

MessyBoard: Lowering the Cost of Communication and Making it More Enjoyable

Adam M. Fass

CMU-CS-05-130

May 2nd, 2005

School of Computer Science
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee
Randy Pausch, Chair
Jodi Forlizzi
Jessica Hodgins
Terry Winograd, Stanford University

Submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy

Copyright © 2005 Adam M. Fass. All rights reserved.

This research was sponsored by various grants from the Office of Naval Research (N000140310261, N00140210439) and the National Science Foundation (IIS-0329090, IIS-0121629, IIS-9812012). The views and conclusions contained within this document are those of the author and should not be interpreted as representing the official policies or endorsements, either expressed or implied of any sponsoring party or of the U.S. Government.

Keywords: Asynchronous collaboration, computer supported cooperative work, group and organization Interfaces, large displays, MessyBoard, screen saver, web-based interaction.

Abstract

Large projects require multiple people to work together. Often, these people do not communicate as much as they should. One reason is that communication takes time and effort and causes interruptions. Another reason is that work related communication can be unpleasant. The goals of this research are to reduce the costs of communication and to make it more enjoyable in order to improve collaboration.

In order to achieve these goals I have created MessyBoard, a communication medium based on the metaphor of a two-dimensional bulletin board. This medium allows people to easily use mixed media and spatial relationships to communicate their ideas. I use large public displays and screen savers to ensure that people see it naturally without being interrupted unexpectedly or having to explicitly set aside time. MessyBoard runs as a Java applet in a web browser so that people can begin using it immediately with any computer.

I have observed 191 MessyBoard spaces used by groups at Carnegie Mellon University and by anonymous users over on the Internet. I collected observations through a combination of automatic logging, interviews and ethnographic observation. I have observed that larger groups (25 or more members) are more likely to adopt MessyBoard than smaller groups (15 or fewer members). Larger groups use MessyBoard for a mixture of playful and goal-directed activities. The small groups that use MessyBoard the most tend to use it as a file sharing tool.

Acknowledgements

My experience as a graduate student has been so much more than the production of this dissertation, and I feel that I have grown and changed in profound ways over the course of this journey. Many people have helped and influenced me and I could not possibly name them all. Here I thank the people who have had the greatest impact on my experience as a graduate student.

Thanks to Randy Pausch for a uniquely well-rounded educational experience. Most advisors will critique their students' research in order to make it better, and Randy was no slouch in that department. However, it is a rare advisor who teaches students to give a flashy demo, make an elevator pitch, secure funding and lead and inspire a group of people. Randy taught me many life lessons that are relevant both in academia and in the "real world".

I'd like to thank the other members of my thesis committee, Jodi Forlizzi, Jessica Hodgins and Terry Winograd, for their helpful suggestions and guidance. Thanks also to Mark Fichman, Bob Kraut and Sara Kiesler for their research advice.

Thanks to Desney Tan, Andrew Faulring, Caitlin Kelleher, Dennis Cosgrove, Jason Pratt, Mike Darga, Tiffany Pomarico and all of the former members of the Stage3 research group for their research advice, stimulating discussions, and for doing some of my grunt work. I cannot overstate the value of being able to talk to experts in research, programming, art and design just by walking across the room. Thanks also to Peter Scupelli for lending his time and expertise in order to help with my early user observations.

Thanks to Dennis Proffitt and his students at the University of Virginia for an enjoyable long-distance collaboration and a great summer internship. The members of the Proffitt Perception lab taught me a lot about psychology and science in general, and we shared a lot of great times together.

My internships were a crucial part of my education, allowing me to view the world from different perspectives. Thanks to Eric Dittert, Eric Bier and all of the folks at Right

Hemisphere for taking me in, making me feel at home and trusting me with great responsibilities.

Thanks to Russ Miller, Helene Kershner, Davin Milun, and everyone else who taught me and looked out for me as an undergraduate at the State University of New York at Buffalo. My time there prepared me well for things to come.

Thanks to the National Science Foundation for funding my work through a scholarship. Thanks also to the Department of Defense Advanced Research Projects Agency, the Office of Naval Research, and the National Science Foundation for their generous grants that have allowed me to pursue my research goals.

I am blessed with two of the most selfless and supportive parents that ever existed, and I cannot begin to express my gratitude for their help and advice throughout my entire life. As pillars of their community and all-around mensches, they continue to serve as shining examples of everything that I want to be. Thanks also to my sister Tracy, my brother Jon and the rest of my family for all of their love and support.

Thanks to Matt Coton for his lifelong companionship. I am incredibly fortunate to have a friend who has stuck by me through thick and thin since elementary school, and I treasure the strength and depth of our friendship.

Thanks to James Ezick for his friendship, guidance and understanding (through experience) of the trials and tribulations of graduate school. It was great to be able to talk to someone who really understands.

