Service) [13]. YANCEES provides a higher level of extensibility and configurability through the use of plug-ins and extensible languages, allowing the infrastructure to be adjusted and tailored to the need of the application. With YANCEES, developers can define their own plug-ins for each aspect of event publishing, routing and dissemination. YANCEES allowed us, for example, to customize the way the event routers are federated. A protocol plug-in implementing IETF Zeroconf was used to integrate different YANCEES instances in each Impromptu peer, providing an event bus that adapts to the current Impromptu configuration. Additionally, each local YANCEES router was also customized with a fast switch event routing plug-in that allowed it to scale to the needs of our interface. Finally, YANCEES supports publish and subscription filters that allowed us to implement security and visualization policies. For example, filters were developed that prevent local events such as the reading or writing to private files to be propagated to other peers.

The architecture is designed for ease of use, especially to minimize configuration, and to allow for flexibility in working styles and in patterns of collaborative engagement. Given that the application scenario is support for face-to-face workgroup meetings, scalability was explicitly not a concern, nor was remote working. This same motivating scenario was the basis of our initial user study, described below.

3. User Study

As described above, Impromptu is intended to serve as a testbed for a set of design experiments in the use of visualization in support of usable security. Having constructed the basic implementation described above, we initiated a trial to study its use. Given the relatively early stage and broad scope of this work, the goal of our study was not to test specific hypotheses, nor to generate quantitative data about the use of particular features; it was *not* a usability trial. Rather, our goal was a broad formative examination of the successes and failures of the initial design, in support of further iterations.

In order to achieve this, we designed an open-ended, semi-naturalistic study in which a high-level task would provide the context for exploration and use of Impromptu, so that we could observe the use made of various features.

3.1 Experimental Design

3.1.1 Subjects

We recruited 24 graduate students all of whom were pursuing degrees with an Informatics concentration. These participants, clearly, represented the upper end of computer skills for our target population; however, they did not have prior familiarity with the project nor the goals of the user interface they were testing.

3.1.2 Method

The study itself was comprised of eight small group sessions. As all participants were students, group members had a mixture of strong and weak ties. Each session contained participants from a variety of research groups, so the session had neither an inherently competitive or collaborative bias from the start.

Each session had three participants using the Impromptu application. All sessions were run by a single facilitator, and each participant had a dedicated observer taking notes on their interactions with the system. Sessions were audio taped.

Following each session, user participants were debriefed individually by one of the note takers. In the course of the debrief we encouraged each of our 24 participants to provide three negative and positive critiques of the user interface.

3.1.3 Task

The overall task was for participants to collaborate on a research budget as part of a grant application. The combined maximum for the budget was $15,000, to cover travel and equipment expenses. Participants received a list of estimates of costs for common pieces of equipment and typical conference travel. They were also allowed to use the Internet to look up additional information. To help ensure participants take the task seriously they were asked to imagine that this opportunity was their one chance to get their advisor to pay for all of the equipment and travel, the everyday financial realities of their research.

Specifically, each participant was asked to compile first an individual budget, and then create a justification for each expense. As part of doing this participants were instructed to import these individual files into the Impromptu workspace. It was left up to the individual participant to decide if the file was to be totally invisible, visible but not accessible, readable, writable, or persistent. Next, they were asked to compile a shared budget that took into account individual requests. The nature of the task meant participants were encouraged by the facilitator to make their individual budgets available to other participants. In practice, this meant that participants who had not already done so felt social pressure to transitioning their files from the invisible or visible but not accessible state to a readable, writable, or persistent state.

The nature of the task meant that participants had considerable leeway as to when and under what circumstances they choose to share their files and to what degree. Further, given that participants were competing for resources they could create strategies to help maximize the amount of money that would be allocated to them. Strategies included free sharing of information from the start (e.g. session 4), hiding personal budget until the last possible minute (e.g. participant A in session 6), sharing despite other's strategies (8b), or maliciously editing other budget justifications to help ensure they received more money (7c). This meant that privacy in the form of setting access control of one's own files were instrumental to the task.

3.2 Findings

As we previously stated this study was not intended to generate quantitative data about the use of particular features but rather to assess the effectiveness to the approach, specifically the integration of configuration and action and the use of dynamic visualization of system activity. In addition we were interested in identifying areas for future attention and research.

3.2.1 User Interface & Implementation

Although the study was not intended to generate data about the use of particular features, the open ended nature of the task often resulted in speculative feedback on the user interface and performance rather than feedback on the tasks themselves.

Our goal was to understand how people would make use of Impromptu. Accordingly, we did not specifically prime participants to focus on security; rather, we wanted to see how these issues would arise in naturalistic interaction. We were gratified, then, that participants viewed Impromptu primarily as an integrated collaboration tool rather than a file sharing application. In fact, the concreteness of the user interface design seemed to create significant expectations for sharing within the interface. For example, nine users complained that documents did not update "live" (i.e. that Microsoft Word, when run from Impromptu, did not become a multi-user tool). While framed as negative comments, then, we actually take these as positive affirmations that, first, the focus on concreteness in the interface generated a strong sense of shared activity, and, second, that sharing and interaction, rather than security, were the primary focus of people's attention in the trial.

