



Institute for Software Research
University of California, Irvine

Incorporating Social Navigation into a Usable Security System



Carolina Johansson
University of California, Irvine
Uppsala University, Sweden
cjohans@ics.uci.edu

July 2007

ISR Technical Report # UCI-ISR-07-3

Institute for Software Research
ICS2 217
University of California, Irvine
Irvine, CA 92697-3455
www.isr.uci.edu

www.isr.uci.edu/tech-reports.html

Incorporating Social Navigation into a Usable Security System

Carolina Johansson
Visitor from Uppsala University

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

ISR Technical Report # UCI-ISR-07-3

July 2007

Abstract:

Security and usability is an evolving field of research, growing bigger by the years as the conclusions are drawn that security systems that are not understood by their users are as good as no security at all. Theoretically secure systems are no guarantee for security. Instead they have to be understood by the end users to become effectively secure.

The Impromptu file sharing application is one of the test beds used by the Swirl research group of the University of California, Irvine, US. Impromptu makes visible the configuration, activity and implications of available security measures, thereby allowing end user to make informed security choices resulting in increased effective security.

In this report I will describe my experiences and findings of the design, implementation and user study of my four new extensions to the Impromptu application. These extensions are four distinct features that display temporal history within a session and also the relationship between users and sessions. These four features are to give the users a better understanding of their situation and the activities in the session, thereby aid them in assessing their own security.

This report will describe the theory behind the design, my design of the features, the selection criteria for the features chosen, the implementation process and the problems encountered, the user study design and finally the results of the study.

My four new extensions of the Impromptu were described and published in the “Seeing Further: Extending Visualizations as a Basis for Usable Security“ paper that was presented at the 2006 Symposium On Usable Privacy and Security (SOUPS) conference, Pittsburgh, United States.

Incorporating social navigation into a usable security system

by

Carolina Johansson, Uppsala University

Supervisor: Prof. Paul Dourish, University of California, Irvine, US

Reviewer: Prof. Jan Gulliksen, Uppsala University, Sweden

Examiner: Anders Jansson, Uppsala University, Sweden

Abstract

Security and usability is an evolving field of research, growing bigger by the years as the conclusions are drawn that security systems that are not understood by their users are as good as no security at all. Theoretically secure systems are no guarantee for security. Instead they have to be understood by the end users to become effectively secure.

The Impromptu file sharing application is one of the test beds used by the Swirl research group of the University of California, Irvine, US. Impromptu makes visible the configuration, activity and implications of available security measures, thereby allowing end user to make informed security choices resulting in increased effective security.

In this report I will describe my experiences and findings of the design, implementation and user study of my four new extensions to the Impromptu application. These extensions are four distinct features that display temporal history within a session and also the relationship between users and sessions. These four features are to give the users a better understanding of their situation and the activities in the session, thereby aid them in assessing their own security.

This report will describe the theory behind the design, my design of the features, the selection criteria for the features chosen, the implementation process and the problems encountered, the user study design and finally the results of the study.

My four new extensions of the Impromptu were described and published in the “Seeing Further: Extending Visualizations as a Basis for Usable Security“ paper [1] that will be presented as the 2006 Symposium On Usable Privacy and Security (SOUPS) conference, Pittsburgh, United States.

Sammanfattning

Data säkerhet och användbarhet är ett område inom människa dator interaktion som på senare tid vuxit sig större och större. Då utveckling och uppbyggnad av säkerhets system förut var mer fokuserade på att uppnå teoretisk säkerhet, så har den senaste tidens studier och forskning inom området fått programmerare och system utvecklare att mer och mer inse den mänskliga faktorns inverkan: säkerhets system som inte förstås av sina användare inte är säkra system. De är till och med att jämföra med att inte ha någon säkerhet alls.

I vissa fall kan t o m dessa oanvändbara system ha en förvärrande effekt på en säkerhets situation än innan systemet sattes i effekt. Hur är det möjligt? Jo, en användare som inte har förstått systemet och följaktligen konfigurerat systemet fel har inte alls den höjda säkerhet han förväntar sig från det nya systemet eller programmet. Trots att situationen inte har förändrats, pga. fel användning, tror sig nu användaren dock vara på säkrare mark och kan följaktligen dela med sig mer information och bete sig mer oaktsamt än innan systemet var satt i funktion. Slutsatsen som fler o fler inom området drar är att teoretiskt säkra system inte är någon garanti för säkerhet. Systemen och programmen måste bli förstådda av de tilltänkta användarna för att garantera, inte bara teoretisk säkerhet, utan även effektiv säkerhet.

Impromptu är ett ad hoc fil delar program som är utvecklat av forskar gruppen inom projektet Swirl, vid University of California, Irvine (UCI) i USA under ledning av professor Paul Dourish. Den ligger som grund till gruppens forskning inom att uppnå effektiv säkerhet genom visualisering av aktivitet inom ett program. Hypotesen som ligger som grund, är att hjälpa användare att bättre förstå säkerhets system genom att göra säkerhets mekanismer synliga. Iden vilar på att användare som inte förstår systemets delar rätt kan heller inte skapa sig korrekta uppfattningar om hur de bör agera och konfigurera systemet för att uppnå en högre säkerhets nivå inom systemet.

Impromptus tekniska infrastruktur möjliggör visualisering av konfiguration, aktiviteter/händelser och konsekvenser av olika säkerhets nivå val för användaren som på så vis ska kunna bilda sig en bättre uppfattning om hur systemet fungerar, vad som händer och vilka följer användarens val får. För att uppnå detta har man baserad Impromptu på en kombination av tre huvudsakliga design principer: visualiserings mekanismer, mångfasetterad händelse registrerande arkitektur och integrationen av händelser och konfiguration.

Användare av Impromptu delar med sig av filer och ser dess filer från en gemensam synvinkel och kan manipulera dessa filer. Användargränssnittet visar när användare läser och ändrar på filer och möjliggör för användarna att flytta filer mellan olika säkerhets nivåer genom att dra dem runt i gränssnittet.

Varje användare ser samma grafiska gränssnitt med en cirkulär "paj" i mitten. Pajen är uppdelad i paj bitar, där varje användare representeras av en paj bit. Varje fil som delas med i Impromptu representeras som en cirkulär punkt inom något område i pajen. Pajen är uppdelad i olika säkerhets nivå områden från att bara kunna se om en fil finns till att kunna använda filen även om dess ägare skulle avsluta och lämna sessionen. Genom att släppa ner en importerad fil in något av de olika områdena så väljer användaren sin säkerhets konfiguration för den filen. En ring runt varje punkt som representerar en fil, visar också den senaste händelsen över filen, dvs. om någon just läst den eller ändrat den.

Original versionen av Impromptu visualiserar bara händelser som sker ”just nu”. Användare kunde ej genom visualiseringarna få en fullaktig bild om dåtida händelser. Bara den allra senaste händelsen på en fil visualiserades.

Mitt examens arbete var att utveckla vidare mer visualiseringar för att upplysa användarna om det förflutna inom en session och även relationer emellan sessioner. Detta resulterade i en utbyggnad av den originella kod basen med fyra nya visualiserings komponenter. Den första visualiseringen byggde ut den redan existerade ringen runt en fil som visar den senaste händelsen över en fil. Denna nya komponent utökade varje fil till att visualisera upp till fyra ringar runt sig, på så vis visualisera mer av den mest nyliga historien över en fil, nämligen de fyra senaste händelserna per fil. Var och en av de tre extra ringarna innehåller också en timer som räknar ner tiden tills denna ring inte längre räknas som aktuell och således försvinner. Den andra visualiseringen kompletterade den första ring visualisering med att visa hela historien över en fil. När en användare pekar musen på en punkt (en fil) så visas en mindre historie paj i nedre högra hörnet av arbetsytan. Den har samma orientering och placering av användares paj bitar som den stora pajen men är tom inuti. Istället fylls den upp med bågar i respektive paj bit som representerar en händelse av den användaren på denna fil. En båge nära centrum av pajen är en mycket nylig händelse medan en båge i utkanten representerar en händelse för länge sedan.

Dessa två första visualiseringar var centrerade över händelser över enskilda filer. Min tredje visualisering tog istället till vara de ackumulerade totala händelserna per användare och visar vilken av användarna som varit mest aktiv under den senaste perioden relativt de andra användarna. Genom att variera bredden på den färggranna kanten av en användares paj bit, så visas en relativt aktiv användare med bred färgad kant och en relativt inaktiv användare med tunn färgad kant. Den fjärde visualiseringen visar samband mellan olika sessioner som användarna har varit i. Varje användare är på den lokala användarens paj visualiserad antingen som känd eller okänd. En känd användare är en användare den lokala användaren har mött på tidigare, under tidigare sessioner. För att räknas som känd användare måste den fysiska adressen som användaren är uppkopplad från och det användarnamn som används matcha ett förut sparad adress/namn par. Alla användare vars adress/namn par inte har setts förut anses som okända och visualiseras genom en gul varningstriangel innan användarnamnet som står längst utanför paj biten på den användaren.

Den här rapporten beskriver hur jag gjorde mitt val av visualiseringar, hur de designades och implementerades och teorierna bakom dem. Den illustrerar först en djupgående analys av problemet: att den ursprungliga Impromptu programmet bara visualiserade händelser just då de hände och den teori som ligger till bas för hela programmet. Vidare beskrivs implementerings fasen och de problem som stöttes på och olika evalueringar mina komponenter undergick. Slutligen presenteras den användare studie jag genomförde för att testa användbarheten hos mina nya visualiseringar. Resultaten av studien diskuteras och analyseras i vardera kapitel. Som appendix finns några av de dokument jag producerade inför användar studien och även tre sammanfattningar av studie resultaten.

Mitt jobb och nya visualiseringar publicerades i en vetenskaplig artikel “Seeing Further: Extending Visualizations as a Basis for Usable Security“ [1] som kommer att presenteras på 2006 Symposium On Usable Privacy and Security (SOUPS) konferensen i Pittsburgh, USA.

Preface

This is the report of my Master's thesis project within the Human Computer Interaction field, with focus on Security and Usability. It is the last stage of my education at the Master of Science in Information Technology Engineering program at Uppsala University, Uppsala, Sweden.

My Master's thesis project was conducted on site at the University of California, Irvine (UCI), in the US. I held the position of visiting researcher for Professor Paul Dourish at the Institute for Software Research for the full eight months of my stay.

My project was done within the Impromptu project which is part of a larger project named Swirl, with Professor Paul Dourish and Professor David Redmiles as principal investigators. My project was conducted in collaboration with the other projects within Swirl and the Swirl research group members.

I would like to thank Professor Paul Dourish for inviting me to conduct my Master's thesis within the Swirl project and for all the help, inspiration and knowledge he has passed my way during my stay. I would also like to thank all members of the Swirl research group for help, opinions and valuable testing of my designs and software as it was being developed and to Jennifer Rode for conducted the cognitive walkthrough of my features with me. Finally a big thanks to Kari Nies, Paul DiGioia, Roberto Silva Filho and David Nguyen, who all devoted long hours of their time to help me with the user study.

This work was supported in part by the US National Science Foundation under award 0326105.

Table of Contents

Incorporating social navigation into a usable security system.....	1
Abstract.....	2
Sammanfattning	3
Preface	5
Table of Contents	6
1. Introduction.....	8
1.1 Background	8
1.1.1 Swirl.....	8
1.1.2 Impromptu	8
1.2 Problem Specification.....	9
1.3 Aims.....	10
1.4 Limitations.....	10
1.5 Structure of this report	11
2. Method.....	11
2.1 Introduction	11
2.2 Literature study.....	11
2.2.1 Usable Security	12
2.2.2 Social navigation.....	13
2.3 Overview of my project	14
2.3.1 Analysis of the system.....	14
2.3.2 Implementation	14
2.3.3 User Study	15
2.3.4 Results	15
3. Analysis.....	15
3.1 Prior work.....	15
3.1.1 The theory behind the design.....	15
3.1.2 The design.....	16
3.1.3 The architecture	17
3.1.4 The graphical interface.....	18
3.1.5 Prior user study	19
3.2 My work.....	21
3.2.1 Rings and Ripples	22
3.2.2 History pie	24
3.2.3 Activity wear	26
3.2.4 User characterization.....	28
3.2.5 Designs not implemented	29
4. Implementation.....	30
4.1 Programming language and tools	30
4.2 Implementation of features.....	30
4.2.1 Rings and ripples.....	30
4.2.2 History pie	31
4.2.3 User characterization.....	31
4.2.4 Activity wear	32
4.3 Problems encountered.....	32
4.3.1 Original code base.....	33
4.3.2 My designs.....	34
4.4 Cognitive walkthrough.....	34

4.4.1 Issues concerning original code base	35
4.4.2 Issues concerning my features	35
5. User Study	36
5.1 Purpose	36
5.2 Study design	36
5.3 Task design	37
5.3.1 Cards task	38
5.3.2 Lists task	38
5.4 Study documents	39
5.4.1 Post task questionnaires	40
5.4.2 Brief interview	41
5.5 Participants	42
5.6 Test observers and confederate	42
5.7 The technology and the setting	43
6. Result	43
6.1 Post task questionnaires	44
6.1.1 Cards task – rings/ripples	44
6.1.2 Lists task - history pie	45
6.1.3 T-test	45
6.2 Posttest questionnaires	47
6.3 Interviews	48
6.3.1 Rings/ripples	49
6.3.2 History pie	50
6.3.3 Activity wear and screenshot 1	51
6.3.4 Screenshots 2 and 3	52
6.3.5 Users did not like	52
6.3.6 Users would like to see more of	53
7. Discussion	54
8. Future work	56
9. Conclusion	57
10. References	59
Appendix 1	60
Appendix 2	61
Appendix 3	62
Appendix 4	64
Appendix 5	65
Appendix 6	66
Appendix 7	67
Appendix 8	68
Appendix 9	69
Appendix 10	73
Appendix 11	75
Appendix 12	79

1. Introduction

1.1 Background

1.1.1 Swirl

My Master's thesis project is in the Human Computer Interaction field, within the area of Security and Usability. It was conducted at the University of California, Irvine (UCI) in the US. My project was based on the Swirl project at the Institute for Software Research, ICS, at UCI [2].

The Swirl project focuses on effective security awareness through visualization. The idea of the project is to develop mechanisms to help end users manage their own security. It works from the standpoint that security measures that aren't understood and agreed to by everyone don't work. If users can not understand the security mechanisms of a software system they can not make effective decisions of how and when to use them. The hypothesis is that by making these mechanisms visible to the end users, they will be able to make more informed choices and thereby heighten the security level of the system.

1.1.2 Impromptu

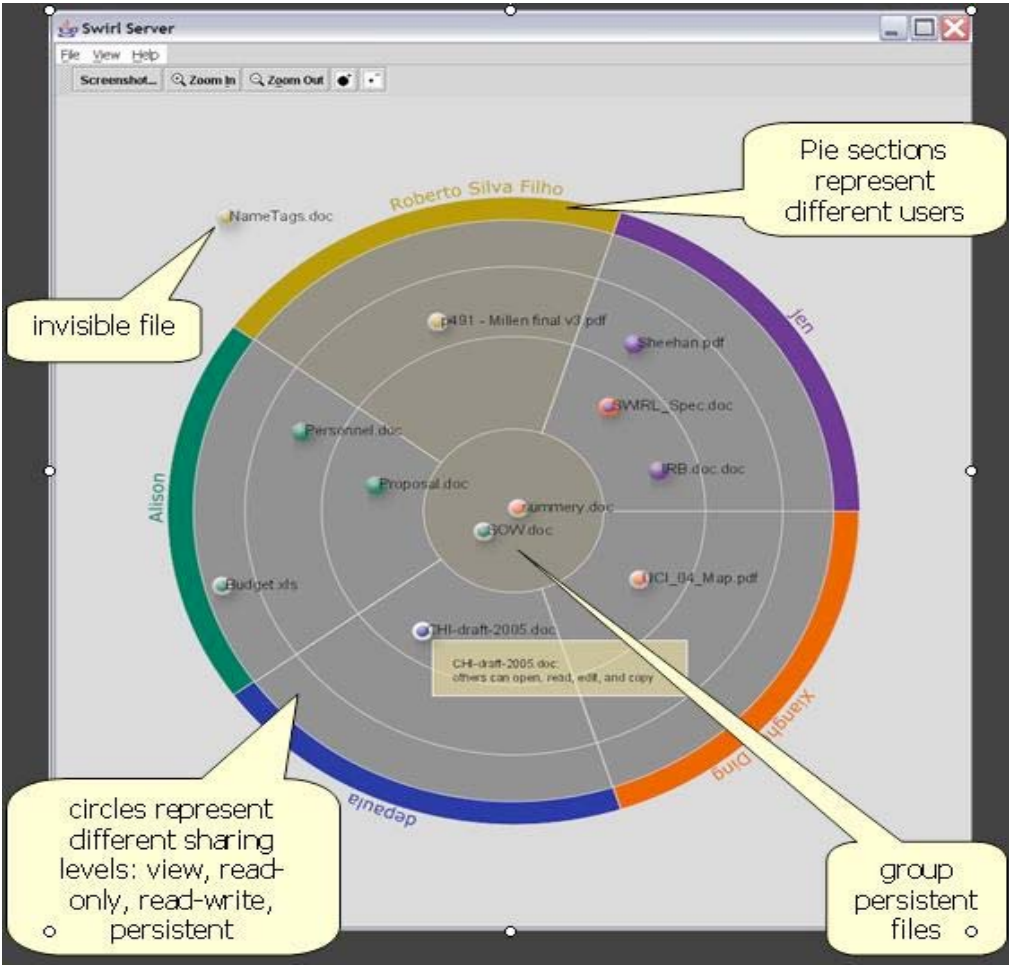
Impromptu is the name of the main project in Swirl and it is an ad hoc file sharing application that allows easy and secure file sharing. Impromptu Workspace allows users to share files and collaboratively view and edit these files. The interface visualize when peers read or modify shared files and allows for the users to modify the file sharing level by dragging the files around.

Each user is presented with the same graphical user interface (GUI), which has a pie in the center. Within the pie, each user is represented by a slice. Each shared file is represented by a dot. The local user's slice looks darker than other slices. Each slice displays what files that user is sharing. Each user can pick a color, which is used to depict the name of the user, a colorful border of the edge of the pie, and all files shared by that user. The outside of the pie represents the private space of a local user, where the user can put files that the user does not want to share yet.

By using a round-table metaphor it visibly shows the user the security level of shared files such as if other users can only see the file, if they are able to read it, if they are able to both read and write or if the file is persistent, meaning it will stay on the network even when the original owner has logged out. It also shows which users are connected to the ad hoc network, if a file has been changed by another user etc.

Impromptu has been coded in Java [3] and based on Scalable Vector Graphics (SVG) [4] which is a language for describing two-dimensional graphics and graphical applications in XML [5].

This description of the Impromptu was the current status of the application when I entered the Swirl group and started my project in October 2006. Figur 1depicts the Impromptu workspace with different features explained in text bubbles.



Figur 1: Screenshot of Impromptu-pie of the original version of the application

1.2 Problem Specification

The purpose of my Master’s thesis was to extend the existing system to not just visualize what is happening right at this moment but to also include visualization of what has happened in the past. To see if this additional information about past actions can be useful for the users to help them make decisions affecting their security.