Finally, and most importantly, thanks to my wife Megan for her love, support and companionship. She is the reason I keep working and the reason that I know when to stop working and come home. Though we met just a short time ago, I cannot imagine my life without her. I love her deeply and I look forward to a future filled with happiness.

Dedication

To my newborn son, Benjamin Scott Fass. I do not know what kind of person you will become, and I cannot wait to find out.

Table of Contents

Abstract	iii
Acknowledgements	iv
Dedication	vi
Table of Contents	vii
List of Figures and Tables	xi

Introduction	18
1.1 Motivation.....	18
1.2 The Medium.....	21
1.3 Natural, Low-Cost Integration Into People's Lives.....	22
1.4 Observations and Results	24
1.5 Thesis Statement	26
1.6 Contributions	26
1.7 Dissertation Organization.....	27

MessyBoard Description and Design Rationale	28
2.1 A Networked Bulletin Board.....	28
2.2 Startup Cost.....	29
2.2.1 Reducing the Startup Cost: Creating a MessyBoard Space.....	32
2.2.2 Reducing the Startup Cost: Accessing a MessyBoard Space.....	34
2.3 Authoring Cost.....	34
2.3.1 A Spatial Medium Reduces Authoring Cost.....	35
2.3.2 The MessyBoard Interface Reduces Authoring Cost.....	38
2.3.3 Identification	40
2.4 Receiving Cost.....	41
2.4.1 Finite Space Reduces Receiving Cost	41
2.4.2 History	42
2.4.3 A Poverty of Attention	43
2.4.4 Salvaging Attention with Projectors and Screen Savers	45
2.5 A Spatial Medium Makes Communication Enjoyable	48

Software Design	50
3.1 Architecture Overview	50
3.2 The GUI Layer.....	51
3.3 The Database Layer	53
3.3.1 Immediate feedback	54
3.3.2 Real-time synchronization.....	55
3.3.3 Requested and Confirmed States	55
3.3.4 Locks and Simultaneous Operations.....	59
3.3.5 Real-time Updates and Un-guaranteed Requests.....	61

3.3.6	Large Data Objects.....	62
3.3.7	History	63
3.3.7.1	History and the GUI	65
3.4	The Network Layer	67
3.5	Implementation and Deployment	67
3.5.1	Technical Hurdles to Reducing Startup Cost.....	68
3.6	The MessyBoard Web Site.....	70
3.6.1	Creating a MessyBoard Space.....	70
3.6.2	Scalability	72
3.6.3	Practical Problems with a Monolithic MessyBoard Server.....	75
Related Work		77
4.1	Shared Drawing Applications	77
4.2	Shared Text Editors.....	77
4.3	Chat (Instant Messaging)	78
4.4	Large Public Displays	78
4.5	Screen Savers	79
4.6	Lightweight Asynchronous Collaboration Tools.....	80
4.7	Closely Related Systems	80
4.7.1	TeamWave Workplace	82
4.7.2	MuSwikis	84
4.7.3	Designers' Outpost.....	85
4.7.4	Notification Collage	86
4.7.5	iCom.....	88
4.7.6	What's Happening.....	89
4.7.7	ScanBoard	90
4.7.8	Semi-Public Displays	91
4.7.9	Groove Pinboard	92
4.7.10	Kansas	93
4.7.11	Netomat.....	94
4.8	Physical Corkboards, Whiteboards and Other Public Artifacts	95
4.8.1	Trauma Center Operating Room Boards	95
4.8.2	Personal Office Whiteboards and Corkboards.....	96
4.8.3	Public Artifacts for Collaboration	97
4.8.4	Family Bulletin Boards	98
4.8.5	Observations of Campus Bulletin Boards.....	98
4.8.5.1	Public Boards	99
4.8.5.2	Departmental Boards	100
4.8.5.3	Public Whiteboards	101
4.8.5.4	Group Boards	102
4.8.5.5	Locked Boards.....	102
4.8.5.6	Dormitory boards.....	103
4.8.6	Discussion	104
4.9	Grounding.....	105
4.10	Awareness.....	107
4.10.1	The Awareness Literature.....	108