Another significant complaint was on the performance of the system. We have since devoted considerable attention to addressing these performance issues, streamlining the implementation in order to eliminate a number of problems that had resulted in significant performance degradation.

3.2.2 Configuration in action

As we had hoped, the structure of the task encouraged different styles of collaboration to emerge, and in turn required that people think about "security" and degrees of sharing differently as the task progressed. So, during the first more mercenary phase of the task one participant commented (6a) *"I can't grab anybody else's files. That's probably a good thing."* Later on, a more collaborative spirit emerged where participants negotiated the setting of file permissions dependent on the task. This negotiation also allowed for the creation of collective norms and strategies towards sharing, as in the following exchange:

Participant 7a: *"Do I have to share?"*

Participant 7c: *"Come on. Put it in the second ring"*

Facilitator: *"Why did you say the second ring?"*

Participant 7c: *"Well, you know. It's the norm, and you don't want to share more than necessary, right."*

This suggests Impromptu supported context sensitive negotiation of sharing, and further encouraged participants to develop explicit strategies as to how to best share files to achieve their task related goals. The assessment and recognition of these norms relies on the fact that actions and configurations are mutually visible to all.

One participant went as far as to express concern configuration in action was too easy: *"For instance it's easy to just drag it from outer spiral to the inner spiral to make it more public because this is a file I really don't want to be seen at all by the people. If it's too easy for me to move to the middle then maybe somebody can see it while I actually can drag it across"* (1c). However, despite this concern addressed in the debrief none of our participants commented, nor did we observe, anyone mistakenly giving participants access to their file which they did not intend.

Further, several participants commented on the benefits of a visual interface as opposed to a more traditional textual view of security settings. These included having *"visual control"* which they felts was *"more immediate than setting permissions"* (1a). Further dragging and dropping meant not having to remember commands, prompting participant 6b to comment, *"I realize it is a much easier than I used to think."* Participant 7b sums up these benefits as *"There is no new conceptions [sic] regarding of security access level, but it gives me visual areas, opens that concept to many people. In Unix, there is access control, but it is not so obvious. I think the new thing is the interface, they way we can change the level, and the colors and visual cues such as blinking make a lot of people understand the accessibility."*

Table 1. List of 13 positive comments on Impromptu's ability to support configuration in action:

4	Easy to share files
4	Easy to set permissions
2	Easy to modify files
1	Doesn't require technical knowledge of permissions
1	Private level is intuitive
1	One can show or hide easily

Impromptu provided individual participants with an ability to configure while completing actions. Next we will discuss Impromptu's partial success in allowing visualization of system activity.

3.2.3 Dynamic visualization of system activity

Impromptu, also, allowed participants to make sharing decisions in context of their situation. However, the interface design decision to make the Impromptu tool a separate window from the application challenged this goal to some extent, as 7b comments, *"it still decouples from the applications we use."* It was possible to obscure the Impromptu UI by maximizing a word document, as participant 4c comments, *"We focus on files and projector, but not [Impromptu]. The monitor is small, and it is easy to cover [Impromptu]."* This presents a significant usability problem which could be overcome through an additional small persistent screen containing the Impromptu user interface, or a persistent panel in the user interface.

Impromptu gave participants a sense of others participation. Our data indicates participants noticed when others added files to the collaboration (4b, 8c) through the appearance of new dots. Impromptu allowed participants to ascertain ownership of files, as indicated by participant 3a's comment that you know "*whose files are whose…you know what's important to share." Further, Impromptu "emphasizes what to explore, what's important."* Several of our participants (5c, 7c, 8a & 8c) relied on the mouse-overs indicating sharing level to help them decide at which level to share their file.

The Impromptu application supported participants' ability to see new files added, the changes in permissions, and to check to see how files had been updated. Impromptu provided participants with a sense of how other participants had interacted with the files:

- 8c: *"Yes, because it's saying read-only but, you know, initially it was – change the whole document on this – in this area."*
- 4a: *"It made it clear if someone can see or view files, but just a little bit. The visualization was a little helpful."*

However, this history of interaction proved inadequate as discussion of the rings around the file indicated. The rings around the file indicated who most recently interacted with the file in this case interacting could mean reading, editing or copying a file. Participants generally understood this meant others had interacted with their file, except for participant 1c who asked, *"but maybe if there could be a mechanism to see whose reading your file right now. Does that exist?"* during the debriefing. While a few participants mistook the rings function generally, most broadly understood the concept though there was confusion about the nuances of their function:

- 4a: *"Someone asks who opened my file? It looks like someone edited my file!"*
- 7c; *"OOOH, the ring is who has it open, or who has ownership of it. Cool!"*
- 8a: *"Oh. Oh, cool. The little – so the little outer ring on the dots is. like, who's got it open, or who's got ownership of it right now. Yes, yes. Oh, that's cool. {Sound of computer chime}. So, I guess I'm the only person that actually went over (inaudible) so I can just trim some of my stuff off (inaudible), if that's cool"*

Participants were confused as to whether the ring indicated the current state of the file (ownership) or whether it represented a past edit of the file. This suggested that a single ring serving as indicator of the current state of the file as well as an indication of previous interactions with the file was inadequate.