A user study of the original version of Impromptu was conducted in spring of 2005 (Dourish et al, 2006 [1]). The results of this study strongly indicated that there was a need for visualizations of history of the users and their actions. In the study, a semi-naturalistic task had the participants collaborating on a research budget for a grant application. During the task the users had word documents open at most of the time. This lead to the Impromptu pie interface being obscured by the Word or Excel window and was out of the user’s vision for the majority of the task. The blinking around a dot that indicated an action on the file was easily missed by those participants who did not have the Impromptu application in the

foreground at that moment. In short, since the Impromptu application was not always in the foreground the users would miss a lot of the events that were happening and thereby be less aware of the situation. While working on a word document that would obscure the screen, the user would miss most of what was going on in the Impromptu workspace. This is a critical problem because when a user is not aware of the current situation and is not brought to attention when new events and changes occur, he/she is less able to correctly assess their own security situation.

The interesting part of this problem would be to see if additional visualization of the history of files and users can help users assess their security. To see if the visual info about for example the status and history of a file can help users make more informed decisions which may heighten their security. For example; they can see if a completely new user is entering the ad hoc. This may cause them to withdraw some shared file due to its sensitive content. Or the user can by watching the history of other similar shared files come to the conclusion that this type of file has been shared by many others and should therefore be relatively safe for him/her to share as well.

1.3 Aims

The aims of my Master's thesis were to enhance the Impromptu application with further history visualizations to facilitate for users to have a better understanding of what is going on when using the application.

My project was to add features to visualize actions taken place in the past, connecting the past to the present. This might help the users make more sense of the information they are already receiving while using the present Impromptu interface, by placing the current activity in the context of past actions. I was to add features that could be tweaked or replaced by other possible icons and time limits to accommodate for different work situations.

To see if my features made sense to the users and to see if they would use them in a fictive setting, I conducted a user study at the end of my project. The improvements of the application should help users with better understanding the current situation and to be able to see past actions.

1.4 Limitations

In my work I will discuss and address the features of the history visualizations that I have added to the application. There were further features discussed during the design phase but we limited my work to four features, due to limitations of time and limitations of real estate space of the screen. The original Impromptu application already existed and had been evaluated in a user study before I joined the project (described in chapter 3.5).

The user study conducted at the end of my project was limited to more specifically answer questions such as; do the users understand the history visualizations, can they make sense of what they mean, did the added features help them when carrying out their study tasks? In experience from the previous study (Dourish et al, 2006 [1]) my study was formed to aim at

more specific questions. The study was focused on evaluating the usefulness of the history visualization features and to see if they helped the users of better knowing what was going on in the session. When getting a better understanding of the activities going on, the user can better understand the situation and make more informed choices. Even though this study is not focused on specific traditional security features such as passwords and encryption, it is part of the bigger theory of using visualization of a system's different aspects to aid users in their assessment of their own security. The features and the study of them are about security, since they are part of the overall hypothesis that visualizations is a way to make users aware of what is going on.

A limitation of the study is that it was conducted in a laboratory setting where the users were given a quick introduction to the software before carrying out some tasks. Had I instead looked at how people learn and use the features over time and over many sessions, which is the actual setting scenario intended for the Impromptu application, the results would probably have been even better than the laboratory study did yield. Time limitations limited the study to the laboratory setting.

1.5 Structure of this report

This report is structured into nine chapters. The first chapter presents the background, problem specification, aims and limitations. The second and third chapters target the method and analysis. This is followed by a chapter each on the implementation of my work and the user study conducted at the end of the project. Chapter six describes the results of the user study followed by a discussion of the study results in chapter seven. Future work, conclusion, discussion and references will finish the report with a chapter each.

2. Method

2.1 Introduction

This chapter describes the way I have been working on my Master's thesis. I will begin to write about the literature I made use of and the theoretical study of usable security and social navigation. I will then present an overview of my project describing the analysis of my problem and the design phase following it. The implementation part of my thesis will then be explained and I will end by describing the User Study that was conducted at the end of the project and its subsequent results.

2.2 Literature study

By way of introduction, I started my project by exploring the area of usability and security and the social navigation concept. I found references and literature in books, scientific papers and via the Internet. I was also given good guidance and help from my colleagues in the Swirl

project of where to find information. My purpose was to familiarize myself within the area of my task, such as previous theories, designs and implementations. I also read up on the Swirl project and its functionalities.

2.2.1 Usable Security

Usability has different meanings in different contexts. Usability of security systems requires user interface design priorities that are not the same as those of general software. General software priorities can be such as efficiency, learnability or flexibility. In a security context, usability must be whatever priorities needed in order for the security to be used effectively (Whitten, Tygar, 1999 [6]).

The interaction between security and usability is a critical issue when designing secure systems. Yet it has been one of the most overlooked aspects of computer security. A security system is only as strong as its weakest link. The users of a system are frequently identified as the systems weakest link, due to the human imperfection and frailty (Whitten, Tygar, 1999 [6]). But until recent years the human factor in security has been neglected both by developers of security technology and those responsible for organizational security.

The necessity for usable security systems was first recognized in 1975 by Saltzer and Schroeder who identified the need of *psychological acceptability* in security systems (Saltzer, Schroeder, 1975 [7]). Their principal of psychological acceptability states that the security mechanisms of a system should add a minimum of extra difficulty to system. In short that means that performing an action or accessing some resource should be made no more difficult after the security mechanisms are introduced, than it was before.

Present security products are usually ineffective because the users do not behave in the way they are supposed to in the theoretical design. This is in part due to the fact that many users do not bother with security mechanisms, since they do not feel enough at risk for an attack. But more importantly; most users do not even know how to use the security systems correctly (Sasse, Flechais, 2005 [8]).

Zurko and Simon were among the first to point out that current security mechanisms make unreasonable demands on its users (Zurko, Simon, 1997 [9]). Later research has confirmed this by pointing out that current security tools are simply too complex for many users. The usability evaluation of PGP 5.0, by Whitten and Tygar in 1999 [6], is one of the most cited and widely known papers on usability and security. Titled “Why Johnny can’t encrypt” it reports how technically experienced persons failed to encrypt and decrypt their mail even after having received instructions and practice [6].

The conclusion to be drawn is that simply building a secure system does not assure its security (Bishop, 2005 [10]). Dourish and Redmiles argue that the critical problem of usable security does not lay in designing more secure systems, but designing useful and trustworthy systems that are effectively secure, rather than theoretically secure (Dourish, Redmiles et al, 2005 [11]). For a system to be secure it can not only be built to be theoretically secure. It has to be built to be secure when actually employed by the users (Tognazzini, 2005 [12]). Only usable security is effective security.

2.2.2 Social navigation

Social navigation is an approach to interaction design initially presented by Dourish and Chalmers (1994 [13]). The basic idea of social navigation is to make people's social trails through information spaces visible, to make available previous patterns and usage behaviors. In short, social navigation aims users in current situations by giving guidance by displaying where and how people have navigated the space.

The fundamental observation behind the social navigation approach is that in everyday life we navigate complex spaces by making use of the trails of other people. We do so not only by observing their spatial organization, but also by making interpretations and understandings of their activity. For example, if we enter a library the portion of dust covering each book will give us an indication as to which books are frequently and often read and which have been laying untouched for a long while. One could also imagine a library, which is for some reason never vacuumed. Eventually the floor will be covered by a layer of dust. By watching the trails left behind by the old librarian's dragging feet we can get a pretty good idea which the isles are that he/she most frequently visits.

It is important to note that the activity of others is in no way something that we must follow but is simply giving us cues as to how we could choose to act and navigate. We might for example be seeking to do what others have not done. The social navigation approach is not, in general, a distinction between what is wrong or right, good or bad, but is a way of helping the user understand a space in terms of the activities of others.

Dourish and Chalmers also drew upon the idea of "edit wear and read wear", introduced by Hill et al (1992 [14]). The idea of "wear" should be seen as similar to the "wear and tear" that physical objects are subjected to. Hill et al describe an interaction approach in which activity on an object will leave traces on the object itself. For example consider magazines at a hair dresser salon. Old magazines which have been frequently read will have dog-ears, be wrinkled and probably have a few coffee stains on them. A fresh new magazine will shine in comparison, being flat, un-wrinkled and clean. This notion of wear on an object, that the object can display past events and accumulated patterns of activity over it, is a key element to the social navigation approach (Dourish, DiGioia, 2005 [15]).

Since social navigation can clearly help people with understanding what is going on in information systems it can also be used to facilitate the understanding of security systems. While security is far from a navigational task we can however think of security applications as spaces where multiple persons act, and where their actions may be displayed as traces on the objects of the application. Dourish and DiGioia presents in their paper "Social Navigation as a Model for Usable Security" three ways in which social navigation can be used within security systems [15]. They suggest that social navigation can be used:

1. To show *the history of a user's action*. Artifacts of the application can display history and actions over it, display patterns and paths follow by other users.
2. To show *patterns of conventional use*. To display both which are the most frequent use and thereby also highlight the deviations from the conventional patterns. By doing so, a user can for example be made attentive to people communicating to unusual parties or using different servers than the usual.

3. To show activities *of others within a system*. In systems where some objects are in a sense shared, these objects can be used to show the activity of other users.

In my thesis I have incorporated all of the three themes described above.

2.3 Overview of my project

My work was essentially made up of two major phases or two major elements. The first was the phase of software design and implementation, with analysis of the system. The second phase was the user evaluation and the gathering, interpretation and calculations performed on the study results.

2.3.1 Analysis of the system

The analysis phase was composed of two parts. The first was to analyze the prior work. To get familiar with the Impromptu application I started out with using the system by myself and with other persons from the Swirl project group. The purpose was to experience the application in its current state and thereby see what information I observed and felt was lacking from the instrument. This was done before I had taken in information of the outcome of the prior user study (Dourish et al, 2006 [1]). I familiarized myself with the papers and publications concerning the application and got verbal information by the developers of the tool. I analyzed the theory behind the design and the design itself. Learned about the Impromptu architecture and studied its graphical interface. When acquainted to the system, I looked at the previous and initial user study (Dourish et al, 2006 [1]) and its data to find out what the users had problems with and what they would like to be able to see more of.

The second part of the analysis phase was the design of my own features. This part will describe the design, theory behind and how I chose the final version out of many for the exact structure of the features. In part I will also present those features that only made the initial design phase but were never implemented, partly due to time limitations.

2.3.2 Implementation

After having analyzed the Impromptu application and the data from the prior user study, I started working on different designs of a handful of parameters. In collaboration and after brainstorming with Prof Paul Dourish, I was able to present six different features that would extend the initial interface. These features were then presented and discussed upon with the research group of Impromptu developers with the final outcome narrowed down to four distinct features to implement. Throughout the Implementation, updates of the process were presented to the research group which would give me feedback and ideas. Test runs and debugging sessions were also performed on a continuous basis throughout the implementation. In the last phase of implementation we did a cognitive walkthrough with the usability expert of the group which addressed things to be changed, improved and updated

both concerning my new features and concerning the initial Impromptu test bed. The code base was written in Java [3] while using SVG [4] for the graphical visualizations.

2.3.3 User Study

After the Implementation was finished I conducted a user study to evaluate three of the four features that I had extended the application with. User characterization was not evaluated since I was conducting a laboratory study. A fair evaluation of this session-after-session related feature would only be possible when studying people using the application over a longer period of time and for multiple sessions. I designed the user study to target specific questions such as; do the users understand the history visualizations, can they make sense of what they mean, did the added features help them when carrying out their study tasks? The study was conducted over a 2 week period with 12 users in total with each session lasting about 80 min. I designed three short game-inspired tasks that were followed by questionnaires about the task and its results. Each session was concluded with a final brief interview of each user about his/her experience of the tasks and the application. The user study will be fully discussed and presented in chapter five.

2.3.4 Results

The study gave some very specific figures, describing the usefulness of the different features under evaluation. By doing comparison of the accuracy for with and without the rings/ripples feature I got a good result of how useful this feature was to the users. The same comparison was done for the accuracy of the user who used and those who did not use the history pie. I performed a statistical test on the data to prove its statistical significance and got positive results. The results of the study will be fully discussed in chapter six.

3. Analysis

3.1 Prior work

3.1.1 The theory behind the design

The goal of interface design and usability practices has traditionally been to hide the architectural complexity of the system and to remove security from the domain of the end-user. However, new research question this approach of making the system security transparent to the user, arguing that it is an insufficient or inadequate way to deal with privacy and security problems (Dourish, Redmiles et al, 2005 [20]). The more the architecture disappears into the background and become invisible, the harder it is for the user to see the implication of his/her actions and the extent to which his/her actions may cause security problems.

Consequently, when unexpected events occur, the user is often unable to respond correctly and take appropriate action.

In the area of peer-to-peer file sharing applications, studies show that users have major problems with using, understanding and configuring these applications. The problems results in users sharing documents that they never intended to share and disclosing private files (Good, Krekelberg, 2003 [23]). The goal with the Impromptu design was to make relevant security features visible to users, allowing them to observe and understand certain events on the network and this way be better prepared to take appropriate actions and also better understand the implications of their actions.

The technical infrastructure of Impromptu makes configurations, activities and implications of present security features visible to the end-user and thereby creates a foundation for the end-user to better understand and recognize the security situation (Dourish, Redmiles, 2005 [20]). To reach this goal, three major design principles were combined: visualization mechanisms, multi-source event-monitoring architecture and integration of action and configuration. By using visualization mechanisms, the users can see and assess the outcome of their actions. By providing feedback on hidden but relevant aspects of the system, the user is more likely to understand the relationship between his/her actions and the system's behavior. By integrating action and configuration into the same user interface, the two related activities of sharing and controlling are then seen in the same view. Traditional interfaces separate configuration and action, both in time and space, although in real life they are one and the same activity. The primary motivation for the Impromptu design is to improve security and privacy by making security and privacy features "apparent" rather than "transparent" (Dourish, Redmiles, 2005 [20]).

The overall ideas presented in this chapter is described in the paper "Two experiences designing for effective security" by Dourish, Redmiles et al (2005 [20]).

3.1.2 The design

Incorporating the three design principles presented in chapter 3.1, Impromptu is a graphical interface for ad hoc face-to-face collaborative activities. The example setting for using the application would be a café where people meet up and want to share files, or a work group collaborating on some documents for a common project. The target user of Impromptu is an experienced computer user. Impromptu is not designed to be a walk-up-and-use tool but rather to be continuously used and incorporated into a users work and practice. For each session the user will become more and more acquainted to the interface and its different features.

The Impromptu tool is designed to be an application for exploring the concept of integrating action and configuration and the concept of dynamically visualizing activity. Action and configuration is integrated by the means that when a user share's a file, he/she also sets the configuration, simply by choosing in which the security level to place the file in the workspace. Dynamic visualization allows a user to visually see the users who are present and connected to the same local network (subnet), the files being shared and the different security levels where the files are placed. The application also makes visible when an action occurs on

a file and allows the user to directly control and easily change the degree of sharing level for a file, by dragging it into a different area of the workspace.

Being a peer-to-peer file sharing application, Impromptu was never designed to be the best file sharing system but rather to be used as a test bed for the design principles of the two interrelated activities of sharing and controlling, integrated into the same user interface.

3.1.3 The architecture

The Impromptu architecture is illustrated in Figur 2. The infrastructure is based on a federated model and is a peer-to-peer architecture, with no centralized server. No server has a unique distinguished role. Instead each user, “client”, running the application is basically one out of many servers connected to make up the session. Each user client runs a local instance of Jetty [24], a Java HTTP server that contains a Slide WebDAV [25] servlet. A Jetty filter stitches these separate servers/repositories together to create on each unit a unified, virtual shared space. WebDAV [26] is an HTTP extension that provides Internet-scale resource storage, retrieval and modification capability.

Each clients Slide WebDAV server manages the access to all shared files. A user’s shared files are stored on the user’s own Slide WebDAV server, so when a user leaves the session his/her files will disappear from the workspace (unless those files that were placed in the persistent sharing level). When a user leaves the session, his/her persistent files (those who placed in the inner center of the pie) are automatically moved on to another machine so that the users of the session can continue to use these files. This local storage of files on the client’s server is not directly seen by the users. Users only interact with the server using the Pie GUI. A session will continue to exist as long as there is at least one user still connected, until there is finally no users left and no more Impromptu clients running.

To keep track of constantly new arriving and departing peers on the network the Impromptu uses an implementation of the IETF Zeroconf protocols [21]. Zeroconf is a set of protocols that implement peer discovery, address allocation, name resolution, and related services over the TCP/IP protocols. By using Zeroconf user peers of Impromptu find each other automatically, with no previous configuration or user intervention necessary. When a user starts running the Impromptu application, the user will automatically find and join the other Impromptu peers on the same network. At the same time, the other users, already in the session, will see the newly arrived user.

For event service, the Impromptu uses Yet Another Configurable and Extensible Event Service (YANCEES) (Silva Filho et al, 2003 [22]), which allows the GUI components to monitor and register access events from both local and remote peers, as well as to present a unified view of the application session. Events are in this way used both to visualize activity and to ensure view consistency of each users interface. YANCEES thus maintains the pie view of each client in sync with the others by informing each client of events taking place on the other client’s Pie’s. This is made possible by a virtual event bus, connecting local and remote repositories with local and remote interface elements. Via application tailored plugins, YANCEES supports publish and subscription filters that allowed for the implementation of security and visualization policies. For example, a filter prevents local events such as the reading or writing to private files to be propagated to other peers.

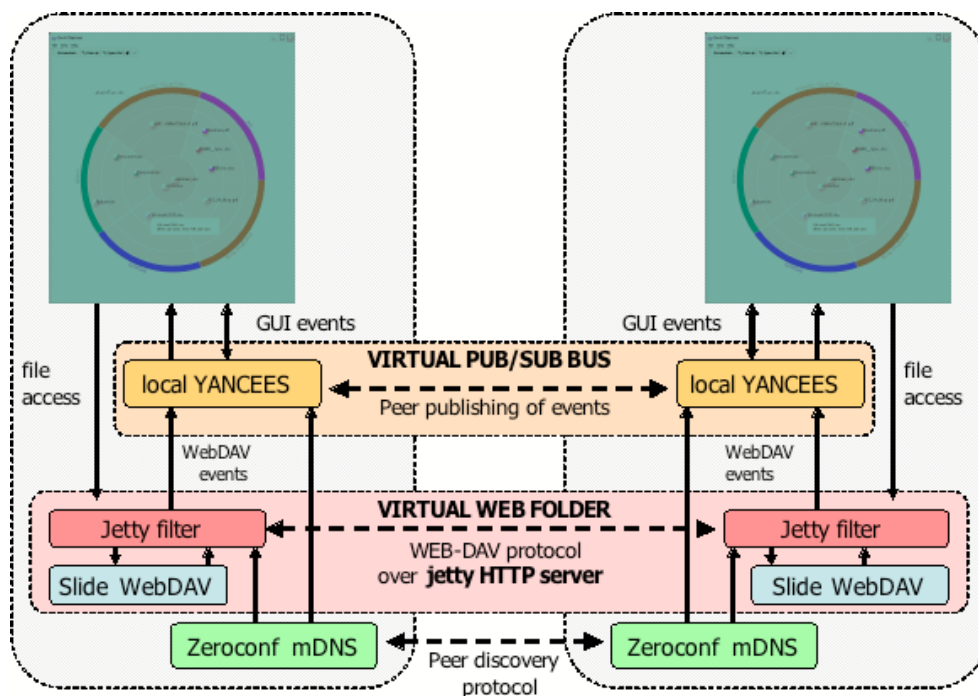


Figure 2: Impromptu Architecture

3.1.4 The graphical interface

Impromptu’s interface is based on the earlier discussed design principles 3.1.1. Visualization and direct manipulation give a user a clear representation of who is currently in the session, which files are shared and their respective sharing levels and the actions of the other users at a given moment. The interface allows for users to easily change the sharing level of any of their own files by simply moving the file around the workspace to a different area, representing a different sharing level. People can only control the accessibility of their own files.