Many of our participants used the application to examine new files and recent changes, which proves promising for security. For instance, in the case of our "malicious" participant, the change of the ring color did successfully, although not immediately, indicate a change had occurred. As a result, the file's owner opened the modified file and discovered the "malicious" alteration of the document. The "malicious" participant commented on this in his debrief, *"I was very careful. I didn't give other participants 'write' access to my files. Others were not so careful. They left some files writable. I changed one participant's justification to make him greedy. That is one of the scary things, which partly makes this an interesting scenario"* (6c). A cordial confrontation occurred which resulted in the "malicious" participant to promise to undo his changes.

The ability to monitor participants' changes and respond to all threat situations which occurred, suggests that Impromptu was successful as a means of visualizing system activity. However, there was a sense that visualizing only the immediate state of the system was inadequate to address all of the participants' needs. This was an important consideration for our subsequent work (below).

A concern did arise as to who could log into the system. Participant 5a commented to this effect that they were *"Unsure who can access - there is no control access"* (5a). Further, participants mentioned wanting ability to set participant by participant permissions. As participant 3b commented, *"It would be good if you could grant very limited access to just one person—a finer granularity that is not just for all people but for a specific user."* While the ability to set participant by participant permissions was outside of the tasks on which we choose to focus, this information does make it clear that there is a need to distinguish familiar and unfamiliar participants.

Table 2. List of 20 positive comments volunteered during debrief about the ability to visualize system activity:

5	The rings and blink around file icons indicate what is open
5	Permits you to see what others are doing, "awareness"
4	Clear indication of which files belong to who
2	Concentric spheres representing levels of privacy
1	Clear who is logging in
1	Clear indication of who is looking at what file
1	Clear indication of who is accessing your own files
1	Good visualization of different levels of access

3.3 Discussion of study results

As indicated, our empirical investigation was not intended to provide a quantitative measure of effectiveness. Rather, we had two goals – first, to assess the effectiveness of the approach in broad terms, and second, to understand areas for future attention.

Broadly, the results support our initial design principles. It was clear that people were able to accomplish the task, were able to interpret activities that they saw manifest within the interface, and were able to configure the interface appropriately to the work being conducted. The integration of action and configuration – as reflected particularly in the spatial arrangement of the interface and its use of direct manipulation techniques – presented few problems and was, largely, picked up easily and naturally. As a number of subjects commented in the post-experimental debrief, the progressive approach to file permissions was natural and easy to pick up even without detailed understandings of file system security. Further, our focus on concreteness and mutual visibility supported the emergence of group norms, as attested to by comments in the debriefing and exchanges during the tasks. Since everyone's actions were "publicly" visible, and since the common views and common orientation of interfaces made for a strong sense of shared presence, informal conventions about configuration emerged; in the experimental task, groups' final configurations displayed a remarkable uniformity between participants. Our primary concerns with respect to both real-time visualization and integration of configuration and action, then, seemed to be justified.

On the negative side, system performance was a major consideration, and a major focus of subsequent attention. This was, in fact, the single largest negative issue reported, but it is not relevant to this paper. A number of specific UI issues arose, as indicated above. However, beyond these, our study provided us with three areas for further research and design attention.

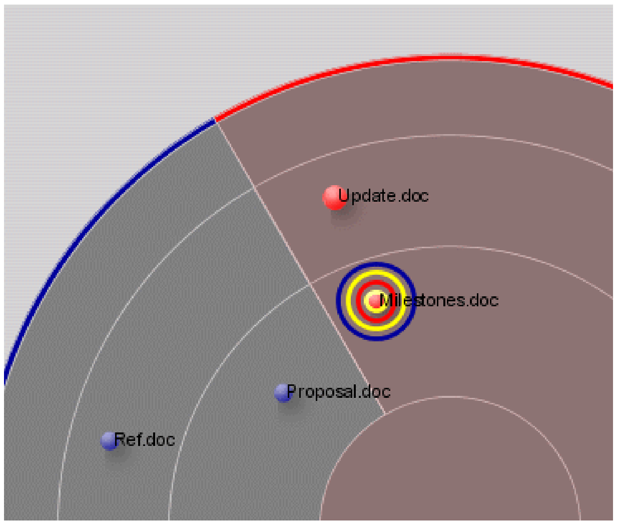

Figure 3: History Rings

First, it drew attention to the problems of screen “real estate” and in particular that the Impromptu user interface could be obscured by maximizing a window. This is the subject of our future work, as we will discuss later.