Figure 1, of chapter 1.1.2, depicts the Impromptu workspace with files shared by the five different users. The circular “pie” is the area where users can place their files for sharing. Each user has a designated pie “slice” where that user can share his/her files. A user’s own slice is represented by a darker shaded slice. The user name of the participant will frame that user’s pie slice. Organization and orientation of the users within the pie is consistent in between all users. In other words all users will be seeing the same pie visualization with the same users in the same places.

A user imports and shares a file in the workspace by dragging and dropping it onto the circular pie. The file is then represented by a labeled dot, with the same unique color as the user has. A user can choose its color by menu options, but is given a default color at start, calculated out of the username. The color of the dot facilitates the easy recognition of which user a file belongs to. A white ring around a dot means that that file has not been opened by anyone yet.

By placing the files in the different sharing levels, the user decides how shared the file will be. A user can only place its files in its own pie “slice”, not any other users’ slices. Built on a round-table metaphor, the pie can be seen as a round table with the users sitting around it. If this was a real table with real people sitting around it, a user wanting to share a document with the others would push it as close to the center as possible for the others to reach it. If a user had some document she wanted to keep more private to herself, she would keep it close to herself at the edge of the table.

The sharing degree levels are designed after this metaphor: the closer the files are placed to the center of the pie the more access the users will have to them. The different sharing levels are:

- Outside the pie - this is the user’s private space, no other users can see the file there (it is not visible in their interface).
- Outer ring by the border of the pie - other users can only see the file name
- Second ring from the border - other users can read the file
- Third ring from the border - other users can read and modify the file
- Center of pie - persistent access, these files can be read and modified even if the owner of the file quits and leaves the Impromptu.

Whenever an activity occurs on a file, the dot will visualize this activity. If a file is opened or modified a ring around the dot will appear and blink for a short period. The color of the ring around the dot indicates the identity of the user who was the actor (i.e. who opened or modified the file). This blinking calls the attention to the new activity over the file. By doing so, it implements the concept of integration of action and configuration as well as of dynamic visualization of activity. For example, in Figur 1 we can see that the file named SWIRL_Spec.doc of user *jen* (upper right corner of the pie) has a red ring around it. This means that the file has been accessed by the red user *Xianghua Ding* (lower right corner).

The interface also includes menus with options such as color change, menu import and export of files, and increasing/decreasing of dot size. The overall ideas presented in this chapter is described in the paper “Two experiences designing for effective security” by Dourish, Redmiles et al (2005 [20]).

3.1.5 Prior user study

3.1.5.1 Study structure

In spring of 2005, a user trial was conducted to study the use of the initial version Impromptu. The goal of the study was not to test a specific hypothesis. Neither was it intended to gather quantitative data about any of the specific features or in short be a usability trial. The study was rather a wide-ranging, formative examination of the initial design to support future versions. The goal was to understand the effectiveness of Impromptu in broad terms and to identify areas for future attention.

The study was designed to be open-ended with a semi-natural scenario in mind, where the participants would conduct a version of a task they would be familiar with, closely resembling real life work tasks. It was designed as a high-level task so that the use of the Impromptu and its features could be observed while the participants were performing the task. In short, the

study was more to observe how the participants used, understood and interpreted the user interface rather than to answer any specific questions.

A total of 24 participants, all with higher computer knowledge, took part in the study which was made up of 8 different sessions. Each session had 3 users interacting with the Impromptu tool with one facilitator running the session. The participants had an observer each, watching them throughout the session and taking notes of their actions. At the end of each session these observers debriefed the participants individually about their experiences and asked them to provide three negative and three positive critiques concerning the application.

The task scenario was for participants to collaborate on a research budget as part of a grant application. Each participant was instructed to compile an individual budget and then to provide justification for each expense. The participants were all asked to import these files onto the workspace, with a free choice of the degree of sharing the file were to have. This led to different strategies of how to conduct the task, such as sharing all information from the start, hiding it until the very end so no other user could see their individual budget, etc. For more detailed description of the task scenario and the study see the paper “Seeing Further: Extending Visualization as a Basis for Usable Security” (Dourish et al, 2006 [1]).

3.1.5.2 Results

Broadly, the study supported the initial design principles. The integration of action and configuration was understood rather easily without problems. The task brought forward different styles of collaboration which showed that the users were thinking, about “security” and degrees of sharing, differently as the task progressed. Their behavior and actions during the task suggested that Impromptu supported context sensitive negotiation of sharing in between the users and the development of task specific norms. This relies on the fact that actions and configurations were mutually visible to all.

Several participants commented on the benefits of having a visual interface in contrast to the more traditionally text based view of security settings. Many participants brought forward complaints about the speed and performance of the tool, issues which has since then been addressed and improved.

Although Impromptu allowed for users to make sharing decisions in the context of a situation, the fact that other open applications would obscure the Impromptu application window, made a lot of activity go unnoticed for some users. Since the initial Impromptu design was very instantaneous, the only way for a user to be sure not to miss any events, was to never obscure the Impromptu window, which was impossible since the task had them working on other documents in other applications.

The dot representation of the files was well understood by the participants who also noticed when new files would appear. The ability to notice other participants’ changes to files and for the participants to respond to all threats situations suggested that Impromptu was successful as a mean of visualizing system activity.

The instantaneous visualization of only the immediate state was inadequate to fulfill all participants’ needs. This suggested a need for future extensions of visualizations of not only what is going on at the moment, but also what has also happened in the past. Such

visualization would help those users who had not viewed the Impromptu window over a period of time, while working in some other application, back on track with what had been going on in the session.

Another issue that arose concerned the insecurity of who could suddenly log in, join the session and have access to all shared files. The concern for this suggested future extensions to be able to distinguish familiar and unfamiliar participants.

A number of specific user interface concerns arose, and system performance was negatively critiqued. But beside those the study yielded three areas for further research and design. The first of these areas was the “real estate” problem of the screen, the fact that a maximized window would obscure the Impromptu workspace. Secondly, the current activities visualization is a very limited view for a user who wishes to re-examine events. Recovering recent context on returning attention to the Impromptu window would be a useful facility to keep users updated with that had occurred during their absence. Third, since any user can join the session at any time the study data illustrated the need to provide more information on new participants. There is a need for the user to be able to distinguish, as a new user enters the session, whether the user is familiar or unfamiliar to the local user.

3.2 My work

To get acquainted with the application and to familiarize myself around the system and its different features, I started out using and experimenting with the Impromptu interface by myself and later also with other colleagues of the Swirl research group. After having done the analysis of the system I studied the initial user study and its outcome.

While experimenting with the application, and knowing the design principles and aims behind it, I was noticing the limitation of only seeing the most recent events displayed. If I were to use this application for sharing files and to be aware of the activities so that I could update my configuration accordingly, I would very much lack the ability of doing so correctly, had I not been watching the workspace attentively without pause and been able to memorize all events. For most situations, this constant attention and memory aspect would not be possible, since the tool is itself designed for such settings as group collaborations on documents etc.

Having identified these concerns also in the three major future research and design attention areas of the user study, I decided to focus my project on the second and third issue being visualization of more distant past actions and characterization of the users as to who is familiar or unfamiliar. The first issue, dealing with the fact that other applications tend to obscure the user interface of Impromptu, is currently addressed in another Swirl project that studies the role of PDA’s as auxiliary awareness channels (Silva Filho R. S., Redmiles, D. F, 2006 [27]).

My work was to design, implement and incorporate visualizations of these new history and user characterization features, into the existing visual framework. I was thereby to extend the existing framework, without disrupting of any of the earlier features.

The focus was on extending the visualizations from the instantaneous display to demonstrate also the different relationships between now, then and possibly the future. The main idea was

in short to display patterns in general, of both the users in the sessions and of their activities. I wanted to answer such questions as how does pattern build up over time? How did you/ others act in the past? Which files have been accessed?

Early in the process I had brainstorming sessions with both my supervisor Paul Dourish and the Swirl research group members to come up with different features to visualize and design. I came up with a total of six different visualization features that would work towards our goal. They will be described in detail below. Out of these I chose the four strongest to implement. They were chosen because they accomplished in visualizing the history aspects we had decided to focus on. The time limit for the project also limited the number of features to implement. Following are the four design features I chose to work on and implement. They are followed by a brief overview of the features that only made it to the first design stage and were never further explored.

My four features are described and published in the paper “Seeing Further: Extending Visualizations as a Basis for Usable Security”[1] that will be presented at the 2006 Symposium On Usable Privacy and Security (SOUPS) conference, US.

3.2.1 Rings and Ripples

The first new feature that came most naturally to mind was to extend the rings around dots into “ripples”. These were created in response to the repeated observations from the user study about a wish to see more than just immediate activity.

In the original version of Impromptu a ring would blink around a dot when any read or write event was registered. After having finished blinking, the ring would become solid and stay on around the dot. However, a new event would cause a new ring blinking around the dot and there would be no way to tell which other event was just prior to the present one. With my feature of ripples, the old event ring would be pushed one step out, leaving the dot with two rings around it, allowing the users to distinguish both the most recent event and the second most recent. This could be repeated again and again at the arrival of new events, resulting in dots with multiple rings around it. If not limited, the ripples of one file would soon take over the whole workspace area and obscure all else. I limited the feature to a total of up to 3 extra concentric rings around a dot, in addition to the initial original innermost ring around the dot, to solve this screen real estate problem.

The algorithm behind the feature is that whenever a new event will occur on the file, the older event will be pushed one step out from the center. This means that the most recent event will always first be displayed as the innermost ring around the dot and as other events occur the “older” event will ripple towards the outside before disappearing altogether. It utilizes the metaphor of ripples in the water after you’ve thrown a rock into it. The ripples will start at the center but start rippling out and eventually disappear.

In Impromptu extended with ripples, when a new event occurs on a file, the innermost circle changes to represent the new event. The innermost ring will always represent the most recent event. The innermost circle will then ripple out to the second ring, since it is now the second most recent event. Subsequently, whenever a new event occurs, second ring will ripple out to the third ring, where the third ring will ripple out to the fourth ring. The fourth ring will ripple

out and disappear from view. This way the rings can at a maximum tell the four latest events on a file, and will not give us the full history. However, for this feature we were more interested in showing the more recent past. The more distant past would be visualized in a different feature (see chapter 3.2.2). The color of the ring would still indicate the actor of the event, meaning which user did a read or a write on the file.

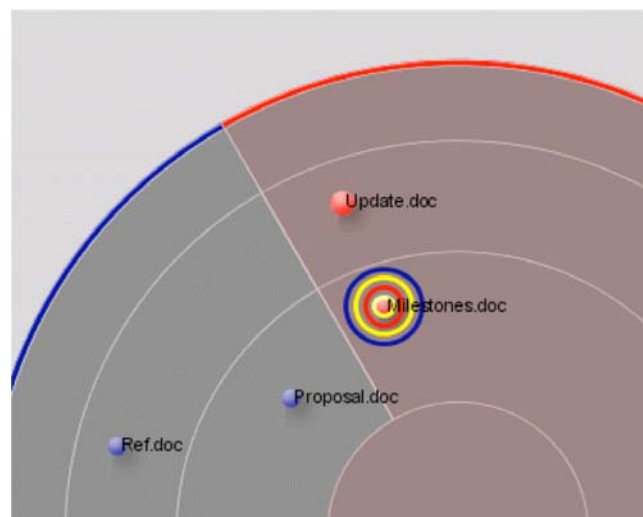
The ripples and the innermost ring differ in behavior and design in the following ways:

- The innermost ring is directly attached to the icon, whereas the three rings have a small separating space around them.
- The three additional rings also have a fade out time, meaning after some time has passed they will disappear from view.
- The innermost ring remains as a persistent indicator of the most recent activity and will not fade out.

A fadeout time of ripples was implemented to give the users a more realistic understanding of the most recent events on a file. A file may have the second most recent event on it being a read by some user hours ago. We argue that this does not count as recent history in time and it should therefore not be displayed. The innermost ring will however always stay on, even if the most recent event was an event that happened five hours ago.

This fadeout of an 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.

Figur 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.



Figur 3: History rings

The two events, read and write, both look the same. There is no way to tell if a ring means a read or a write event. We initially considered differentiating a read from a write. Several different ideas were explored as to how to separate them visually, such as using different textures (dotted, striped etc) or using intensity of color (light color for a read and darker color for a write). We did though choose not to differentiate the two events, since we are in fact more interested in visualizing the occurrence of activity rather than the exact nature of that action itself. Patterns would also be difficult to see in the small thin rings, and lighter vs. darker colors does not work well since some users may choose a very light color from start and make varying it hard.

3.2.2 History pie

I designed the feature “history pie” to provide a complete temporal description of all activities on a file over the full duration of a session. On a mouse over of the dot, the history pie of that specific file will become visible in the lower right corner of the interface.

3.2.2.1 Initial designs

With the ring and ripples we had accomplished to extend the interface to show the most recent events at that moment. But since the rings and ripples only displays up to four of the last recent activities on a file, there was a need for a more overall feature to display the full history of file.

The early thoughts were to provide a list of all activity to be visualized on a mouse over on the dot. The list would consist of:

READ
WRITE
READ
.....

The list would have a scrollbar to the right for scrolling down to more distant history. The first item in the list would be the most recent event and so forth. Each READ, WRITE would also have the color behind it, displaying the actor of the event (who had read or written to the file). This list idea soon proved to be a bad idea since it required a lot of scrolling and since all information would not be visible at a glance.

The next design idea was to have a mini-pie in lower right corner to be made visual for a mouse over on a dot. This smaller pie would have the same orientation, organization and color assignment of the users as the central big pie display. It would have the same pie-slices positioned in the same orientation as the big pie. READ and WRITE text lines were to be placed inside the pie slice of a user in this smaller pie to indicate that user’s activity over the specific file. Although this solution was a bit clearer, it had no way of showing which event was the most recent etc.

Building on the same smaller pie idea, it would visualize activities as small dots placed inside the pie slice of a user in this smaller pie to indicate that user’s activity over the specific file. A dot closer to the center would be a more recent read or write on the file in question and a dot

at the edge of the pie would be a more distant activity. To distinguish between read and write actions an unfilled dot would represent a read event and a filled dot a write. Although, the dots in the small pie would indicate timing-order within a slice, by one user, it would be hard to exactly see which user had done the first event since the dots would be centered within in each users slice. Also we ran into another real estate issue. Since the smaller pie was to fit in lower right corner, it would take on 10-20% the size of the large pie. If one user had done a lot of activity on one specific pie, its pie slice in the smaller pie would soon fill up with dots to make the details indistinguishable.

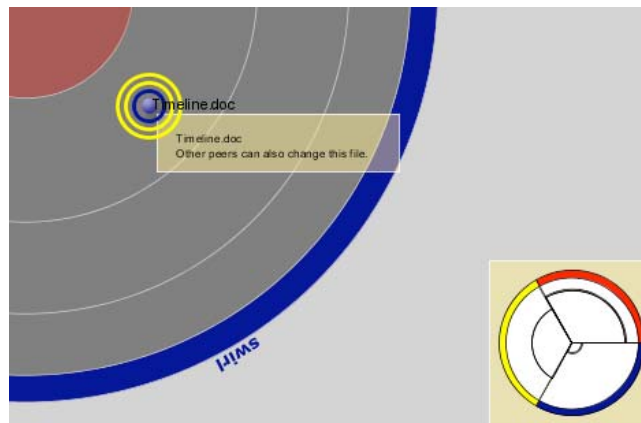
3.2.2.2 Final version

At the fourth iteration, I reached my final design version. Still building on the use of a smaller pie idea, arcs in each user slice now represents activity. An arc in a participants slice represents an action in the past by that participant on the file. Building on the same idea as for the rings, the arcs closer to the center represents more recent events whereas arcs at the edge of the pie are events of a more distant past. It can be compared to the rings that indicate the growth pattern in a tree, the more recent the closer to the center the arcs are. The radius of the history pie is set to be a timeline representing the whole session time. It can easily be changed to represent a more recent, shorter time period.

I designed the feature “history pie” to provide a complete temporal description of all activities on a file over the full duration of a session. On a mouse over on the dot, the history pie of that specific file will become visible in the lower right corner of the interface. In this context, I define history of a file as all activity that has happened on a file since I, the user, joined the session. The history pie is supposed to be giving all information about the activities on a file, which a user might have missed when he/she was not staring at the Impromptu workspace. This means that any events that happened on the file earlier on in the session, before the local user “I” entered, are not considered history for that local user and will not be shown in history visualizations.

A read or a write event are both visually alike, a simple arc depicts them both. We had as an option to make read arcs thinner and write arcs thicker but since the history pie to start off with is very small, it would be hard do distinguish between one thick line and two thin lines close together.

Figur 4 portrays a mouse-over of the file Timeline.doc, a file owned by the blue user “swirl”. 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.



Figur 4: History pie

I also designed the option to show text lines of the six most recent events in a short list, positioned right above the history pie in the workspace. The most recent event is highest up (first) in the list. These lines will display textual information about the type of event, actor of the event and the time the event occurred. This extension of the feature is not displayed in default mode but is rather there as an extra aid in further development of events, for debugging and for those users that are interested in the exact time an event occurred.

3.2.3 Activity wear

In a security setting, it is always interesting to be alerted if one person is a source of more activity relative to the other users. The interpretation could be either good or bad, depending on the setting and the task performed. We do not imply that little or much activity is either bad or good; we simply want to show the fact of activity of users.

With the rings and ripples and the history pie we can visually get all history of all users on one document. This was complemented with a visualization of each user's activity on all files, which was also a response to the need of wanting to know more of what's going on that was identified in the user study.

The feature for this display is based on the idea of edit wear and read wear (discussed in chapter 2.2.2), introduced by Hill et al (1992 [14]) that repeated actions will result in a digital wear on the artifacts that display them. In my case, I wanted to display the accumulated activity of a user relevant to the others.

Several different visualizations of how to display accumulated history of a user were explored. One early design was to have grey lines growing up behind the user name at the edge of the pie slice. This would yield a shadow building up behind the user name to display times when that particular user was very active. Another possible way to display an active user was to make its user color darker and darker. But once again, we stumble on the color problem that every user is allowed to choose its own color at start of the session and a color can be chosen light or dark. Printing the user name in bolder and bolder letters to display more activity was neither a good solution. It would be too difficult to distinguish when dealing with really short names vs. long names. Also the range of boldness of letters in SVG [4] is not large enough to really depict differences. Showing activity by enlarging the user

name framing the pie slice was yet another alternative we discussed but dismissed since we are measuring the activity relative to the others and this would yield some very small font sizes when others grew large.

The final version was built on the initial idea of having activity building up. But instead of building up a shadow behind the user name, I choose to let the border of the pie display how active a user is. The width of the border displays the activity of that participant over the last five minutes. Opening a file and writing to a file are the two events that are considered activity. The activity of a participant is always measured relative to the activity of other participants. This means that an inactively displayed user could in fact have been quite active but relative to the total activity in the session he/she is considered relatively inactive. If all participants have been equally active they will have equally wide edges. A thin edge displays a relatively inactive participant and a thick edge a relatively active participant. The edges will grow and shrink continuously during the session. A thick edge can be seen as is often the cartoon effect of motion, a drawing of a person in motion has lines around it, and a thick edge represents a lot of motion.

The activity wear can be set either to represent the activity of the user over the whole session or during a more recent time period. I choose the later alternative for my study, since it displays more recent activity measures. This means that when some time has passed the activity wear will shrink down to a minimum again and the border display gets thinner, when a user is not as active anymore. By eyeing the border, a user can at a glance get an understanding of which user is the most active at the moment. Here again, we do not define neither an active nor an inactive user to be any better or worse. We simply want to display the accumulation of activity in a user. The parameter was set to show the activity over the last five minutes but this can easily be changed to any other time interval. This activity display is not meant to show exactly the number of actions by the user but rather its amount of accumulated activity relative to the other users.

Figur 5 displays the full view of a history visualizations Impromptu workspace. The width of the users pie slices displays that user *Kari* (lower right) has been the most active user during the most recent five minutes and user *Lina* (upper right) the most relatively inactive.

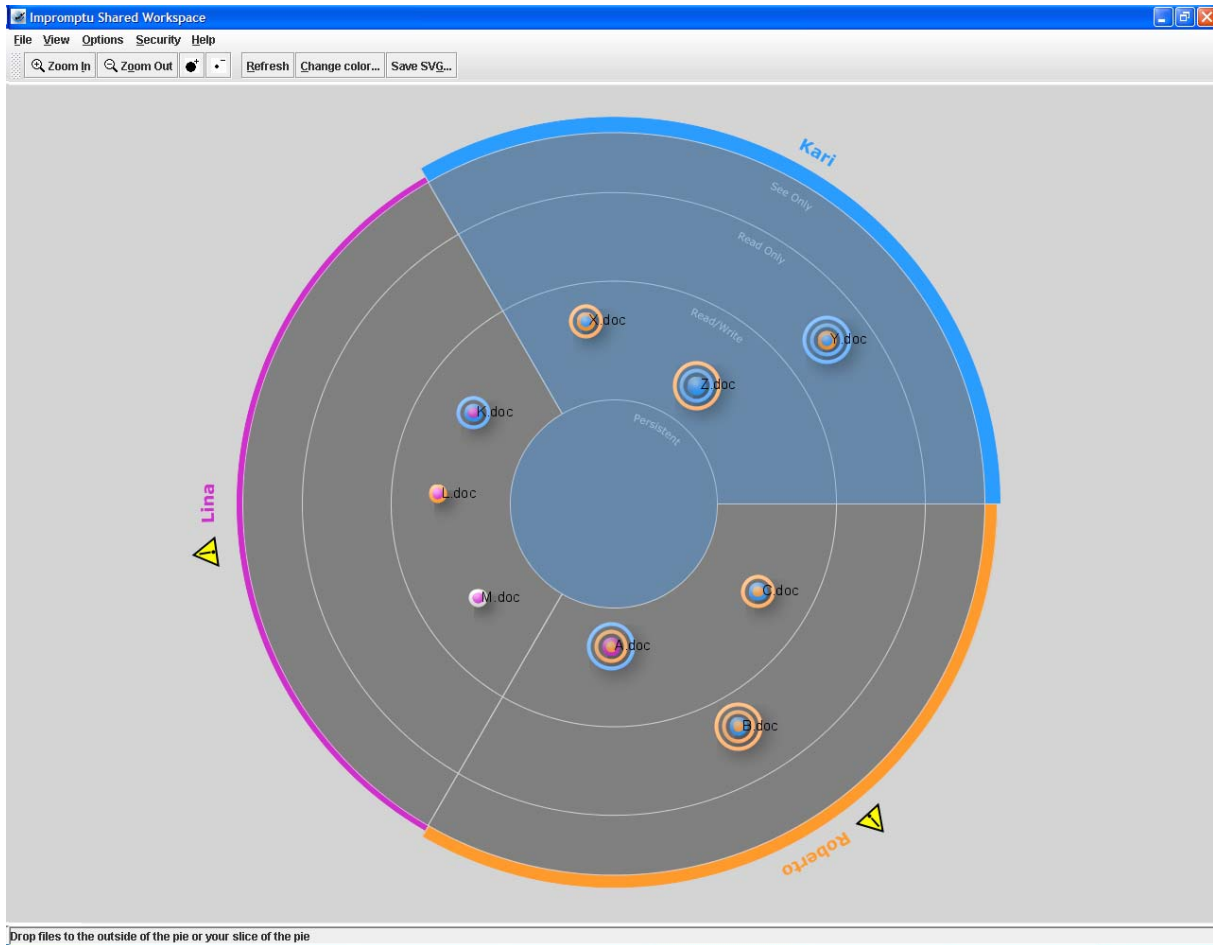


Figure 5: Impromptu with history visualizations

3.2.4 User characterization

So far, the features that I have described have all dealt with history within a session. Since many people work with the same people over a long period of time and might be engaged in multiple sessions, the history of activities over time provides another useful source of information. To broaden the perspective I designed a feature to take into account the history of users, which would be a connection between earlier sessions and the present session. This user characterization feature was also in response to the user study outcome that users wanted to know more about unknown users suddenly entering the network.

I wanted, in general, to display whether a system configuration that a user would encounter at any time, both at startup of session and during the session, was what the user might expect. That is to say, I wanted to display familiar and unfamiliar users to the local user in question. If a user can see via visualization that all other users are users known to him/her that could aid in his/her assessment of the security situation. Whereas a new user joining the session displayed as unfamiliar and unknown to the local user, the local user could make use of this information as for setting possibly new file sharing degrees of security etc.

To distinguish a user as either known or unknown the local user client will compare the pair of the user's physical address (MAC address) and the user name, to a list of saved MAC

address/user name pairs in that clients repository. If there is a match, the pair has been seen before; the new user will be displayed as familiar to the local user. If not, the user will be displayed as unknown.

Different design ideas as to how to display known vs. unknown users were explored. One early idea was to print the user name at the edge of the pie slice in different colors, such as red for unknown user and green for known user. This was quickly discarded, since we do not want to build too much of our designs on just colors. Another idea was to print the name in bold or not bold, where bold would represent an unknown user. This proved not to be very visible when implemented and had to be discarded. The idea of having an alert icon represent an unknown user proved to be the most clear and easily understood design. Putting a star * after a username was one alternative to show an unknown user but a more intuitively understandable symbol was to add a yellow warning triangle before the user name. The warning triangle now displays that a user is unknown to the local user.

In Figur 5 the screenshot of the workspace has been taken from the screen seen by user *Kari*. This workspace displays that both of the other users *Lina* and *Roberto* are unknown users to Kari, meaning she has never encountered them in an earlier session (in regards to their user name and physical address pairs).

3.2.5 Designs not implemented

There were also more design options along the way that were discarded mainly due to time limitations. One of them was the display of the history of earlier locations of a file in the workspace. A dotted line leading from the dot would on a mouse over show the actual paths the file had been moved along in-between different security levels. This would inform the users of a file's movements through the workspace, showing if for example if it had first been in a low security level but later been moved to a higher security level. Another version of the same idea, was to have all different security levels that the file had resided in turn darker when a mouse over on the dot would occur. But both of these ideas would probably prove to cause too much flickering on the main workspace when just moving the cursor over the screen and prove to be too interrupting for the head task of the application. A slight improvement to this feature would be to only show the last move of the file, with a line or arrow drawn from the dot to the previous security layer it had been located in. But this feature was not strongly enforced by the outcome of the user study and was never implemented.

Another feature not implemented was the display of history of shared files, in this case answering questions like: have I shared this file before in the Impromptu application? The idea for the feature came from the folder metaphor presented by Dourish and DiGioia (2005 [15]), where an often shared folder would be visualized as a more open folder and a never opened folder depicted as a closed folder. Neither this feature was implemented mostly due to time limitation. but could remain as a future possible extension of the application (see chapter 8, future work).

4. Implementation

4.1 Programming language and tools

Following the code basis of the Impromptu application, all code was written in Java [3] and based on Scalable Vector Graphics (SVG) [4] for the dynamic visualizations of the pie and its components. SVG is a W3C [16] recommendation, based on XML [5] grammar that enables rich 2D graphics including features such as transparency, filter effects (shadows, colors), script etc. SVG was specifically chosen for the Impromptu GUI because it's scalability. Since the graphics are vector based they will not lose any of their quality if zoomed or resized, which is desirable when considering future use of the application on different sized screens. The Batik toolkit [17] was used to enable generation, parsing, viewing or conversion of SVG contents using Java technology.

All code was developed on the Eclipse platform [18]. Eclipse is a universal tool platform, an open source extensible Integrated Development Environment (IDE). It is a multiple language IDE that assists programmers in developing software.

4.2 Implementation of features

Throughout the first phase of the implementation, I investigated and implemented different versions of visualizing the same features, to try them out in the actual setting. As already described in chapter 3.2, different versions of how to display the feature were explored before the final version was decided upon. In my code, the fade out and update values can all easily be changed to different values. This enables longer fade outs (or practically no fade out) on rings and the activity wear can also be set to reflect the whole session instead of the five minute window I choose for my study. Also the update of the history pie, while on a constant mouse over, can be changed to update more frequently or oppositely not update at all. Another extension to the interface was that I added menu options to turn on and off the four new features that I had added to the application. The availability of switching on and off these features proved very useful for the study and also adds to the sense of control over the configuration from the user's point of view.

4.2.1 Rings and ripples

When implementing the extension of rings and ripples around a dot, I based my code on the already existing one-ring template. I extended it with three extra rings and timestamps were added to every action, so that I could compare when an action had occurred and its current place along the time line. Every ring visualized was also extended with a fade out timer set to go off after two minutes.

4.2.2 History pie

A first pre stage to the actual history pie was to print all the actions in a list over in the bottom right corner. A total of six events, the six most recent, would be showed in that list. Each line of an action consists of the actor, action and the actions timestamp. All the actor/action/timestamp triplets of a dot are saved into a vector and this is where the history pie retrieves the full history of a file.

Before the implementation phase we had talked about having a display of the full history list of a file as an option in a menu, or as an option when the user did a right-click on the dot. This would however obscure the whole window and for a normal user, this exact history is not needed. Due to time constraints this option was never implemented, but the last six actions on a file can be chosen for display right above the small history pie. It is however not an option in a menu, but a parameter in the code that has to be set to either true or false. Mostly, these six text lines are intended to be used for debugging and testing of code.

The bigger, original pie, served as a template for creating the smaller history pie, but only as far as having the same orientation and positioning of users and their colors. The big pie is divided into different security levels whereas the history pie has an empty inner space that is eventually filled with arcs representing actions on the file. The background to the history pie is a square with the same soft background color as is already used for the file description window that is also showed on a mouse over and is situated right by the dot. I wanted to keep it as consistent as possible so it would be more intuitive for the user that these two were both displayed as a result of a mouse over on the dot.

I also implemented a timer to update the history pie every 20 seconds for those cases when the user would leave the mouse on the dot and the history pie would be on constant display. This timer would update the history pie display every 20 seconds until the user would remove the mouse from the dot.

4.2.3 User characterization

For this feature, I implemented many different initial versions before we agreed on the final version, being the alert icon. For this user characterization feature several different visualizations were examined such as printing the user name in red/green color or bold/not bold, at edge of pie slice. Finally the alert icon was chosen. The warning triangle, with yellow background, was drawn using SVG [4] and required some rather complicated calculations before being able to place it correctly in regards to its orientation around the pie. User names, being texts, had the advantage of already being placed on an SVG path for its placement around the pie. But this was not possible to do for SVG drawn triangle, thereby the need for specific new calculations.

To distinguish between known and unknown user I compare the physical address, the MAC address of a user, in conjunction with the user's name in the session to a table of already seen MAC address/user name pairs. If I have been in a session with this user named X before but in that session user X was connected from another physical address than the one presently, I'm considering user X as unknown. This could be a case of one person trying to impersonate the real user X, so that other users may hold off their guard, feel more secure and share more

files. But displaying X as unknown helps the user see that this physical address/username has never been encountered in an earlier session.

To get the MAC addresses of the users I employed the Address Resolution Protocol (ARP) [19]. By examining the ARP cache, I can see which physical address is belonging to which user. Whenever a new user would join and at startup of session (i.e. whenever the client of a user would add a new user) each client would run a check on the new user by running either of two arp commands depending on the type of the local users computer. For those clients running the session on a Mac computer, they would use the “/usr/sbin/arp “command followed by the new user’s Internet Protocol address (IP) for. For PC clients would run the “arp -a “command followed by the IP address of the new user. To distinguish between which command to run I checked the properties of the users system to find out if it was of type Mac or PC. The IP address of the new user was retrieved from the Service Information belonging to that new peer.

The warning triangle, representing an unknown user, can easily be exchanged by any other alert icon, and the code to calculate its position and orientation could be reused for placement of other additional, possibly future icons.

4.2.4 Activity wear

The final design chosen to display activity wear was the growing and shrinking of the border of the pie slice of a user. I defined a minimum and a maximum border value and in between these two the width would reflect the recent accumulated activity of that user. I chose the to display the activity accumulated over the last five minutes in the session, although this is a parameter very easy to change to have it reflect on the whole session or any other time interval. The parameter chosen for this feature has to be matched to the pace of the work the people using it are doing. For our study tasks and for intended work-setting for the application, the parameter was set to five minutes since the work is rather fast paced. Since the activity of a user is measured relative to the other users the calculation is always taking the activity count of one user and comparing it to the activity of the whole session for the specified time period. As activity, I count read and write events on files, namely the same events that give rise to rings around dots and that are depicted as arcs in the history pie. The border radius of all the different users, displaying the accumulated activity, will be updated for every new activity that occurs by any user in the session.

4.3 Problems encountered

All through the implementation phase I was continuously running testing sessions, whenever I had a new chunk of code, with other members of the Swirl research group. The testing sessions were very valuable for making sure my code worked for multiple user sessions and for debugging it. We would also find errors or things not working correct with the original code base, some of which were already known from the year before and some which had never been caught. With my new features and the fact that I was logging all events that occurred, strange behavior of the under laying structure was brought up to the surface. These

were system behaviors that had probably always been there but had never been found, due to the limited early visualizations.

In this chapter I will present in two different sections the problems encountered during the implementation. The first section will discuss problems related to the original platform. The second section will discuss the problems encountered that were closely affecting or part of my designs.

4.3.1 Original code base

To start with, the original platform was not very stable, when having 3 or more users in a session. The problem was in large part due to incorrect identification and registration in among the peers of the network. Impromptu utilizes the IETF Zeroconf protocols [21] to identify and manage the peers that are continuously arriving and departing from the network. This was wrongly configured and yielded events to be broadcasted as belonging to wrong users, wrong IP addresses were being sent around. This bug was fixed rather early on.

Other bugs were also attended to in parallel to my implementation. Besides the bugs, some other updates were also made to the original code base. For example: the big pie was extended to now display short descriptions of the security level of each section of the pie. These descriptions are displayed in the user's own pie slice, where he/she will be sharing files. In Fig 6 these sharing level descriptions are displayed within the pie slice of the local user *Kari*, ranging from the center's persistency to the outer sharing levels see only of files.



Figure 6: Descriptions of the sharing levels

There were also the constant mounting problem of the I drive that would hold the files of the local user. This procedure was improved and batch files to mount it manually by hand were written. Even though we got a better rate of successful mounts this was still a problem up until the end of my project. It seemed to depend in part on the specific machine the application would run on.

4.3.2 My designs

Early on we also discovered that double events were being sent out for certain file types. A read and write on a Word document would be sometimes be seen as two separate reads, because of the under laying structure of Microsoft Office (that was the default application to open the word documents). These duplicated events would occur on many other file types as well. To solve this, a duplicate events filter was installed, to filter the second read or write during a short period after the first had arrived. This filtering period had to be extended up to 15 seconds to catch all duplicated events. This would however bear the limitation that if two different writes would happen within 15 seconds of each other, one of them would be filtered away. In spite of this, we decided the tradeoff was small enough to get a more correct view of what was going on in the session.

When preparing for the user study, careful studies of exactly all events showed that an import of a file would give rise to both a read and a write event, even though this was not a read and write from our perspective. This was taken care of by adding an initial filtering when files were imported.

What proved to be the biggest challenge for debugging my code was the difference in system time on the different laptops we were running the sessions on. In many cases I wrongly spent time analyzing my code for mistakes, when the cause of the weird visualization was the difference in timestamps put on actions by the different user machines. If the clocks were off only a few seconds it could cause faulty visualizations, especially when drawing the arcs in the history pie. For example if user A's PC is 20 seconds ahead of the time of user B's, an action with a timestamp of the time the action occurred at user A would be received by the client of user B who would then compare this time to its own system time with the result; the event has not even happened yet! It is from the future! We solved this by installing an atomic clock synchronization program on all laptops and later also on the PC's used in the user study. I also implemented an extra check, when calculating where to draw the arcs by the time line in the history pie, to check if the action is from a time in the future. If so, the time of the action is seen temporarily as the current present time until the local system clock catches up and the actual action time becomes a past time.

4.4 Cognitive walkthrough

In the last part of Implementation phase we did a cognitive walkthrough of the features with the usability expert of the Swirl research group. A cognitive walkthrough is a usability evaluation technique modeled after the software engineering practice of walkthroughs (Dumas, Redish, 1999 [28], p 68-69). In the cognitive walkthrough we "walked through" the use of the application and its new features, step by step, as if we were novice users not acquainted to the system. All the time we were trying to mentally simulate what we thought a novice users understanding would be for each of the features, looking for errors and potential areas of confusion. The goal of the walkthrough was to identify the strengths and weaknesses of the features and the application as a whole.

The results of the walkthrough addressed issues to be changed, improved and updated both concerning my new features and concerning the initial Impromptu test bed.

4.4.1 Issues concerning original code base

Issues concerning the original code base were:

1. The font size of user name, file description in popup window etc all had to be enlarged to be more visible. This is of special importance when the application is run on a small screen PC. Font size was increased.
2. The ability for the user to choose his/her user name when entering the session. A new command was made available so that when starting up the application via the command prompt, the user can now choose wished user name for the session.
3. It is possible to place one dot on top of another when importing a file into the workspace. This was however decided by the research group to not be tended to, since many developers argued that our aim is to visualize the users actions, it would be more visually correct to actually display where exactly the user choose to place the file, instead of having it magically jump to the left of some other file.
4. The display of the user name at the edge of the pie slice was not centered but was put had a hard coded placement. By using an SVG method, the user name is now always displayed centered on the edge of the slice.