Second, an aspect of behavior that we particularly noticed during task performance was the understanding of previous activities. While the facilities provided in Impromptu support real-time visualization of activity, events are not available for later re-examination. We already saw cases of people using, for example, ring color to indicate not just current activities but also action in the recent past, but this history is very limited. In addition, as people work on tasks supported by Impromptu, they work in other applications in order to edit files, etc., and so their attention is not always directed towards the Impromptu window. As we had noticed in previous experiments, this is particularly problematic when screen real estate is limited. Recovering recent context on returning attention to the Impromptu window is a useful facility. However, it was important for us to do this in ways that do not interfere with the concreteness and directness that characterizes the interface.

Third, given that we had chosen to develop an open system where anyone could join the collaboration, data from our participants illustrated a need to provide more information on new participants to allow familiar and unfamiliar participants to be easily distinguished. Further, this would allow participants to asses the security risks posed by new participants so they can configure their responses.

Accordingly, our design efforts after this user trial focused on addressing the second and third issues—the history of the visualization and additional information on new participants. Our efforts were directed towards attempts to go beyond instantaneous views of activity, and to incorporate a wider range of considerations into the same visual framework.

4. EXTENDING THE DESIGN

In our work since the user trial reported above, we have sought to extend the visualizations in the Impromptu framework, building on what worked well and extending into areas that seemed to require more coverage. We have been particularly concerned with history and temporal consistency. We discuss the different visual extensions individually here.

4.1 Rings and Ripples

The ability to be able to see more than simply immediate action was a repeated observation in the user study. In order to display more history than just the most recent activity, while maintaining the physical metaphor that sustains the rest of the Impromptu design, we extended the rings into “ripples.” The initial “rings” were borders of the document icons that would flash to indicate activity over the document by another user. After having flashed for a shorter period the ring would stay on permanently around the icon until a new activity occurred. The color of the ring indicated the identity of the user generating the activity indication, although simply the fact of activity rather than the nature of activity was often more significant. In order to add more persistence to this display, we extended it so that the rings ‘ripple out’ from the document icon. Up to three additional concentric rings indicate recent activities. These three rings are not, as is the case with the inner ring, directly attached to the icon but have a small separating space.

The first, innermost ring continues to be a persistent indicator of the files state, allowing users to easily distinguish between an untouched file and a one that has been edited or read. Note that rings now only change color as a result of read and write events. This first ring continues to show the color of the person who has most recently interacted with the file. So, one user’s activity initially activates the fist ring bordering the file dot icon, but subsequently ripples towards the outside before disappearing altogether. The second ring’s color shows the second most recent person to interact with the file, and the same holds with the third and fourth rings. For all rings new activity on the document would cause older activity to ripple out, but the “rings” indicating activity can also disappear when reaching a specific fadeout time. We extended the 2nd, 3rd, and 4th rings with fadeout in order to more specifically give the users an understanding of more recent events and which documents have most recently had activity. The fadeout of the activity can be set to equal time for both read and write events or to different times depending on which of these two events it represents. A write, being considered a “heavier” event with more possible impact on the document, could be given a longer time before fading out compared to a read event.

The difference of behavior of these two types of rings is visually indicated to the user in two ways. First, the permanent ring is attached to the file. Second, the two types of rings are differentiated by the gap between them.

Figure 3 depicts the most recent history of file “Milestones.doc”, belonging to the “red” user (shows up as medium gray in grayscale printing). Around the icon we can see that three different users with three different colors have left traces of activity. The most recent activity is by the user with yellow (the lightest color), which activity also gave rise to the third most recent event indicated by the third ring. The user with blue (the darkest color) caused the fourth most recent event, which display of activity over the file will be rippled out of the visualization when a new activity occurs. The second most recent activity over Milestone.doc originated from the owner of the file itself, the “red” user.

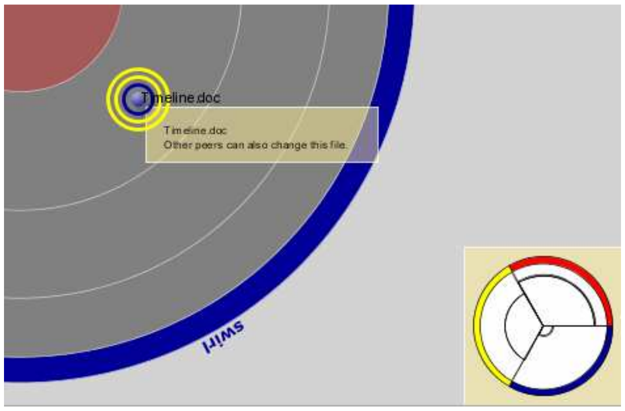

Figure 4: History Pie

4.2 History Pie

The extension from rings to ripples allowed for more activity to be displayed but since events ripple or gets pushed out they will not display more than the 4 most recent activities. We designed the “history pie” to provide a complete temporal description of all activities on a file over the full duration of a session. It could also be set to only convey activities under a more recent, shorter period of time. This builds on the same concern to see more than immediate activity, as indicated in the study, but takes a broader view. The same basic design principle used for ripples – that records of activity start towards a center and ripple towards the edges – is the basis of this display which presents more history. While the rippling rings indicate only immediate activity, this view shows the entire history of activity over a particular file.