4.4.2 Issues concerning my features

The issues concerning my features were:

1. At the time of the walkthrough, the innermost ring around a dot would also fade after some time. This was seen as a problem as it would be no way of telling from rings which were untouched “virgin” files and which files had had activity on them but which had faded out. The history pie would still show the complete and full history of the file but our usability expert preferred that this would also be displayed in the rings features. As a result of this, the fade out on the innermost ring was taken out so that the inner ring will stay on until it gets pushed/rippled out to the second ring.
2. The space between the rings around a dot was not seen as part of the dot. This caused the history pie to flicker when moving the mouse over the dot with its rings. To fix this I added an under laying circle that would shrink and grow in accordance with the fade out and ripples of the rings, to make the whole dot with rings seen as one solid object.
3. For the activity wear feature, there was no gap between the user name at the edge of the pie slice and the border, when at maximum width. This was addressed so that a there is always some space between a maximum wide border and the user name.
4. The box surrounding the history pie was wrongly drawn on top of the history pie. This caused the colors of the pie slice borders’ of the history pie to be slightly dimmed and blurred. This was attended to so that the pie is now drawn on top of the box.

5. User Study

The first step towards conducting the user study was to get the study approved by the Institutional Review Board (IRB) [29] at UCI. This application required a very detailed narrative of all the aspects of the study to be written, from fellow researchers to the recruitment process of participants. The IRB application was very helpful in the aspect that it made me think through various study aspects at a very early stage. All associated with and working on the study also had to complete a human research tutorial.

5.1 Purpose

To evaluate my features, the new extensions of the application, I conducted a user study at the end of the project. I designed it to specifically target the use of three out of the four features that I had extended the application with. Those were: the rings/ripples, the history pie and the activity wear. The user characterization was not evaluated in this study. The reason for this was that all of the three other features reflect history within one session and was suited to be evaluated in a laboratory single session setting. The user characterization feature was on the other hand reflecting history of users over multiple sessions. A fair evaluation of this session-after-session related feature would only be possible in a naturalistic study setting when studying people using the application over a longer period of time and for multiple sessions.

With the study I wanted to answer questions such as: can the users make sense of what the history visualizations mean, did the added features help them when carrying out their study tasks, did the users like the history visualizations? I wanted to see if my added features helped them to better understand what was going on when using the application and if they were able to see and understand past activity.

5.2 Study design

As described in chapter 5.1, I designed the study to focus on the history features. But since the features are in fact part of a large application it is inevitable to avoid getting also some feedback and comments from the participants regarding the original Impromptu application issues. I designed the tasks and the questionnaires to target my history features, but was prepared that the users would give feedback and opinions of the whole system, not just my features. After the study was concluded, the majority of the feedback was though very much focused on my features, touching on some aspects of the general interface.

The user study was conducted over a two week period, situated in a Lab on UCI campus. A total of six sessions were conducted, with each session lasting approximately 80 minutes. Each session was composed by two study subjects, being recruited participants, and one “inside-man”, our confederate, being a member of the Swirl research group. This would make up a total of three users connected and interacting in each session. The sessions was run by the Test Administrator, me, with the aid of two test observers, which roles were to note down the actions and behavior of the participants and later do the brief interview (see chapter 5.6).

In user studies, the test administrator is the person in charge of the actual test and thereby has primary responsibility for all that occurs (Rubin, 1994 [30] p 62).

A session consisted of three short game inspired tasks (described in detail in chapter 5.3), each followed by a brief questionnaire asking questions about what happened in the sessions and reminding the users to use the Impromptu interface at any time to aid in answering the questions. Each session was opened with a brief Pretest Questionnaire of the background of the users and was closed by a Posttest Questionnaire followed by a one-to-one interview of each of the two participants. There was also a training task performed to make the participants familiar to the application and to the specific history visualizations.

The structure of each session was the following:

1. Welcome and verbal consent from each participant after reading the Study Information Sheet (see Appendix 1). Each user would choose their user name for the session.
2. Filling out of Pretest Questionnaire that asked the participants about their technological background and basic demographics (see Appendix 2).
3. Performing a brief training task to make the participants acquainted to the application and the specific use of the history features (see Appendix 3).
4. Introduction and scenario description of Task 1: Cards (see Appendix 4).
5. Performing Task 1: Cards_1
6. Filling out a Cards_1 Post task Questionnaire asking questions about the task just performed (see Appendix 6).
7. Performing Task 2: Cards_2
8. Filling out a Cards_2 Post task Questionnaire asking questions about the task just performed (see Appendix 6).
9. Introduction and scenario description of Task 3: Lists (see Appendix 5).
10. Performing Task 3: Lists
11. Filling out a Lists Post task Questionnaire asking questions about the task just performed (see Appendix 7).
12. Filling out a Posttest Questionnaire dealing with the various features of the application (see Appendix 8).
13. Final brief interview of each participant separately, asking about their general attitudes and concerns (see Appendix 9).

In the following chapters 5.3 and 5.4, I will describe all study documents, including the task designs, their scenarios, the purpose and design of the post task questionnaires and present the structure of the final interview.

5.3 Task design

The tasks were designed to make use of the history visualizations that I wanted to evaluate for the study, namely: the rings/ripples, the history pie and the activity wear. As explained in chapter 5.1 the user characterization feature was not evaluated in this study.

In the first two tasks the history was a critical issue while performing the task, whereas in the third task the history was more important and came into focus when answering the post task questionnaire.

5.3.1 Cards task

Designed to test the usefulness of the rings, the first and second cards tasks were basically one and the same task, but each displaying a slightly different interface to the user. For session 1-3 the first cards task would not have any rings/ripples displayed in the interface, where the second cards task would have the rings/ripples displayed. For session 4-6 the order was switched so that in the first cards task the rings were visible, but in the second cards task they were not. This was done to enable comparison of the accuracy of the users' answers, between the task when the user had had the extra rings as help or when only the inner ring had been display, to questions about what happened in the task of the post task questionnaire.

The participants did not know ahead of time that they were going to perform the same task twice. The switched order of without-with to with-without the display of rings in between sessions 1-3 and 4-6 was implemented to fairly have both versions being the second task equal times. This ensured that neither version would have the advantage of always being run as the second task, by which task the users already had an understanding of the task and knew to be more watchful to be able to answer the post task questionnaire later on. The history pie was not enabled for any of the two cards task, since I specifically wanted these tasks to answer questions about the usefulness of the rings vs. without rings.

The cards task scenario was a simple cards game. The participants were to look at playing cards, add their values and try to receive the highest sum. An "Ace" was considered 1 point, "Two" was 2 points, a "Three" 3 points etc. A Jack, Queen and King were each worth 10 points. Each user were given 5 different TIF images residing in a folder named either "Cards_1" or "Cards_2" depending on the task, that resided in the computer's desktop area. Each image showed a playing card of a specific value.

When opening the folder at start of the task, the participants would see their own five cards and their values. The participants would then share 3-5 cards of their choice in the workspace. When directed to go ahead, the participants would open 3 cards of their choice, either belonging to themselves or to somebody else. These 3 cards were then added up and the participant with the largest sum won the game.

The fact that each participant would know the value of his/her own cards were specifically incorporated into the task design since this would make a users opening of cards less random. Participants could, for example realize that if another participant chose to open on of his/her own cards that was probably be a high card. By following this strategy I could say for certain that they had used the ring/rings. For detailed cards task scenario description see Appendix 4. The participants were also informed, before the task that they would later be asked to answer a set of questions concerning what went on during the task.

5.3.2 Lists task

The third task was the lists task, for which the task motivation was to win a game of adding items to lists. This task was designed for the users to generate a lot of history which I could afterwards ask questions about. During this task and while answering the post task questions,

all history features were displayed including the history pie (which was not displayed during the first two task). It was never the intention of this task design that the users were to use and look at the history features during the task, but rather to have the users produce a lot of history to afterwards answer questions about. The questions were designed for users not to be able to deduct the answer to a question from watching only the rings, since these would have faded. Rather, to be sure of an answer, the user would have to consult a file's history pie to get the full and complete history.

The lists task scenario a game where users would open and edit files containing lists of different categories. The users were told that both the number of items added and the number of different lists edited would count towards a better ranking in the game. There was however no final calculation as to which user really won the game, since that was not a concern for the study. Each user would share 5 files, already provided to them, residing in a folder named "Lists" on the desktop area. They would import the files and place them in such security levels so that other users could open and edit them.

The users were only allowed to have one file open at a time, and to hit save and thereafter close the file before opening a new one. All lists were Word document being opened by the Microsoft Office Word application. This default application was employed to ensure no two users could edit the same document simultaneously. As soon as a user opened any file, the file was "locked" for the other users. Any other user that tried to open that file would get a message that the file was read-only at that moment.

The users had five minutes to complete this task. For detailed lists task scenario description see Appendix 5.

5.4 Study documents

In this chapter I will describe the post task questionnaires and the final debriefing interview in detail. In addition to those documents, a whole set of other sheets and documents had to be prepared to make up the full study.

For user studies, consistency is the underlying theme to ensure experimental rigor. To have valid results it is important that each participant will all experience the same material in the same way under the same conditions (Rubin, 1994 [30], p 93-94). To ensure that each session would be identical to the next, I composed a session script for me to follow as a test administrator. This script described into the smallest detail how the session was to be run with written scripted notes for what to say at what time. It also included the exact timing of when the test observers should perform certain actions (see chapter 5.6), when to enable/disable certain history features, which documents should be handed to the participants at which time, etc. In addition to the test administrator script I also wrote scripts for both the test observers and the confederate, for them to always know where in the program of the session we were.

To ensure that none of the details were forgotten during a session, I also wrote a checklist to complement the test administrator script. The checklist ensures that none of the important points are missed during the session (Rubin, 1994 [30], p 224-225).

I had to define debriefing guidelines as to how the test observers should conduct the final interview and what they were supposed to note down during the sessions tasks. I composed a recruitment sheet that was handed out to under graduate students describing the study in short. For the study, the study information sheet that was given to the participants at the beginning of the session was very important. This sheet informed the participants of what the study was to include, where they could turn if they had any questions after the study was completed, etc. The participants were asked to verbally agree to go on with the study after they had read through the study information sheet (see Appendix 1).

The pretest questionnaire was designed to get the background information of the participants, such as their earlier technical experience, years in computer related studies, programming language experience etc. The pretest questionnaire is there to help the analysis of the study data afterwards by providing the background of the participants (Rubin, 1994 [30], p 151-155).

Before the tasks got started the participants were performing a training task to introduce them to the application and the features. The participants were given short written information presenting the different sharing levels of the pie and their meanings. After reading this, the participants went on to perform the training task, which was following the training task script, headed by the test administrator. In this task the users got to share files in the workspace, perform different actions and was asked to watch the features appearing or changing due to the activity. This training task had the participants looking at all three features under evaluation, which were each described in detail (Appendix 3).

5.4.1 Post task questionnaires

To catch some reactions of usage and to see how the participants were able to answer questions about the activities of the task just performed, I created task-tailored post task questionnaires. A post task questionnaire should be rather short, with five to six questions at the most where the participants also have some room to make comments (Dumas, Redish, 1999 [28], p 210-211). The questionnaires asked the users about what had been happening during the last task. They were designed to target the history of the task and to see whether or not the users made use of the history visualizations. By asking specific questions I was able to later do comparison and get exact numbers for the outcome of the two with and without versions for both the rings and the history pie.

The cards post task questionnaire asked the participants questions such as which card was opened by the most participants, which cards were opened by participant *Roberto* (always being the confederate, see chapter 5.6), did you notice if any card was opened by more than one person, how did you conclude that answers, etc. After having performed cards task 2 the participants were given the exact same questionnaire as they were after cards task 1. This way I could do direct comparison in between the two versions. When filling out the questionnaire the participants were reminded that they could use the Impromptu interface at any time to help them with the answers. For cards post task questionnaire see appendix 6.

The lists post task questionnaire asked the participants questions such as which files were not opened, who last accessed file X, who was the last to touch file Y, which participants have touched file Z, etc. By asking such task specific questions it was easy to make direct

comparison of the percentage of accuracy of the answers between the participants who used the history pie and those who did not. Thereby this questionnaire would answer the question whether the history pie was useful or not. When filling out the questionnaire the participants were reminded that they could use the Impromptu interface at any time to help them with the answers. For cards post task questionnaire see appendix 7. After all three tasks were completed, the participants filled out a posttest questionnaire that asked them to rate the various history features of the application. They would get statements such as: I could see the history of a file, and were then to circle the words that best described their experience ranging from strongly disagree, disagree, neutral, agree to strongly agree. The opinions provided in this questionnaire, were later taken into account in the final debriefing interview.

5.4.2 Brief interview

At the end of a session, when all tasks and questionnaires had been completed, each participant was informally interviewed by respective test observer. The interview was when the participants really had the chance to fully explain their views, opinions and understandings of the interface and its features. In the interview the participants were asked what they liked the best and the least about using the application. Depending on what they had rated as “strongly disagree with” or “strongly agree with” in the posttest questionnaire, they were asked to describe this opinion more, give examples of why they found it either way, how did they use it, etc. It asked them of how they played the cards game, if the rings helped them for guidance of which cards had been opened by others and if they would like to see further visualizations of history.

The interview was concluded with two questions regarding three different screenshots of different Impromptu workspaces. For the first question, the participants were presented a screenshot taken during a session with three users. The participants were informed that one of the users had just joined the session 30 seconds ago and the other two had been in the session for a longer time. The participant was asked to guess which user had just recently joined the session, and to explain why he/she guessed at that particular user. This question and screenshot was designed to see whether the user would use the activity wear as an indicator of who just joined or if would just go by the rings.

For the second question, the participants were showed two different screenshots. They were explained to been taken from two different working situations. One of them was from a session with a teacher and two students. The students were explained to be working separately on their own papers for a project, where the teacher would help them by adding comments in their documents. The teacher was also sharing templates for the students to be helped by. The second screenshot was explained to be from a session where the users performed the same lists task that the participants had just performed earlier, where the users were to add items to lists of as many different files as possible. The participants were then asked to guess at which screenshot belonged to which of the two working situations just explained and to provide justification to their guesses. This two-part question was designed to get an understanding of how the users would interpret the activity visible in the screenshots and to see if they had fully understood the history visualizations presented there.

For these two screenshots questions it was not most important that the participant was able to guess the correct answer, but rather how they explained they reached their conclusion, for

example: “regarding the rings I see...”, “the activity border of user X looks like....” etc. For debriefing interview see Appendix 9.

5.5 Participants

The user study was run with a total of 12 participants. For user studies, a minimum of 10-12 participants should be per condition is considered needed for true experimental design. If running the study with fewer participants, the sample size might not yield statistically valid results (Rubin, 1994 [30], p 93). For definition of statistical significance, see chapter 6.1.3.

We were recruiting undergraduate and graduate students from the School of Information and Computer Sciences. We had some initial problems with recruiting students since the user study was to be held during the same week as mid-terms would be in session (the exam period halfway through the semester). We recruited further participants through personal contacts, a majority being graduate students. The age of the participants ranged from 19 to 33 years old, with the mean age of 25.25 years. The average of years of programming experience was 6.7 years and the average years of computer studies were 5.3 years per participant. The gender division was one female per four participants, making a total of 3 female participants out of 12.

5.6 Test observers and confederate

Besides the test administrator and the two test participants, there were also two test observers and one confederate present at each session. The role of the test observers was to observe and note down the behavior and actions of the test participants. Each participant had respectively one test observer seated next to them at their desk. During the task the test observers were also instructed to help the participants with basic issues like asking them to be patient if a word document was slow to open, help them with clicking ok when an authentication window would popup (this happened quite often when dealing with word documents), etc. The test observers also conducted the debriefing interview at the end of the session with respective participant.

The role of the confederate was to be the third user of the session, in addition to the two “outside” participants. The confederate was for all 6 sessions one and the same, being a member of the Swirl research. The confederate had scripted instructions of how to perform in the tasks, which were the only parts of the session that he was active in. For both cards tasks he were to quickly open up one of his own high cards and then wait and open cards that the other users had open. This was to ensure that we would have some files with history of at least two users. For the lists task he was instructed to open lists that other participants had opened, also to produce files with a lot of different activity. For this task he was also to open a file twice with some other participant having opened it and edited it in between. This was to make sure one of the questions of the post task questionnaire would have valid history (see question 6A in lists post task questionnaire, Appendix 7).

Right at the end of each task the test observers and the confederate would all simultaneously on cue from the test administrator, move in and take a screenshot of respective screen to be

saved with study data. Immediately following, they would also hit the “Save log files” menu option to write all logs of the files to a backup file, to be later saved with the study data. Test observers and the confederate would also on direction from the test administrator change and set the correct options for each task of which history features should be displayed or not. Lastly they would hit the “Reset session” menu option before each new task to ensure that the session time would be restarted. This was done to clear the space of old task files and also to make sure that the new activity arcs in the history pie would be more distinctly visible. Since the history pie’s timeline spans over the whole session, an action that happened at for example time 14:40 in a session that started at 14:00 would for a long period only be visible as a small dot just at the center of the dot, before some would pass and the arc would slowly ripple out towards the edge of the history pie.

For saving data of the tasks and for enabling restarting of session in between tasks I implemented two new options under the menu Options in the interface, where also all the options of history features display were placed. The first addition was an option to save all log files that would write the history log of all files in the session to a specific log file in the folder where the Impromptu was run from. The second was a reset of session button that would clear the workspace of all files and reset the sessions start time to that of the present. By hitting reset, special reset-log files were also saved to the same specific folder as the normal log files were saved in, as a back up to make sure we would really have all data saved from the task. The loss of data had been a problem in the initial user study conducted in 2005, which lead me to be more cautious and have extra back ups for the recording and saving of data.

5.7 The technology and the setting

The study was conducted in a Lab on UCI campus. Three high-speed desktop PC’s were used to run the Impromptu application on. They all had 19 inch (48,26 centimeters) large LCD monitors, enabling a much larger display of the application than the prior study that was run on machines with smaller screens. These three PC’s each sitting on a separate table situated in a circle with all three participants facing in towards the center so that they could see each others as well as the screens. The test observers were positioned slightly behind to the side of the participants, as not to disturb them.

The debriefing interviews were audio recorded on digital recorders, after receiving an oral approval from the participants for this. For the interview, one pair of participant and test observer to would move to a more distant part of the room to allow for a more relaxed private interview environment where the participant could speak freely.

6. Results

In this chapter I will present the results that emerged from the user study. Included will also be the observations made by the test observers of the users during the session. I will start by presenting the numbers and results of the post task questionnaires and describe the statistical test that was performed on this data. This is followed by a summary of the users' opinions and ratings of the features of the post test questionnaire. The last section will present what emerged from the final interviews with the participants. All data from the user study has been summarized in three spreadsheets; Appendix 10, 11 and 12.

6.1 Post task questionnaires

Since the post task questionnaires were very precisely structured (see Appendix 6 and 7), I could calculate very exact figures for the answers to the questions of the activity during the task. To evaluate the accuracy of the participants' answers in the questionnaire I compared them to the logs files of all the files' history and actions that had been saved after each task of each session. I evaluated the post task questionnaires question by question, grading the answers with 1 for a correctly answered question and with 0 for an incorrect answer.