This history is displayed on a smaller version of the circular pie interface element, which takes on the same organization, orientation, and color assignment of the users as the central display. This view is displayed when the user mouses over a document icon in the main interface. When this happens, the history of previous activities over that document is displayed in the small history pie in the lower right corner of the interface. Each historical action is indicated by an arc in the small display. Arcs are drawn in the radial “pie-slice” section that corresponds to the user whose actions are represented, and they are drawn on a timeline that stretches from the recent past at the center to the distant past at the edge of the pie. The effect is rather like the rings that indicate the growth pattern of a tree.

Figure 4 portrays a mouse-over of the file Timeline.doc, a file owned by the blue user “swirl”. The mouse-over triggers the display of the history pie. By only viewing the icon and its rings one can see the most recent history but viewing the history pie of the file more information is given. The icon shows that the “blue” user (lower right) most recently had activity over the file and earlier in time the “yellow” user (middle left) had touched the file. But the history pie, set to display the full history, shows that the “red” user (upper right) was the first user to touch the file. The display does not, however, differ in visualizing read or write events. The history pie gives the user a good indication of which users have been interested in the file in question and when in time during the session they were active on it.

4.3 Activity Wear

The history display provides a convenient view showing the activity of all users over one document. This is complemented by a view to show the activity of each user over all files. Again, this responds to the need for “overview” indicated in our study. Drawing on the idea of “edit wear and read wear” introduced by Hill et al. [10] in which repeated actions result in patterns of wear on the artifacts over which they are performed, we use the edge of each pie slice to display an indication of the accumulated history of an individual’s action. Each user’s activity is calculated relative to the other users’ activities. A maximum width border indicates a very active user and a thin minimum width a relatively inactive user. This means that the inactive user could in fact also have been quite active but compared to the total activity of the session he/she is considered relatively inactive. Reading a file and writing to a file are the two types of activities that we measure. The idea here is to be able to tell the difference between particularly active users and relatively inactive ones. It is not because we take activity or inactivity to be signs of inappropriate or problematic behavior; rather, we want to make any differences between people’s roles and apparent activities and their actual actions visible in the interface. The activity wear can be set to represent the activity of the user over the whole session or during the most recent time period. The later choice will display a user as active only when it is in the recent past. When some time has passed the users activity wear will shrink down and the border display get thinner. This border will at a glace give the users an understanding which user is the most active at the moment.

In Figure 5 the width of the users pie slice’s displays that user “swirl” (lower right) has been the most active user during the most recent 5 minutes and user “lina” (upper right) the most relatively inactive.

4.4 User Characterization

Each of these three previous visualizations has extended the concreteness and immediacy of the original Swirl interface with mechanisms to make aspects of history available. However, the history that has become manifest in these views is the history of a particular session, or the part of the session that any given user might see. Most people, however, work together over long periods of time and might be engaged in multiple sessions. The history of activities over time provides another useful source of information. In this case, what we want to know is whether the system configuration that we encounter at any given point is what we might expect. We believe that, while normative distinctions between “acceptable” and “unacceptable” security are impossible to determine, distinctions between “familiar” and “unfamiliar” or “usual” and “unusual” can more easily be incorporated into user practice.

One opportunity to do this exploits the fact that most of the use of Impromptu is based on personal devices, and particularly laptops. By examining the Address Resolution Protocol (ARP) [12] cache, we can determine whether a user name is associated with the hardware Ethernet address that we expect.[1] When someone appears on a different address than we have seen before, it may simply be because they have a new laptop or a new network card;

[1] MAC addresses can be spoofed, of course, and so there are dangers on relying on this in a real, rather than illustrative, case.