To get a measure of the usefulness of the rings/ripples, I compared the results of accuracy between the cards task when the rings/ripples were enabled, visible, and the cards task when the rings/ripples were disabled, not visible. This yielded a very precise result of how much better or worse the participants were able to answer the questions correctly when having the rings/ripples visible or not visible.

For the list task I compared the accuracy in answering history questions between those participants who used the history pie and for those who did not use it. This also yielded precise results as to how much better or worse the participants were able to answer the questions correct when using the history pie compared to those who didn't.

All data from the questionnaires is presented in Appendix 10, 11 and 12.

6.1.1 Cards task – rings/ripples

As explained in the chapter 5.3.1, the cards task were organized so that in the first half of the study, sessions 1-3, the first cards task would not have any rings/ripples displayed in the interface, where the second cards task would have the rings/ripples displayed. For the second half of the study, session 4-6, the order was switched so that in the first cards task the rings were visible, but in the second cards task they were not.

For the cards task when the rings/ripples were not displayed and invisible, the average accuracy of the answers was 2.83 out of a total 4, for session 1-3, and 2.33 for session 4-6. In total, the average accuracy was 2.58 out of a possible 4 for all sessions for the cards task when rings/ripples were not displayed.

For the cards task when the rings/ripples were displayed and visible, the average accuracy of the answers was 3.5 out of a total 4, for session 1-3, and 3.33 for session 4-6. In total, the average accuracy was 3.42 out of a possible 4 for all sessions for the cards task when rings/ripples were not displayed. Measure in percentage, the participants averaged on 65% accuracy when not seeing the rings/ripples displayed compared to an average of 85% accurate answers for the cards task when the participants did see the rings/ripples.

6.1.2 Lists task - history pie

The post task questionnaire of the list task was also very specifically targeting the activity during that task. As described in chapter 5.3.2, the questions were designed so that the users would have to consult the history pie of files to be sure to answer the questions correctly.

Out of the 12 users in the study, 7 users used the history pie to answer the questions of the post task questionnaire, i.e. 58 % of the users used the history pie. Out of the 5 other users, 3 users talked about the history pie later on in the interview, 1 of them admitted to having forgotten to use it during the list (user 4.1) The remaining 2 users did neither use nor talk about the history pie.

To get a measure of the usefulness of the history pie I compared the accuracy of the post task questionnaire answers of the two groups being those 7 users who used the history pie vs. those 5 users who did not use it.

The users who used the history pie had an accuracy of 82% when answering the questions, meaning each user got an average of 5.7 out of 7 answers correct. For the second group of users, the five users which did not use the history pie, the accuracy was 49% for answering the questions correct. The average user of this group got 3.4 out of a total of 7 answers correct.

6.1.3 T-test

To assess whether two groups being compared have means that are statistically different to each other, one can perform the t-test. By proving that the data is statistically significant it is considered very unlikely that the difference of two study groups is not real. The t-test is a statistical test that judges the difference between the means of two sample groups relative to the spread or variability of their scores (Russell Bernard, 2002 [31], p 562-568). The formula to the t-test is a ratio, with the top part of the ratio being just the difference between the two means. The bottom part is a measure of the variability or dispersion of the scores which is calculated by taking the variance for each group divided by the number of people in that group, adding these two values and taking their square root.

In other words the t-test formula, shown in Figur 7 , is comparing the difference between the group means to the variability of the groups.

$$t = \frac{\bar{X}_T - \bar{X}_C}{\sqrt{\frac{\text{var}_T}{n_T} + \frac{\text{var}_C}{n_C}}}$$

Figur 7: T-test formula

To test the significance, one has to set a risk level, called the alpha level. This risk level signifies the odds that the observed results are due to chance. In most social research the rule of thumb is to set the alpha level to 0.05 meaning that five times out of a hundred you would find a statistically significant difference between the means even if there was none. The degrees of freedom (df) must also be set, which is calculated being the total number of people in the two groups minus two. Once the t-value has been calculated, in combination with the alpha level and the df, the t-value can be compared to a t-value in a standard table of significance to determine if the t-value obtained is large enough to be significant.

When testing the possibility that one mean will be higher than another, the one-tailed t-test is the one to use (Russell Bernard, 2002 [31], p 565). Since this was the case for our testing, the one-tailed t-test was used.

6.1.1.1 T-test on rings/ripples data

To assess whether the means of the two groups were statistically different from each other, I performed the t-test on the data.

For the t-test of the rings, group one was the sample data from tasks without the rings, and group two was the sample data from the tasks with rings visible. The degrees of freedom was 22 (N-2, where N is the total number of people in both groups) and the alpha level was set to 0.05.

Given the values above, the t-test resulted in a t-value of 1.88. To determine if a t-value is large enough to be significant it is compared to a value in a standard table of significance. If the t-level is large enough, it concludes that the means of the two groups are different. The table gave that for this data the limit for this samples statistical difference was 1.717 (Russell Bernard, 2002 [31], p 707). Our value 1.88 is above this and the sample data is thereby proven statistically different.

6.1.3.2 T-test on history pie data

To prove that the data was statistically significant, the t-test was performed. For this t-test of the history pie, group one was the sample data from users using the history pie, and group two was the sample data from the users not using the history pie feature when answering the lists post task questionnaire

The degrees of freedom was 10 (N-2, where N is the total number of people in both groups) and the alpha level was set to 0.05.

Given the values above, the t-test resulted in a t-value of 2.52. The standard table of significance gave that for this data the limit for this samples statistical difference was 1.812 (Russell Bernard, 2002 [31], p 707). Our value 2.52 is well above this and the sample data is thereby proven statistically different.

6.2 Posttest questionnaires

The posttest questionnaire asked the participants about their experience using the software (see Appendix 8). The participants were asked to rate the various statements about the history features of the application on a scale ranging from strongly disagree to strongly agree. For grading, I've translated the scale to the following point system:

strongly disagree	=	1p
disagree	=	2p
neutral	=	3p
agree	=	4p
strongly agree	=	5p

By replacing each user's answers by this point scale system I was able to calculate mean values for the various statements. Using this translation table, a statement receiving a 5.0 mean would be a statement all users strongly agreed with and consequently a statement receiving 1.0 would be a statement they all strongly disagreed with.

The first statement, in the posttest questionnaire, was that the rings helped the user see what other users were doing. All users either agreed or strongly agreed with this statement, and it had a mean value of 4.17.

The second statement dealt with the matter of being able to see the history of a file. Most users strongly agreed with this statement (7 users). Only one user was neutral to this statement. The same user later expressed confusion and dislike over the fading of rings. The remaining 4 users agreed with the statement giving it a mean value of 4.5.

Third up was the "I could see what other users were doing" statement that resulted in the lowest mean of 3.83 of all the statements. 8 of the users were equally divided between strongly agree and agree. Out of the 2 users who expressed themselves neutral and the 2 users who disagreed with the statement, 2 users pointed out that they could not tell concurrent actions (user 2.2 and user 4.1). They explained that they could see first afterward if a user had edited a file, i.e. when the save of the file occurred, but could not tell if an open file by a user was currently being edited or only kept open. 1 user also said expressed that there was no way to tell the exact nature of the activity, whether it was a read or a write (user 4.2).

The fourth statement that the user could see which user was the most active at the moment, which received an average 3.92 rating. One user expressed confusion over the disappearing rings (user 2.1) yet another user said that it was hard to keep track of which user was most active since the word window (during list task) would be in the foreground when worked on, and obscure the Impromptu workspace under it (user 3.2). This was an issue that arose frequently in the prior study and is currently addressed in another Swirl project that studies

the role of PDA's as auxiliary awareness channels (Silva Filho R. S., Redmiles, D. F, 2006 [27]). Unfortunately, no user mentioned explicitly how they kept track of the accumulated activity of the users, whether it was by the activity wear display (the width of border of the pie slice) or by comparing the number of colored rings in the workspace. But since these two features are meant to work together it is not explicitly important for the results to know which of the features helped the users, but rather the fact that most of them were able to keep track of it

The strongest rating in the posttest questionnaire was acquired by the fifth statement that said that the user knew when other people were interacting with his/her files. It received an average rating of 4.58 with 7 users strongly agreeing and the other 5 users agreeing. The users would see when others were interacting with their files by seeing rings appear around their dots.

Statement six dealt with the issue that the user knew that other people could see what the user was doing. It received the average rating of 4.0 due to two low ratings among the other 10 users who either agreed or strongly agreed. The two users who disagreed expressed that they did not simply think of the fact that others could see what they were doing (user 3.2 and user 5.1), with one of them stating that there was no feedback to call the attention to that others knew what they were doing (user 3.2). The other agreeing users expressed that since they could see what others were doing, they knew that the opposite was also true: that others could consequently see their own actions.

The last and seventh statement that the rings around the dots helped the user with better knowing the history of files, while doing the two card tasks, had the mean rating of 4.17. With 4 users strongly agreeing and 6 users agreeing, one of the 2 users who were neutral to this statement expressed that the rings had faded when the user was to answer the post task questions (user 1.1). The second neutral user could not remember, when asked later in interview, why he/she rated it as only neutral (user 1.2

In total, for all users regarding all statements the rating was distributed as following: 30 ratings of strongly agree, 43 rating of agree, 6 rating of neutral, 5 rating of disagree, 0 rating of strongly disagree.

6.3 Interviews

Where the post task questionnaires gave very precise data to analyze the accuracy of, the interview was the final part of the session when the user could freely express all his/her thoughts and experiences of the application (see Appendix 9). In the interview the users were also asked to explain more about some of the ratings they had done in the posttest questionnaire (some of these explanations have already been presented in chapter 6.2). The screenshot questions at the end of the interview also gave interesting information of how the users perceived and understood the interface and its features.

Overall, the interviews showed a strong support and approval of the visualization of history from the users. When asked what the users liked the best about the application, user 3.1 replied "*To see the history of the activity.*" This answer, in short, sums up the general feeling of all the users.

6.3.1 Rings/ripples

All users liked and found the rings and ripples useful. User 5.2 answered the following when asked what he/she liked best about the application: *“The ring effect was nifty, the ability to see what has happened to a thing in the past. And the fact that the rings grew over time gave you a sense of activity, I knew this was more of a hot file than the others.”* The user also added that more colors meant more activity; aiming at the fact that the colors caught his/her attention to that activity was happening on a file. User 6.2 also said that the colorfulness helped him/her keep the attention focused on the workspace.

9 out of 12 users used the rings and ripples as guidance of which cards the others had picked during the card task. User 4.1 explained his/her strategy during the card tasks as: *“I was looking at the history of users’ own cards. The first time I was just looking at which files were opened most. The second time I focused on who was opening their own files.”* User 4.1 continued by saying: *“It was useful to see the history of the file. If someone picks their own card, it must be a good card. I used that strategy.”*

User 5.1 said that since the card tasks were slower paced (than the lists task), he/she had more time to pay attention to what was going on. He/she said to have immediately noticed the when a new ring would appear, especially in the task when the rings were visible. *“So I knew which files people were opening and it was kind of part of the strategy of the game to let other people open files first and see which files they opened. So it definitely let me know what files people were opening... that might prompt me to open the same files or not, depending.”* (User 5.1).

For the lists task, 7 users looked at the rings to get an idea of which lists others had already been added to or opened. One user liked the fade away effect of the rings and thought it was cool (user 1.2). 3 users added extra writing on their questionnaires saying that they had to rely on memory for answering some questions since the rings had faded out (the history pie was not visible during the two card tasks. Only 1 user out of 12 did not understand the ripple out effect of rings, that the innermost ring always represents the most recent event (user 4.2). All users looked at the screen when answering the questions of the post task questionnaires, at which point they were going off the rings for the answers.

An interesting observation was how some users reasoned that two consecutive rings in a row, from the same user, would mean an opening of the file and an edit to the file. Three users explicitly expressed this when talking about how they could see history. User 4.2 said he/she used this as a way of telling whether a file was still open by someone and under edit so to speak. User 2.1 explained that: *“Double rings tell me the user opened and edited the file. One ring means they could be editing now”*. With small history pie this user said that he/she could see if they had only opened it or also saved it. The user expressed that one line means the user who opened the file could be editing it right now. This way of interpreting two consecutive actions from one user, represented as two rings around the dot and two arcs in the history pie, was something we made use when debugging and trying out the study tasks. But this way of using the fact of two consecutive rings by the same user, as interpreting as the first being an open (read) and the second most probably being a save (write), was never presented to the

users of the study. They figured this out by themselves. This also yields strong support that the design of the display of actions worked and was well understood by the users.’

6.3.2 History pie

The interview questions did not specifically ask the users about the history pie feature. The feedback and opinions about the history pie do all amount from when the users talked about what they liked best of the application, how they were able to see history etc. “*Even if I loose the ring, you can still see that*” said user2.1, who liked the persistence of the history pie.

User 3.2 expressed that without “*the history view in the smaller corner thing*” (i.e. the history pie), the user would not had been able to answer the questions of the post task questionnaire, since the word window often had covered the application window during the task, making the user miss events going on. User 3.2 also explained that he/she could see the full history of a file: “*Based on the little history panel [the history pie], that small thing that came up, not the concentric circles. I really liked that par [the history pie]t. That was constant; I could go back and refer to it.*”

6 users in total explicitly expressed general liking of the history pie. As user 1.1 said, when asked what he/she liked best with the application: “*The little history figure on the right side – history pie. It was the most reliable place to see history because circles kept fading. You may not get the whole history unless you’re paying attention*”. The interviews also confirmed the design thought of the history pie to be a complement to the rings. By showing the full long-time history, a user can always fall back on the history pie to get updates of what happened while the user was away from the interface. User 1.1 strongly agreed to statement 2 of the post task questionnaire, saying that he/she could see the history of a file. When asked about this rating in the interview the user answered: “*It was very clear once I saw the small history figure*”, referring to the history pie.

Many users expressed similar approval of the history pie, confirming its usefulness. User 1.2 liked the history pie best, when asked about the application. “*I definitely liked being able to having it persistent, so once the rings would go away, it’s nice to be able to see that again [the history of the file]. I felt I had a pretty good understanding of the accesses to the file by looking at that view.*” (user 1.2)

Following the same line, when user 4.2 was asked what he/she liked best of the application, the answer was: “*I liked that little corner thing [history pie], ‘cause you could see the order very easily of who had accessed it [the file]. It was a lot easier to see than the rings around, ‘cause they faded. It was very useful, afterwards, to go back and see, down the line, which people had done it [accessed the file]*”. The same user also strongly agreed with the posttest questionnaire statement that he/she could see the history of a file. The user explained that: “*“on the screen, by the time the rings had faded away, and I still wanted to see the order of who had accessed it [the file], I could just hover on it [the dot] and it would appear and when I was done with it, it [the history pie] would go away. So it was there when I wanted it, and it wasn’t in my way when I didn’t want it. That’s why it was really easy to use. It was really easy to get it and then make it go away.*” The design idea of displaying the history pie on mouse over and not having it pop up in the center of the view, which would obstruct the

workspace and files behind it, but rather show in the bottom right corner outside the pie is confirmed and clearly approved by this user.

7 out of 12 users used the history pie to answer the questions of the lists post task questionnaire. As presented in chapter 6.1.2 those users answered 82% of the questions correct, whereas the users who didn't use the history pie only scored 49% correct. Clearly, the history pie visualization helped the users to better know the history of files.

6.3.3 Activity wear and screenshot 1

The two last questions of the interview focused on three different screenshots and what they displayed (see Appendix 9). The question asked of screenshot 1 was designed to have users look at the activity wear to come to the right conclusion. Unfortunately there were no other parts of the session or the tasks when the users specifically came to use this feature.

5 users out of the total 12 explicitly expressed that they looked at the activity wear, the thickness of the pie slice border, to derive the conclusion as to which user had most recently joined the session depicted in screenshot 1. User 2.1 concluded that the newest arrived was: *"Lina, because she has the thinnest outside ring, which means the least activity."* User 3.2 also chose Lina and said that he/she concluded this by: *"going by the circumference width, the thickness of the circumference."* This was also expressed by user 6.2 who answered Lina and justified the answer with: *"The outer ring is smaller"*.

Most users used the activity wear in combination with the rings. User 3.1 also answered with *Lina* with the following explanation: *"She doesn't have many rings. She only has one inside ring. Also her edge is not thick. It is also possible she may have just been inactive for a period of time."* User 4.1 also went by a combination of the two features to derive his/her reply: *"Lina. Her activity bar is smaller. I'm guessing she hasn't had time to do much access on the files. She only has one ring and it's an inner ring."*

Yet others did not talk specifically about the activity bar but gave answers such as user 5.2: *"Lina. Others have developed rings on outer elements. Lina's only progress was inner-most rings – which is the most recent activity."* User 5.1 answered ambiguously saying: *"My guess would be Lina since she has the least amount of activities"*.

For this screenshot number 1 question there is not 100 % correct answer. It could be a result of many different activity scenarios, all depending on how fast the users in the screenshot had gone about doing the activity displayed. But by consulting the rings, *Lina* only has one inner ring and no other. Where as the other two users has quite a lot of rings. *Lina* also has the thinnest border of the pie slice representing she has done least actions of the three users during the last 5 minutes. 11 out of 12 users answered with *Lina* although quite a few of them also mentioned that it could also have been user *Kari*, who had just jumped into the session doing a lot of activity and that the rings of *Lina* could have faded away (meaning older activities).

Since none of the three tasks were designed to have the users looking at the activity wear, most users never mentioned looking at this during their debriefing interview. There was however one user who explained that he/she had used it in combination with the rings to help him/her know when other people were interacting with his/her files: *"The rings, and the*

colors and that their wedge would increase. So at first I would see the rings there and then it would be confirmed; their wedges are increasing, so ok they're looking at it right now. "

6.3.4 Screenshots 2 and 3

For screenshots 2 and 3 that were representing two different working situations (see Appendix 9) there were correct answers. 11 out of 12 users answered correctly, that screenshot 2 was the teacher-student session and that screenshot 3 was taken during a lists task session. The users talked about how they saw in screenshot 2 that *Kari* and *Roberto* had not touched each others files, but that they had both opened *Lina's*, and that *Lina* had opened both of their files respectively.

For screenshot 3, the lists task, user 1.2 described: *"It has a lot of read actions and write actions, more activity overall."* This was the third user to refer to two consecutive actions in a row by the same user would mean a read and write. Others looked more of the accumulation of rings and colors to determine that there was an equal amount of activities going on in-between the users. When explaining screenshot 3 as being the lists task, user 2.1 said: *"Because there are more rings, and more colors per documents. So looks like lots of people are opening and closing files."*

6.3.5 Users did not like

There was also negative critique about the application and the features. Most of them were oriented around the fade out of the rings. 5 users expressed that they felt the rings faded out too quickly. 2 users did not like the fade out at all. 1 user did not like the fact that the innermost ring, directly attached to the dot did not fade out but stayed on.

There was also concern over the inability to clearly see an inner ring of the same user who owned the file. Since there is no gap between the inner ring and the dot, if a user opens his/her own file, the dot will only look to become slightly larger. This was problem that 3 users expressed during the interview. The white ring around the dot was confusing to one of the users, who admitted to having missed the explanation of this during the training task.

One user thought that the outer rings were more eye-catching than the innermost ring around a dot. This user understood the rings and ripples feature but did not think it intuitive. A few users mentioned the fact that when opening some other application, this window would obscure the Impromptu workspace and potentially make them miss important events. This was an issue brought up also in the prior user study and is currently under work in the thin client project (Silva Filho R. S., Redmiles, D. F, 2006 [27]).

The other big issue of dislike was the speed. 5 users complained about the application being slow, the word documents took some time before they opened etc. One user pointed out that he/she had noticed a lag between the mouse-click and the reaction of it in the interface. 2 users noted that when putting the mouse over a dot, they had to wait for the history arcs to draw. An impatient user could mistake a file's history pie to be empty, which means a file with no actions on it.

In reference to seeing what other people were currently doing, 2 users pointed out that they could first see afterwards what others had done, not while they were doing it. The ring representing the action, and the arc in the history pie would not become visible until the save/write event occurred. The users would have liked to have seen if other users were currently under way of editing a file or if the file was just opened (or even just closed again without edit).

Some the negative critique did in fact support the intention of my design features and confirmed that the users had used them as intended. This was most clear in the case of the fading rings. When asked about what he/she least liked about the application user 5.1 replied: *“The rings disappear kind of fast, so that the actual rings in the center of the display were less useful than the history graph on the side.”* As in regards to the fade out time the same user continued to explain: *“It was a little bit too fast for me, I started missing out on events so that I had to look at the history.” [History Pie]*. Even though the users obviously had wanted rings to stay on longer for the sake of this specific task, his/her experience and behavior of resorting to the history pie when rings had faded follows exactly how I intended the designs to work and confirms the design idea of the history pie to be used in complement with the rings.

User 2.1 also explained similar usage of the history pie: *“The rings faded away quickly, which made me go back and use the infinite view, when the rings had faded [history pie]. So when the rings would go away I had to go back and consult that little thing in the corner [the history pie]”*.

6.3.6 Users would like to see more of

Being able to distinguish visually between a read and write action (open and edit) on a file was the most popular feature that emerged when the users were asked what they would have liked to see more of. 5 out of the 12 users wanted to see some difference between read and write actions both in the rings, and in the history pie.

2 users expressed that they would like to see if a file is currently being open by the other users or if the file had been closed it down. Some three users also wished to have seen how much and edited file had been changed, shown in percentage. These users were saying it could have helped them during the lists task to know how many lines the previous user had edited it with.

In response to not seeing any signs of concurrent actions, 2 users said they would like to see the action of editing while it is under way, and not just get notified of it afterwards when the file is saved. 1 user also thought it might be useful to know how long a user had “been in” the file, i.e. had had it open.

7. Discussion

The outcome of user study strongly confirmed the usability and intended function of the three history visualization features under evaluation that I had extended the application with. The t-test and the comparison to accuracy rates for those who did not use the history pie vs. those who used it, strongly suggests that the history pie helped the users with better getting a full picture of the activities during the session. Similarly the awareness of activities was proved stronger in those tasks when the users could see the rings/ripples to those tasks without this feature. Both the t-test and the percentage of accuracy confirmed this.

There is also the probability that in a more work-like setting, the rings/ripples feature would have proved even stronger. As one of the users pointed out; the card tasks were slow paced, there were only a total of nine actions and there were no other open applications to obscure the workspace since no work had to be performed on the opened cards. These conditions meant the users were eyeing the workspace closely at all time and might have answered out of memory for the card task when the rings were disabled. If, on the other hand, the card task had involved more activity, the gap between the accuracy of the answers of the post task questionnaires (for with and without the rings/ripples) might have been even bigger proving the usefulness of the rings even greater.

The conditions concerning the placement of the lists task, being executed after the users had performed the two card tasks, might also have influenced how the users were apt to use the history pie when answering the lists post task questionnaire. During the first two tasks the history pie was disabled, in favor of specifically measuring the usefulness of the rings/ripples. Consequently I argue that a lot of the users had this way gotten accustomed to not seeing the history pie when mousing over a dot and also gotten familiar with the rings/ripples feature. Even though the history pie was explained and examined in the training task, the two card tasks had such focus on the rings, so that by the third task the users were well accustomed to the rings and many of them had forgotten about the history pie feature. Further more, they were acquainted to NOT seeing the history pie on a mouse over of a dot. To make the argument even stronger, the rings/ripples is a feature that is always there (assuming this feature is enabled in options menu), the users can't avoid noticing it. In contrast, to see the history pie, the user has to take deliberate action and put the mouse over the dot. In chapter 6.1.2 I presented how 3 users talked about the history pie later in the interview, and that one of them admitted to having forgotten to use it at the time. If these 3 users had been more acquainted to the application, which is the intended setting of use for Impromptu, and would have remembered the history pie feature, the study would have yielded 10 out of 12 users using the history pie with even higher numbers of accuracy compared to those users who didn't use it.

A further must also be taken into account concerning the three different features being evaluated. The rings are situated right around the dots that they represent action over. Further more they are within the big pie in the center of the workspace and thus in the center of attention. The other two features are more placed in the periphery of the workspace: the activity wear on the border or the pie slice and the history pie in the lower right corner. Both of these features were deliberately placed outside the center as to not obscure the workspace.

For evaluating the activity wear the study yielded support to that the feature works and is understood as intended and designed. Also in this case, the study could have yielded even stronger support for the usefulness of this feature if we had made sure to ask each user to

clarify and explain in more detail how they were able to perceive activity when answering the question about screenshot 1. User 5.1 said for example that: “*Lina has the least amount of activity*” (user 5.1). Since this answer does not clarify whether the user drew that conclusion from watching the rings or the activity wear, it was counted as not referring to the activity wear. By making sure there were no false positives, the resulting number of study participants who used the activity wear feature is probably lower than was the actual case. In a different actual work-setting, the activity wear feature may also play a bigger role and be a more important feature as to understanding the current activities of the session, than were the case during the session tasks.

As was all the temporal history features, the activity wear and rings/ripples were designed and intended to be used in combination. This idea was strongly supported by the fact that most users who talked about the activity wear when describing screenshot 1, was also talking about the rings. The activity wear can be seen as a sort of summary of all the different rings that user has around files in the workspace. But where rings fade after two minutes, the activity wear shows the accumulated actions over the last five minutes. Since the two complement each other it is not wrong to look at just one of them, but good to use in combination.

The way in which many users reasoned about screenshot 1 strongly supports the conclusion that they had understood and used the visualizations in a correct way. By also considering both *Lina* and *Kari* as being the most recently joined user, they display and understanding of the fadeout of rings, understanding of the general use of the rings, understanding of the activity wear but most importantly how these all work together.

For the specific tasks performed in this study, the dislike of the fading of the rings was expected. Since the fade out makes it harder for the users to answer the questions in the post task questionnaires, it's a given outcome that they will not like this behavior of the rings. However, as also some users agreed during the interview, were they working in a more natural work-setting the fade out of rings would enable a way of knowing what has happened most recently. If all rings would stay on, all dots would eventually fill up with rings around them. There would be no way of telling the difference of current activeness of files between for example file A who had some activity occurring on it over an hour ago but has since then been untouched and a file B which was just opened and is currently worked on. With the current fade out of the rings/ripples the users will always get updated visual information of what is going on in the session.

8. Future work

As suggested by the results of the user study, there is a variety of features the application could be extended with. Some of these suggested by the study participants were also early design ideas that I had in the first part of the project, ideas to features that were not chosen for further implementation due to time limitations. The visualization of the last move of a file, that is show if it has been moved from lower to higher security levels, could be a possible extension. Also displaying the history of shared files might help heighten the security and the awareness since a user would be more informed of which files have been shared in earlier sessions and could thereby be interpreted as more or less safe to share the second time.

One very obvious further development of the rings/ripples and the arcs in the history pie, would be to visualize read and write events differently. This could be implemented by using for example patterns where a dotted ring/arc could represent a read and a full ring/arc would represent a write.

The ability to see concurrent actions was also voiced in the user study as a preference. This might however be fairly difficult to implement due to all the different applications that could be interacting with the files shared in the Impromptu workspace. The same goes for visually showing whether a file is currently still opened by a user or if the user has closed it.

It may be interesting to develop my early design idea of displaying on a mouse over the earlier location of a file, if it has been moved from higher or to lower security levels. Similarly, it might be interesting to display the temporal history of the files shared over many sessions. For example to see if I have shared a particular file before in an earlier session or not. This may affect how I choose to share different files, and to which security levels I will import them.

An evaluation of the fourth history feature: the user characterization would also be preferable. One Swirl group member is currently working on visualizing other similar network characterizations. The future user study of those features may come to include also my user characterization, since these features are all very similar to each other.

9. Conclusion

Imagine the perfect security system without any breaches, theoretically speaking. Then imagine the end-user, one of a larger team, employing this security system to keep data confidential and safe. Now what happens if the user does not fully understand the system? He configures it, uses it and considers himself safe from attacks. Is he? Not in this case, since by not understanding the system and by using it incorrectly he might actually be putting himself and the data at greater risk than before the system was employed. Conclusion? Security systems that are not usable and understood by end-users can never render effective security.

In this report I have explored the idea of visualizing the activities going on in a system, so to better make users aware of what is going on. My visualizations have drawn on the concept of social navigation to make people's social trails through information spaces visible and thereby aid the users in making more informed choices about their security.

My project has been built on the base application of Impromptu, an ad hoc peer-to-peer file sharing application. Impromptu was implemented to make relevant security features visible to users, allowing them to observe and understand certain events on the network and be better prepared to take appropriate actions. Three major design principles are combined in Impromptu: visualization mechanisms, multi-source event-monitoring architecture and integration of action and configuration.

The aims of my Master's thesis were to enhance the file sharing application Impromptu with further visualizations of temporal history of both activities within the session and relationship between users of present and earlier sessions. My extensions to the applications were to facilitate for users to have a better understanding of what is going on in a session. To help them make more sense of the different activities and thereby help augment their own assessment of security, by better knowing and understanding the situation.

My designs were a direct response to the outcome of the prior 2005 user study of the basic Impromptu interface. Four distinct features were designed and implemented. The dots, representing files, were extended to include three additional rings/ripples to be able to show more of the recent history of a file. The history pie was designed and implemented to complement the rings by displaying the full history of a file, being all activities that had happened on the file during the session. The third feature was a visualization of each user's accumulated activity over all files, to show which user has been the most active recently relative to the other users. The fourth feature was user characterization that displayed history over sessions; whether users in the current session has been encountered before in earlier sessions or whether new and unfamiliar users are present.

To get a usability measure of these new features I conducted a user study in a laboratory setting, evaluating three of the four features. User characterization was not evaluated since I was conducting a laboratory study. A fair evaluation of this session-after-session related feature would only be possible when studying people using the application over a longer period of time and for multiple sessions.

The results of the user study strongly supported the usability of my features, and confirmed that they aided the users to better understand the activities of the session. Since the user study was very specifically targeting my features the outcome produced exact figures easy to measure and compare. By performing the t-test on the data, I also proved that it was

statistically significant. My features augmented the user's ability to see the collected activities of the session. In short, with the features the users knew better what was going on.

By knowing better the situation the users are in, the better chances they have to evaluate their own security situation correctly. By knowing what is going on, the users are better informed to take appropriate decisions concerning their own security. My visualizations clearly help users to better know what is happening in a session and thereby aid them in their assessment of their own security.

10. References

- [1]. Rode J., Johansson C., DiGioia P., Silva Filho R., Nies K., Nguyen D., Dourish P. and Redmiles D. 2006. Seeing Further: Extending Visualizations as a Basis for Usable Security. 2006 Symposium on Usable Privacy and Security (SOUPS 2006), Pittsburgh, PA, United States.
- [2]. The Swirl Project, <http://www.isr.uci.edu/projects/swirl/index.html>, 2006-06-10.
- [3]. Java Sun Microsystems, <http://www.java.com>, 2006-06-10.
- [4]. Scalable Vector Graphics, <http://www.w3.org/Graphics/SVG/>, 2006-06-10.
- [5]. Extensible Markup Language, <http://www.w3.org/XML/>, 2006-06-10.
- [6]. Whitten, A. and Tygar, D. 1999. Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0. Proc 8th Usenix Security Symposium.
- [7]. Saltzer, H. J. and Schroeder, D. M. 1975. The Protection of Information in Computer Systems. Proceedings of the IEEE, 63:9 (1975), 1278 – 1308.
- [8]. Cranor, L.F. and Garfinkel, S. 2005. Security and Usability. Chap 2: Usable Security by Sasse, M. A. and Flechais, I. United States: O'Reilly
- [9]. Zurko, M. E and Simon, R. T. 1997. User-Centered Security. New Security Paradigms Workshop 1997.
- [10]. Cranor, L.F and Garfinkel, S. 2005. Security and Usability. Chap 1: Psychological Acceptability Revisited by Bishop, M. United States: O'Reilly
- [11]. DePaula, R., Ding, X., Dourish, P., Nies, K., Pillet, B., Redmiles, D.F., Ren, J., Rode, J.A., and Silva Filho, R. (2005). In the Eye of the Beholder: A Visualization-based Approach to Information System Security. International Journal of Human-Computer Studies, 63(1-2), 5-24.
- [12]. Cranor, L.F and Garfinkel, S. 2005. Security and Usability. Chap 3: Design for Usability by Tognazzini, B. United States: O'Reilly
- [13]. Dourish, P. and Chalmers, M. 1994. Running Out of Space: Models of Information Navigation. Short paper presented at HCI'94, Glasgow, United Kingdom.
- [14]. Hill, W., Hollan, J., Wroblewski, D., McCandless, J. 1992. Edit Wear and Read Wear. Proc. ACM Conf. Human Factors in Computing Systems CHI'92, Monterey, California, United States.
- [15]. DiGioia, P. and Dourish, P. 2005. Social Navigation as a Model for Usable Security. Symposium on Usable Privacy and Security SOUPS 2005, Pittsburgh, PA, United States, 101-108.
- [16]. W3C, <http://www.w3.org/>, 2006-06-10.
- [17]. Batik SVG Toolkit, <http://xmlgraphics.apache.org/batik/>, 2006-06-10.
- [18]. Eclipse, <http://www.eclipse.org/>, 2006-06-10.
- [19]. 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.
- [20]. DePaula, R., Ding, X., Dourish, P., Nies, K., Pillet, B., Redmiles, D.F., Ren, J., Rode, J.A., and Silva Filho, R. 2005. Two Experiences Designing for Effective Security. Symposium on Usable Privacy and Security SOUPS 2005, Pittsburgh, PA, United States, 25-34.
- [21]. Steinberg, D. and S. Cheshire. 2005. Zero Configuration Networking: The Definitive Guide. United States: O'Reilly.
- [22]. 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, California, United States.

- [23]. Good, N. and Krekelberg, A. 2003. Usability and privacy: A study of Kazaa P2P file-sharing. Proc. ACM Conf. Human Factors in Computing Systems CHI 2003, Ft Lauderdale, Florida, United States.
- [24]. Jetty Java HTTP Servlet Server, <http://jetty.mortbay.org/jetty/>, 2006-06-10.
- [25]. The Jakarta Slide Projects, <http://jakarta.apache.org/slide/>, 2006-06-10.
- [26]. Goland, Y., E. J. Whitehead, et al. 1999. HTTP Extensions for Distributed Authoring -- WEBDAV, Internet Engineering Task Force: 1-94, RFC 2518.
- [27]. Silva Filho R. S., Redmiles, D. F. 2006. Extending Desktop Applications with Pocket-size Devices. 2006 Symposium on Usable Privacy and Security (SOUPS'06) Pittsburgh, PA ,United States.
<http://awareness.ics.uci.edu/~rsilvafi/papers/Posters/SOUPS2006-ThinClient.pdf>
- [28]. Dumas, J. S. and Redish, J.C. 1999. A Practical Guide to Usability Testing. UK: Intellect Books.
- [29]. Institutional Review Board, <http://www.rgs.uci.edu/ora/forms/irb.htm>, 2006-06-10.
- [30]. Rubin J. 1994. Handbook of Usability Testing: How to Plan, Design and Conduct Effective Tests. United States: Wiley.
- [31]. Russell Bernard, H. 2002. Research Methods in Anthropology: Qualitative and Quantitative Approaches, Third Edition. United States: AltaMira Press.

Appendix 1

Study Information Sheet

University of California, Irvine
Study Information Sheet
for

“Trustable Technology: Temporal Visualization for Balancing Privacy and Accessibility in Mobile Ad Hoc Networks”

Carolina Johansson, ISR Visiting Researcher
Cjohanss@ics.uci.edu
562-881-1414

Donald Bren School of Information and Computer Science, Informatics

- You are being asked to participate in a research study about how to share files with other participants and then monitor and set privacy permissions on these files.
- The study will consist of a session of roughly 1 hour. The session will consist of 5-7 shorter tasks to be completed. Each task will be followed by a short debrief. After all tasks are completed a complete debrief and interview section will follow. Please let us know if you have any questions. We look forward to your participation.

- The only foreseeable discomfort associated with the study is the time you spend in the study. We will provide all the data and computers. There are no direct benefits from participation in the study. However, this study may allow for advances in privacy research.
- Participation in this study is voluntary. You may refuse to participate or discontinue your involvement at any time without penalty. You may choose to skip a question as well as decline to be audio taped. If you chose to be recorded the tapes will be erased at the end of the study. All information collected will be kept in a locked closet or cabinet and participants will be identified by number in order to guarantee confidentiality of all participant data.
- If you have any comments or questions regarding the conduct of this research or your rights as a research participant, you may contact the University of California, Irvine, Office of Research Administration by phone at (949) 824-6662.

Appendix 2

IMPORTANT: Your name will not be tied to your responses on this questionnaire. Your answers **will not be used** by us or others for marketing or sales purposes. This is **only for research**—to help us build technology.

<i>SWIRL Research Use Only</i>								
<i>Session No:</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>
<i>Participant No:</i>		<i>1</i>					<i>2</i>	

Pretest - Background Questionnaire

1. Your **age**: _____
2. What is **your first language**? (check one)
 - English
 - Other (please specify: _____)
3. How many years of programming experience do you have? _____ years
4. What programming languages do you know?
 - Java
 - C, C++
 - HTML
 - Other. Please specify: _____

5. Have you done any user interface design?

Yes

No

6. How many years of Computer Studies (Computer Science, Informatics) have you done?

_____ years

Thank you!

Please return this questionnaire to the Test Administrator.

Appendix 3

Training Task Script

As you can see on your screen, you are now three users connected and using the application.

You will see each of your username framing your specific pie slice.

On your desktop you see a file named “Testing_X” (where X = A, B or C). Import this file

“Testing_X” into your pie slice in the workspace. Drag it into the third ring of sharing counted from the border, labeled “read/write” where other participants will be able to read and modify the file. The file will be represented as a dot, with the same color as you have. The white ring around the dot shows that this file has not been opened by anyone yet.

For all the files that you will be importing into the workspace today, please import them one by one by dragging them separately onto the workspace.