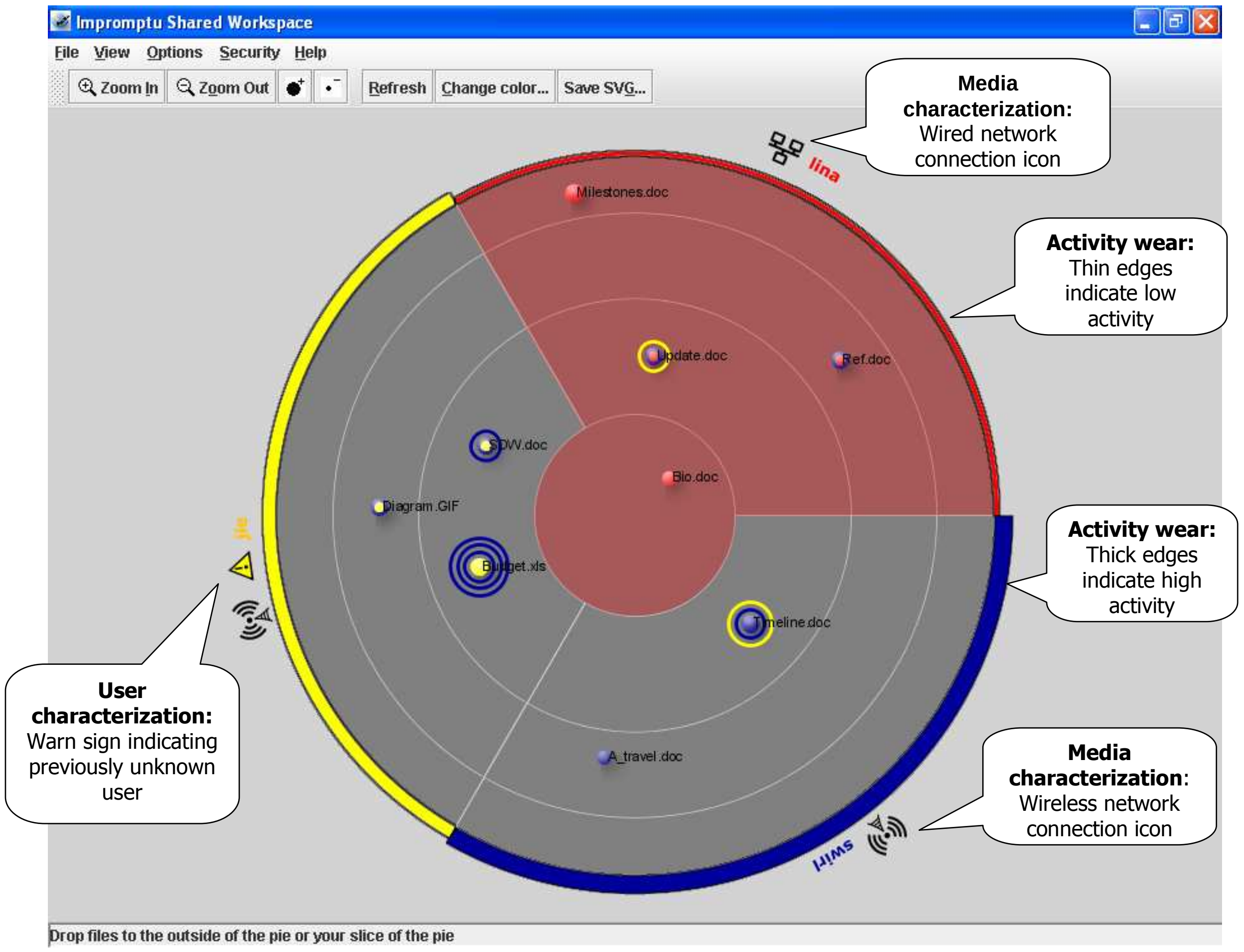
Impromptu Shared Workspace
File View Options Security Help
Zoom In
Zoom Out
Refresh
Change color...
Save SVG...
Media characterization: Wired network connection icon
lina
Milestones.doc
Activity wear: Thin edges indicate low activity
Update.doc
Ref.doc
SDW.doc
Bio.doc
Diagram.GIF
jie
Activity wear: Thick edges indicate high activity
Budget.xls
Timeline.doc
User characterization: Warn sign indicating previously unknown user
A_travel.doc
Media characterization: Wireless network connection icon
swirl
Drop files to the outside of the pie or your slice of the pie

Figure 5: Impromptu Client with Activity Wear, User Characterization, and Media Characterization

or it may indicate a man-in-the-middle attack. Following our usual procedure, our goal is simply to make clear the differences that might allow one to make an informed judgment.

In our implementation, alert icons are used to indicate unknown or unexpected mappings of users to Ethernet connections. In Figure 5 we see the screenshot from the screen of user “lina” (upper right). User “swirl” (lower right) is displayed as a known user meaning its hardware Ethernet address matches the address stored for this username. That is the user “lina” has been in a previous session with user “swirl”. User “jie” (middle left) is on the other hand an unknown user to user “lina” and a warning triangle beside the username “jie” displays this.

4.5 Media Characterization

The previous discussion of visualizations has focused primarily on representations of historical information – dealing with users that join a particular session, and the files they share. This is clearly relevant to people’s activities. However, the characterization of users is certainly not the only relevant feature here. We are interested in understanding the ways in which the network is configured, and one starting point for this is the method through which users connect to the application.

While it may be immediately apparent to a single user how he or she connects to any shared workspace (using a wifi-enabled laptop, a wired desktop, or a wireless handheld device) simply by virtue of using that device, this information is generally not apparent to the other users of the shared workspace. This concept of keeping the connection medium transparent to the application draws its roots from the TCP/IP stack, which owes much of its success to its ability to mask the intricacies of different media on the lower layers of the protocol stack with its higher, media-independent layer. While this transparency is useful in some settings, we believe that revealing the connection method of the users of the workspace presents the opportunity for users to make more informed decisions about their sharing activities.

Consider some examples. Traditional (coax) Ethernet is a shared medium, in which all packets traverse the same cable path, making them potentially available to all hosts. In twisted pair Ethernet, hubs redistribute information this way, while switches do not. The question of precisely how one is connected to the network, then, has important implications for data visibility – and, of course, this might be a feature of how others are connected to the network too. The introduction of VPN and wireless networks introduce further complexities. Our intention initially is not to display the immediate data leakage of wireless data transmission, as in Kowitz and Cranor's work [11], but rather to convey the notion that different sorts of connections hold different sorts of consequences. Media transparency, in other words, is a good idea for system interoperability but may be a poor idea for usable security.

One method of revealing the details of a user's connection involves the inspection of the network interface details at the client, allowing us to determine whether the connection is a wired or wireless type. Additionally, we may examine whether the connection is encrypted by means of a VPN tunnel. Here, as in the user characterization visualization, icon representations are used to indicate the type of connection (e.g., wireless 802.11a/b/g or wired Ethernet) used by each participant connecting to the Impromptu application. In Figure 5 these icons are situated adjacent to the usernames. The users "swirl" (lower right) and "jie" (left), are both connected to the session over a wireless channel; "lina" (upper right), on the other hand, is using a hard-wired Ethernet connection. Thus, in this example, the "lina" might be concerned that an unknown user "jie" has performed an action on her over a wireless connection.

Again, we feel that it is impossible – both for the user as well as the system – to make clear-cut distinctions between "good users" and "bad users." However, the less abstract distinctions between "wireless" and "wired," or "using VPN" and "not using VPN" are concepts that are more readily understood by users. While a user may initially be warm to the idea of sharing his files to an exclusively-wired Swirl session, his opinion may change upon receiving a notification that a new user has connected wirelessly. The introduction of a wireless laptop to the Swirl session brings with it the potential for information leakage through the wireless channel. Our visualization is meant to be used as a method of notification: the decision of whether or not this is a concern – and what action should be taken (the movement of one's files to a "less-shared" area, for example) – is to be made by the user.

5. DESIGN CONSIDERATIONS

The goal of these efforts has been to extend the range of the visualizations in the Impromptu prototype, in line with our goal of providing people with insight into their system behavior in support of informed decisions. In our initial design, we found that concreteness was a key property. Not only did it support the integration of configuration and action that was one of our basic design principles, but it also provided a rich metaphor for collaborative interaction. By concreteness, here, we mean the way in which the elements form which interaction is constructed have a direct, single, visible manifestation in the interface. People are not abstract entities, but represented as regions; access levels, similarly, are visible as regions of the interface; and all files in the system are visible concurrently, as individual objects.

This concreteness also gave rise to a concern that the interface manifest the same appearance to all users. Previous studies in collaborative systems have shown that this can be particularly important for collaborative applications in support of face-to-face interaction; it supports easy mutual reference and disambiguation [15]. Intriguingly, though, the addition of some of these new features begins to question this.

User characterization, for example, does not result in the same visual appearance for all users. One obvious example is that one is always a familiar user to oneself, but not always to others, resulting in different appearances. More generally, what user characterization presents is a per-user view of interaction history, and so must inherently differ from user to user. When we incorporate not just the history of files but also the history of users and interactions into the interface, then we begin to introduce elements that must challenge our initial goal.

File history, however, also presented some unexpected challenges. Particular problems arise from the fact that the configuration of a session may change over time, as people join and leave. This raises a question for displaying the history of a file – just what history should be shown? What period is the basis for a historical view? Maintaining a common view for all suggests that history should be recorded per-session; that is, the entire history of that file during the session would be recorded, whether or not any particular user had participated in the session for that entire period. This clearly, however, discloses information that a user would not otherwise have had access too. The alternative is to restrict history for each user to be just the history over the period during which they participated, although of course if users have different participation trajectories, then they will see different histories. Even more difficulty is introduced by the fact that users might join, leave, and then later re-join. In our current implementation, in fact, this generates a curious case where the continual participants get to see the entire history for the intermittent participant, while that participant may see only the history since their most recent arrival.

Outside of this edge case, though, the principle that describes the resulting design – that the interface always show only what you might have seen yourself if you had been watching the window continually – seems a reasonable one, and one that can be easily explained. It does, however, result in a loss of concreteness.

6. FUTURE WORK

While our new use of rings, ripples, the history pie, activity wear, user and media characterization address the issues of temporality and feedback on new users that arose in the study, these features are themselves untested. In our future work, we plan to conduct a user study to evaluate these features. The study will have participants performing short game inspired tasks designed to target the new features and to evaluate their use. Compared to the first study, which was designed to be a broad examination of the initial design, this future study will target more specific evaluations of the history visualization features. The study will also include a comparison of tasks performed with and without these new features.

Our previous research had confirmed that security is not confined to the system itself, but rather is spread across the system and the contexts within which it is used. There are two relevant contexts – a physical context and a working context. The physical context of use is face to face collaboration; Impromptu was designed not to support distance or distributed collaboration, but rather as an adjunct to face to face work. People talked to each other a great deal while using Impromptu, commenting on their actions, describing their plans, and of course talking about the work that they were doing. The use of Impromptu as a support, rather than a replacement, for face-to-face interaction is clearly important in the design. The working context is slightly more problematic. File sharing is rarely an end in itself; it is a means to support other working activities. Impromptu, then, is expected to be used alongside other applications. In our early trials, we noted that these other applications would sometimes obscure the Impromptu system, making it harder to notice changes and updates. We are looking, therefore, at a range of ways of conveying information about shared activities to people, not only through a dedicated interface but also through ancillary displays that can augment other interfaces.