Now, go ahead and open the file of the participant clockwise ahead of you on the screen. (*Say present users' names: "For example: "X" will open "Y's" file,"Y" will open "Z's" etc.)*) You open the file by double-clicking on it. Now close the Word-application showing the file and look again at the Impromptu pie. A ring has appeared around the dot. This ring indicates that someone has opened or modified the file in some way. The color of the ring displays which user was the actor, meaning who opened or modified the file.

Now open the file of the participant counter-clockwise ahead of you. (*Say present users' names: "Swirl" will open "Adam's" file," John" will open "Swirl's" etc.)*) Write a word in the file, save it and close it. Look at the dot representing the file you just edited. Your color will be around the innermost ring around the dot, displaying that you were the last actor to touch the file. Note that there is no spacing between the dot and the innermost ring. The older event, of the user who opened the file before you, has now been pushed out to the second ring. Whenever a new event will occur on the file, the older event will be pushed one step out from the center. This means that the most recent event will always first be displayed as the innermost ring around the dot and as other events occur the "older" event will ripple towards the outside before disappearing altogether. Think of it as ripples in the water after you've thrown a rock into it. They will start at the center but start rippling out and eventually disappear. All rings also have a fade out time, meaning that after some time has passed they will disappear from view.

Now put the mouse over any dot. On the mouse over you will see a smaller pie displayed in the bottom right corner. This "history-pie" displays all the history over that specific file. The history pie has the same user orientation and pie-slices as the big pie. An arc in a participants slice represents an action in the past by that participant on the file. Building on the same idea

as for the rings, the arcs closer to the center represents more recent events whereas arcs at the edge of the pie are events of a more distant past. Think of it as the rings that indicate the growth pattern in a tree, the more recent the closer to the center the arcs are. (*Look at session specific files, describe how they are different, what they tell*).

Now take a look at the big pie again. Notice the edges of each participants pie slice. Each edge is colored in that participants color. The width of the edge displays the activity of that participant over the last five minutes. Opening a file and writing to a file are the two events that are considered activity. The activity of a participant is always measured relative to the activity of other participants. If all participants have been equally active they will have equally wide edges. A thin edge displays a relatively inactive participant and a thick edge a relatively active participant. The edges will grow and shrink continuously during the session. (*Make one user do a lot of action, point out how activity of this user will grow whereas the others will shrink, give session specific examples*).

Are there any questions?

Ok, let's move on the first task.

Appendix 4

Task Scenario 1 - Cards

Welcome to the game!

In this first task you will look at playing cards, add their values and try to receive the highest sum. An "Ace" is considered 1 point, "Two" is 2 points, a "Three" 3 points etc. A Jack, Queen and King are each worth 10 points. You have been given 5 different TIF images, already residing in the folder named "Cards_1" on your computer's desktop area. They are named: "Card_A", "Card_B" etc. Each image shows a playing card of a specific value.

When the Test Administrator tells you to start, you can view any of these images by opening the folder (the images you view at this point will not count as part of the game, you will simply view the value of your cards before sharing them). Chose 3-5 images and then place them in the Impromptu workspace, by dragging them onto it. Place the files so that other test participants can read. After all participants have shared their files the Test Administrator will tell you to go ahead and open 3 files, “turn” the images over to see their value. Choose any 3 files from the workspace, either belonging to you or to some other participant. Any file, your own or someone else’s that you choose to open will count as a turned card. Leave the window showing the card open, do not close it. Remember, any image opened will count as “turned card” and can not be un-turned by closing the image. Only the three turned cards, opened via the pie workspace, will count to your total, not any other cards residing in your pie-slice or on your desktop.

After the task is completed you will be asked some questions concerning what went on during the task. For example you will be asked questions such as which files were opened by which participants and which was the most opened file.

When task is completed your test administrator will note down your 3 opened cards and we will announce the winner.

Are there any questions?

Good luck!

Appendix 5

Task Scenario 2 - Lists

In this second task you will open and edit files containing lists of all different categories. You have been given 5 list files, residing in the folder named “Lists” on your computer’s desktop area. They have names of different categories such as: “Presidents”, “Mountains” etc. When the Test Administrator tells you to start, import these files onto the Impromptu workspace. Place the files so that other test participants can open and edit them.

After you have shared these 5 files, go ahead and open any files you like, yours and other participants’ files. Your task will be to add on items to the lists of these files. For example you may encounter a list looking like this:

Banana
Apple
Kiwi

A possible continuation for this list would be to add for example “Orange”. When you add one or more items to a list, put your username and the word “begins” at one line and thereafter

write all your items. Finish by writing your username followed by “ends”. This will keep track of who added which items. It will look like this:

Banana
Apple
Kiwi
Carolina begins:
 Orange
 Pineapple
Carolina ends.

Before opening a new file, save the current file by hitting “Save” and thereafter close it. Do not choose “Save As” to save it locally to your computer. You will not save any files locally on your desktop. You may only keep one file open at the time, and it is important that you close the file before you open a new one. If you open a file and get the message that it is a read only file this means, in this case, that someone else has the file open and is editing it. In this case close the file and try another one.

The more items you add and the more lists you edit, the better ranking you will get in the game. You may only edit one file at a time. Multiple people can read/open a file at one time, but only one person can edit it at a time. This task must be completed within 5 minutes.

Are there any questions?

Good luck!

Appendix 6

Posttask Questionnaire: Cards

<i>SWIRL Research Use Only</i>								
<i>Session No:</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>
<i>Participant No:</i>		<i>1</i>				<i>2</i>		

1. Which card would you guess was opened by the most participants? _____

2. Which cards were opened by the participant named _____ ? (Answer with: “Card_X” etc)_____

3. A Did you see other people opening your files? _____

3. B If you answered Yes to 3A, how did you see this? _____

4. A Did you notice if there was any card that was opened by more than one person, which card was it? (Pick one if there were multiple) _____

4. B Who was the first participant to open the card from question 4.A? _____

4. C How did you conclude the answer for 4.A and 4.B? _____

Thank you!

Please return this questionnaire to the observer sitting next to you.

Appendix 7

Posttask Questionnaire: Lists

<i>SWIRL Research Use Only</i>								
<i>Session No:</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>
<i>Participant No:</i>	<i>1</i>	<i>2</i>						

1. Which files were NOT opened?

2. Name three of the most opened files: _____

3. Who last accessed file ? _____

4. Who was first to touch file ? _____

5. Which participants have touched file ? _____

6.A Do you think that there was any file opened by the same user twice?

Yes

No

6.B If you answered Yes on 6.A, what was the name of the file? (If multiple, chose one)

6.C Which participant had opened the file in 6.B twice?

7. Did you have any strategy for which files you chose to open? _____

Thank you!

Please return this questionnaire to the observer sitting next to you.

Appendix 8

Posttest Questionnaire

Instructions: Please choose the words that best describe your experience using the software. Circle one of the given alternatives for each statement

<i>SWIRL Research Use Only</i>								
<i>Session No:</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>
<i>Participant No:</i>		<i>1</i>					<i>2</i>	

1. The rings let me see what other participants were doing:

strongly disagree disagree neutral agree strongly agree

2. I could see the history of a file:

strongly disagree disagree neutral agree strongly agree

3. I could see what other participants were doing:

strongly disagree disagree neutral agree strongly agree

4. I could see which participant was being most active at the moment:

strongly disagree disagree neutral agree strongly agree

5. I knew when other people were interacting with my files:

strongly disagree disagree neutral agree strongly agree

6. I knew that other people could see what I was doing:

strongly disagree disagree neutral agree strongly agree

7. During the 2 card tasks, the rings around the dots helped me with better knowing the history of the files:

strongly disagree disagree neutral agree strongly agree

Thank you!

Please return this questionnaire to the observer sitting next to you. The observer would like to talk to you about your experiences now.

Appendix 9

Debriefing Interview

For the use of Test Monitors only

Instructions:

- Make the subject comfortable

<i>SWIRL Research Use Only</i>							
<i>Session No:</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7 8</i>
<i>Participant No:</i>	<i>1</i>	<i>2</i>					

- Tell them that you value their honest opinion
- Look over their questionnaire for things marked “highly” or “very”.
- Focus on difficulties and problems.
- Ask for specifics and clarifications.

1. What did you like best about using this History Visualization Impromptu? Why?

2. What did you like least about using this History Visualization Impromptu? Why?

3. (Ask if appropriate) In your questionnaire you rated _____ as being something you strongly agreed with. Could you tell me more about that? (ASK FOR EXAMPLES, HOW DID YOU USE IT, HOW DID IT HELP YOU CHOSE FILES ETC.)

4. (Ask if appropriate) In your questionnaire you rated _____ as being something you strongly disagreed with. Could you tell me more about that? (HOW WAS IT CONFUSING, WHEN DID YOU NEED TO USE IT/DO THAT, WHAT WOULD YOU’VE LIKED IT TO HAVE DONE ETC.)

5. In the Cards-task, how did you decide what files to open?

6. Did you look at the rings around the dots for guidance of what cards other participants had chosen or for what lists other participants had opened?

7. Would you want to see more of what other people are doing? (Could you specify that?)

8. I will now show you a screenshot from another session. In this session there are three users. One user joined the session just 30 seconds ago. The other two had been in the session for longer time. Which user would you guess is the user that has just joined the session? <**Show screenshot No 1**> What made you guess at that user?

9. I will now show you two different screenshots. They each are taken from session of two different working situations. One screenshot is from a session with a teacher and two students. The students are working separately on their own papers for a project. The teacher is helping them on the right track with adding comments in their documents. The teacher is also sharing templates of the structure of different papers that the students can look at.

The second screenshot is from a session with 3 users. They are doing the same Lists task that you just did. They are competing to add items to lists of as many files as possible.

<Show screenshot No 2 and No 3> Which screenshot would you guess is the from the session with a teacher and two students? What made you guess at that one?

Which screenshot would you guess is the one from the session with the users doing the Lists task? What made you guess at that?

Appendix 10

Summary of correct answers Post task Questionnaires

	User 11	User 12	User 21	User 22	User 31	User 32
Used the History Pie	1	1				1
A = Did not use, but talked about the History Pie					1	
B = Did not use or talk about History Pie			1	1		
A + B			1	1	1	
Used/looked at Activity Wear (border)		1	1		1	1
A = Did not use, but talked about Activity Wear						
B = Did not use or talk specifically about Activity Wear, but could have used it	1			1		
A + B	1			1		
All tasks:						
Answers - Cards 1 task (without rings) 4p max	4	4	3	3	1	2
Answers - Cards 2 task (with rings) 4p max	2	4	4	3	4	4
Answers - Lists task 7p max	7	5	4	3	2	5
Answers - Cards 1 task (with rings) 4p max						
Answers - Cards 2 task (without rings) 4p max						
Answers - Lists task 7p max						
Total answers Cards task without rings 4p max	4	4	3	3	1	2
Total answers Cards task with rings 4p max	2	4	4	3	4	4
List task: 7p max						
Used the History Pie	7	5				5
A = Did not use, but talked about the History Pie					2	
B = Did not use or talk about History Pie			4	3		
A + B: did not use history pie			4	3	2	
Mean of all users for list task	7	5	4	3	2	5
Screenshot Q's from Interview 1p max						
Q 8. screenshot 1, Lina newest arrived	1	1	1	1	1	1
Looked at Activity	0	1	1	0	1	1
Q 9a screenshot no 2 (Lina is the teacher)	1	1	1	1	1	1
Q 9b screenshot no 3	1	1	1	1	1	1

1=correct, 0=wrong

With rings 4p max	2	4	4	3	4	4
Standard deviation (STDEV)	0,793					
Variance (STDEV ^ 2)	0,6288					
Without rings 4p max	4	4	3	3	1	2
Standard deviation (STDEV)	1,3114					
Variance (STDEV ^ 2)	1,7198					
Using History pie 7p max	7	5				5
Standard deviation (STDEV)	0,7559					
Variance (STDEV ^ 2)	0,5714					
NOT using History pie 7p max			4	3	2	
Standard deviation (STDEV)	2,3022					
Variance (STDEV ^ 2)	5,3002					
	User	User	User	User	User	User
	11	12	21	22	31	32

continues below:

User 41	User 42	User 51	User 52	User 61	User 62	Total	Maximum	Percentage (Average)	Mean
	1	1	1	1		7	12	58%	
1					1	3	12	25%	
						2	12	17%	
1					1	5	12	42%	
1	1				1	7	12	58%	
		1	1			2	12	17%	
				1		3	12	25%	
		1	1	1		5	12	42%	
						17	24	71%	2,8333333
						21	24	88%	3,5
						26	42	62%	4,3333333
3	2	4	4	4	3	20	24	83%	3,3333333
2	2	4	4	2	0	14	24	58%	2,3333333
7	5	6	6	6	1	31	42	74%	5,1666667
2	2	4	4	2	0	31	48	65%	2,5833333
3	2	4	4	4	3	41	48	85%	3,4166667
	5	6	6	6		40	49	82%	5,7142857
7					1	10	21	48%	3,3333333
						7	14	50%	3,5
7					1	17	35	49%	3,4

	7	5	6	6	6	1		57	84	68%	4,75
1	0	1	1	1	1			11	12	92%	
1	1	0	0	0	1			7	12	58%	
1	0	1	1	1	1			11	12	92%	
1	0	1	1	1	1			11	12	92%	

STDEV

	3	2	4	4	4	3	0,793	41	48	85%
	2	2	4	4	2	0	1,3114	31	48	65%
		5	6	6	6		0,7559	40	49	82%
	7					1	2,3022	17	35	49%

User 41 User 42 User 51 User 52 User 61 User 62

Appendix 11

Summary of interviews and session notes

	User 11 ok	User 12 ok	User 21 ok	User 22 ok	User 31 ok	User 32 ok
Rings						
Used the rings	1	1	1	1	1	1
Liked the rings	1	1	1	1	1	1
Used rings as guidance of which cards other						

users had opened	1	1		1	1	1
Used rings when reasoning about screenshot 1	1			1	1	1
Used rings when reasoning about screenshot 2 & 3	1	1	1	1	1	1
Looked at screen when answering Cards & List posttask questions	1	1	1	1	1	1
Looked at rings to see which lists had been opened	1	1	1	1		
Liked the fade out, cool		1				
Waited for other users to open cards first	1	1		1	1	1
Used memory, for answering posttask Q's, since rings had faded	1					
Did not understand rings correctly						
History pie						
Used the History pie	1	1				1
Said that liked the History pie	1	1			1	1
Used the History pie when answering Lists posttask questions	1	1				1
Said that looked at pie when rings had faded, liked that it was persistent	1	1				1
Activity wear						
Used activity wear when reasoning about screenshot 1			1		1	1
Said: activity measure is low		1				
Said: user X has least amount of activities						
Used activity wear when reasoning about what happened during the tasks, had looked at it						
Did not like						
Small inner ring, acting different, not fading	1					
Fade out too quickly on rings	1	1				
Hard to see a file opened by its owner	1	1				1
Colors are a problem		1				
Disappearing, fading rings			1			1
Slow			1	1	1	1
Lag between mouseclick and action on screen		1				
Had to wait for history arcs to draw, slow, could mistake as an empty history pie	1					1
Color yellow difficult to see				1		
Could only see afterwards what others had done, not while they were doing it				1		
The white ring around a dot confusing						1
Thought outer ring was more eye catching, than the inner most recent ring						
Word window covered workspace, could not see at all time what was happening						
Posttest Questionnaire 5 max						
1. Ring let me see what other users were doing	4	4	4	4	4	4

2. I could see the history of a file	5	5	3	4	5	5
3. I could see what other users were doing	5	5	4	2	4	5
4. I could see which user was most active at the moment	4	4	2	4	4	3
5. I knew when other people were interacting with my files	5	5	4	4	5	5
6. I knew that other people could see what I was doing	5	5	4	4	4	2
7. During the 2 card tasks, the rings around dots helped me with better knowing history of a file	3	3	4	4	5	4

1 = strongly disagree
2 = disagree
3 = neutral
4 = agree
5 = strongly agree

Would like to see more...

If a file is locked, opened by anyone (Lists task)	1				1
See diff between read and write event (open/edit)			1		1
See diff in history pie betw read and write					
Display percentage of size change of the file		1		1	
See concurrent action, while editing, not just know it afterwards (when file is saved)					
How long a user was "in a file", ie having it open					

	User 11	User 12	User 21	User 22	User 31	User 32
User	Kari	Paul	David	Paul	Kari	Paul
Observer						

continues below:

User 41	User 42	User 51	User 52	User 61	User 62	Total	Maximum	Percentage
ok	ok	ok	ok	ok	ok			
1	1	1	1	1	1	12	12	100%
1	1	1	1	1	1	12	12	100%
1	1	1		1		9	12	75%
1	1		1	1	1	9	12	75%
1	1	1	1	1	1	12	12	100%
1	1	1	1	1	1	12	12	100%
1	1	1				7	12	58%

1		1		1		8	12	67%	
1			1			3	12	25%	
	1					1	12	8%	
	1	1	1	1		7	12	58%	
	1	1				6	12	50%	
	1	1	1	1		7	12	58%	
	1					4	12	33%	
1					1	5	12	42%	
		1				1	12	8%	
						1	12	8%	
	1					1	12	8%	
1				1	1	1	12	8%	
						5	12	42%	
						3	12	25%	
						1	12	8%	
						2	12	17%	
			1			5	12	42%	
						1	12	8%	
						2	12	17%	
						1	12	8%	
1						2	12	17%	
						1	12	8%	
		1				1	12	8%	
				1		1	12	8%	
									Mean
5	4	4	4	5	4	50	60	83%	4,166667
5	5	4	4	5	4	54	60	90%	4,5
3	2	3	4	5	4	46	60	77%	3,833333
4	5	4	4	5	4	47	60	78%	3,916667
5	5	5	4	4	4	55	60	92%	4,583333
4	5	2	4	5	4	48	60	80%	4
5	4	5	4	5	4	50	60	83%	4,166667

				2	12	17%
	1	1	1	5	12	42%
			1	1	12	8%
	1			3	12	25%
1			1	2	12	17%
	1			1	12	8%

User 41	User 42	User 51	User 52	User 61	User 62
Kari	Paul	Kari	Paul	Kari	David

Appendix 12

Summary of Pretest Questionnaire

	User 11	User 12	User 21	User 22	User 31	User 32
1. Age	28	23	25	24	33	29
2. First language	Romanian	English	English	English	Japanese	Bengali
3. Years of programming experience	15	3	7	6	24	4
4. Known programming languages:						
Java	X	X	X	X	X	X
C, C++	X		X		X	
HTML	X	X		X	X	X
Other				R, PHP	See *	
5. Have done user interface design	Yes	Yes	No	Yes	Yes	Yes
6. Years of Computer Studies done	9	5	6	2	15	6

* = Ruby, Pearl, Python, assemblers, objectiv C

continues below:

User 41	User 42	User 51	User 52	User 61	User 62	Mean
21	19	25	25	27	24	25,25
English	English	Chinese/ mandarin	English	English	English	
4	2	7,5	7	0	1	6,708333
X	X	X	X			
X	X	X	X		X	
VB.NET		Scheme/ Lisp	?			
Yes	Yes	Yes	Yes	No	No	
4	2	7	7	0	1	5,333333