To this end we are developing a “thin client” which will provide a summary view of the contents of Impromptu as well as visual indicators of activities and summaries of activity history. This client is intended to run on a PDA. Future work will present this client as well as a study of its usability.

7. CONCLUSIONS

Our approach to usable security is a holistic one. Rather than focus on the usability characteristics of particular software components, we aim to support the practices of security; the ways in which people carry out their work in ways that might be more or less secure. In our research, the emphasis is on activities where security is not or should not be the primary activity of using computing. That is, privacy and security have become issues in almost all computing applications but we have followed a general approach of how people might focus on their primary objectives and not secondary issues.

This approach is complementary to a more traditional usability focus; however, it provides a richer basis for understanding security “in the wild” and for thinking more broadly about future application developments.

In previous work, we have not previously been able to present empirical evidence that visualization does allow people to incorporate security concerns into their work in an effective manner. Here, we have presented the results of an evaluation emphasizing open- ended, naturalistic use of the testbed application which incorporates a range of visualization features. The initial trial data presented here bears out our hypothesis.

More usefully, perhaps, it also turns our attention to a set of critical design criteria. We have been focused on security as a collective practice [8]. That is, we are concerned not with one person’s action and another, but people “doing security together.” The emphasis on concreteness that characterized our initial designs has proven particularly important in this regard.

The collective visibility of action that it provides in turn supports the emergence of collective norms. Accordingly, we have been attempting to extend this concreteness into the temporal dimension, so that historical patterns of action can also become visible collectively. This does, as we have noted, lead into some complex questions as we grapple with the problems of different historical views. We expect these to surface, too, in our ongoing work with multiple interfaces. Our empirical results give us some confidence in the generality of the approach, however.

While strict usability can provide important results that reduce specific problems in the use of networked information systems and applications, it must inherently do so within fixed terms. Our concern is with the ways in which people appropriate information technologies and create new ways of working. By helping computer users to see further into the systems and networks that support their activities, we hope to see further ourselves and inquire into new forms of technological practice.

8. ACKNOWLEDGMENTS

We would like to thank Rogerio DePaula, Charlotte Lee and Xianghua Ding for their invaluable assistance in performing the study. This work was supported in part by the National Science Foundation under awards 0541462, 0133749, 0205724, 0326105, 0527729, and 0524033, and a grant from Intel Corporation.

9. REFERENCES

[1] Jetty Java HTTP Servlet Server, Mort Bay Consulting <http://jetty.mortbay.org/jetty/>.

[2] The Jakarta Slide Projects, <http://jakarta.apache.org/slide/>

[3] Altman, I. (1975). The Environment and Social Behavior. Privacy Personal Space, Territory and Crowding. Monterey, CA. Brooks/Cole Pub. Co., Inc.

[4] Altman, I. (1977). Privacy Regulation: Culturally Universal or Culturally Specific? Journal of Social Issues, 33(3), 66-84.

[5] DePaula, R., X. Ding, et al. (2005). In the Eye of the Beholder: A Visualization-based Approach to Information System Security. International Journal of Human-Computer Studies (IJHCS) Special Issue on HCI Research in Privacy and Security, 63(1-2), 5-24.

[6] DePaula, R., X. Ding, et al. (2005). Two Experiences Designing for Effective Security. In Proceedings of the 2005 Symposium on Usable Privacy and Security (SOUPS 2005), Pittsburgh, PA.

[7] DiGioia, P. and P. Dourish (2005). Social Navigation as a Model for Usable Security. In Proceedinfs of the 2005 Symposium on Usable Privacy and Security (SOUPS 2005), Pittsburgh, PA.

[8] Dourish, P. and Anderson, K. In press. Collective Information Practice: Exploring Privacy and Security as Social and Cultural Phenomena. Human-Computer Interaction.

[9] Goland, Y., E. J. Whitehead, et al. (1999). HTTP Extensions for Distributed Authoring -- WEBDAV, Internet Engineering Task Force: 1-94, RFC 2518.

[10] Hill, W. C., J. D. Hollan, et al. (1992). Read wear and edit wear. In Proceedings of the ACM Conference on Human Factors in Computing Systems, (CHI `92), Monterey, California.

[11] Kowitz, B. and L. Cranor (2005). Peripheral Privacy Notifications for Wireless Networks. In Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society, Alexandria, VA.

[12] Plummer, D.C. (1986). Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware, IETF RFC826.

[13] Silva Filho, R. S., D. S. C. R. B., et al. (2003). The Design of a Configurable, Extensible and Dynamic Notification Service. In Proceedings of the Second International Workshop on Distributed Event-Based Systems (DEBS'03), San Diego, CA.

[14] Steinberg, D. and S. Cheshire (2005). Zero Configuration Networking: The Definitive Guide, O'Reilly Media.

[15] Tatar, D., Foster, G., and Bobrow, D. 1991. Designing for Conversation: Lessons from Cognoter. International Journal of Man- Machine Studies, 34(2), 185-209.