4.10.2	Media Spaces	110
4.10.3	Event Notification Services	111
4.10.4	MessyBoard as an Awareness System	112
4.11	Groupware Adoption	113
4.11.1	Work/Benefit disparity	114
4.11.2	Critical mass	115
4.11.3	Social and Political Factors	116
Observations of MessyBoard Use		118
5.1	Research Questions	118
5.2	Methodology Overview	119
5.3	Prototype Deployment	119
5.3.1	Methods	120
5.3.2	Observations	123
5.3.2.1	Differences between research groups	123
5.3.2.2	Projection	124
5.3.2.3	Awareness: Who's Around	125
5.3.2.4	MessyBoard is Good for Scheduling a Meeting	125
5.3.2.5	Authorship and Politeness	126
5.3.2.6	Simple Games Attract Attention	127
5.3.2.7	Instant "War Room"	128
5.3.3	Lessons learned	129
5.4	Automation and Scripting	131
5.5	Java MessyBoard Carnegie Mellon Deployment	132
5.5.1	Large Groups	133
5.5.1.1	Methods	133
5.5.1.2	Group A Observations	138
5.5.1.3	Group B Observations	142
5.5.1.4	Group C Observations	149
5.5.1.5	Discussion of Large Group Observations	154
5.5.2	Small Groups	156
5.5.2.1	Methods	157
5.5.2.2	Student Project Group Observations	158
5.5.2.3	Research Group Observations	171
5.5.2.4	Discussion of Small Group Observations	175
5.6	Other Observations of MessyBoard Use	177
5.6.1	Broadcast from Few to Many	177
5.6.2	Place-based Use	178
5.6.3	Administrative Assistants	178
5.6.4	Academic Conferences	180
5.6.4.1	Methods	181
5.6.4.2	Observations	184
5.6.5	Short-term Groups	186
5.7	Internet Deployment	187
5.7.1	Methods	188
5.7.2	Observations	191

5.7.2.1	Why People Used MessyBoard	191
5.7.2.2	How Long People Used MessyBoard	195
5.7.2.3	How Many People Used Each MessyBoard Space	197
5.7.2.4	Real-time Collaboration.....	198
5.7.2.5	Which Features Were Used	199
5.7.2.6	Screen Saver Use.....	201
5.7.2.7	Interesting Spaces.....	201
5.7.3	Discussion and Conclusion.....	206
Conclusion and Future Work		208
6.1	Summary.....	208
6.2	Principles of Communication.....	209
6.2.1	Applying Principles to Results	211
6.2.2	Applying Principles to the Design of Communication Tools	213
6.2.3	Communication Principles and Rational Decisions.....	214
6.3	Future Work	215
6.3.1	Interface Improvements.....	215
6.3.2	Integration with a Repository	215
Appendix A: Third Party Libraries and Components		218
Appendix B: Interview Questions for Bulletin Boards		219
Appendix C: Online Survey Items		220
Appendix D: Interview Questions for Group A		230
Appendix E: Interview Questions for Course Alpha		231
Bibliography		232

List of Figures and Tables

Introduction

Figure 1.1. A mixed media conversation on MessyBoard about the design of a drag and drop programming interface.	23
Figure 1.2. A collaborative humorous collage on MessyBoard.	23
Figure 1.3. A large group uses MessyBoard to choose offices.	25
Figure 1.4. A small group uses MessyBoard as an annotated file repository.	25

MessyBoard Description and Design Rationale

Figure 2.1. The MessyBoard interface, running in a window as a stand-alone application.	28
Figure 2.2. The MessyBoard Java applet running in Microsoft Internet Explorer (left) and Mozilla Firefox (right). The applet runs in a variety of web browsers on the Windows, Macintosh and Linux platforms.	29
Figure 2.3. A user must fill out several forms to create a Yahoo! group.	30
Figure 2.4. To create a new MessyBoard space, a user types a name, clicks a button and then fills out a consent form. On completing the consent form, the user is taken directly to the MessyBoard space. Other users will see a similar consent form the first time they view the space.	33
Figure 2.5. A design conversation in a prototype version of MessyBoard. The authors of the note that reads "how about this?" and the thought bubble that reads "I am the new for-loop" both use proximity and overlap to establish a referent.	36
Figure 2.6. MessyBoard users arrange files in clusters on top of notes to indicate that there are two separate groups.	36
Figure 2.7. A MessyBoard user places comments over the pictures to which they refer.	37
Figure 2.8. Users place notes over a floor plan in order to assign occupants to individual rooms.	37
Figure 2.9. The MessyBoard main menu.	39
Figure 2.10. A note with the author's screen name and picture.	41
Figure 2.12. Common communication methods and the ways that they guarantee that receivers will pay attention.	45
Figure 2.13. MessyBoard projected on the wall in a shared work space.	47
Figure 2.14. The MessyBoard screen saver.	47

Figure 2.15. Two groups of users in different locations use an early MessyBoard prototype to play a game of pong.	48
Figure 2.16. MessyBoard users sign up for a table tennis tournament (left) and for their next lab meeting (right).	49

Software Design

Figure 3.1. The MessyBoard architecture comprises three layers: GUI, database and network. The database and network layers each have a client and a server component.	51
Figure 3.2. The MessyBoard space is represented by a MessyArea object. Each object that the user sees on MessyBoard is represented by a ViewController object which specifies the rendering and interactive behavior. A Model object encapsulates the shared data for each object and stores it as properties of a database element.	52
Figure 3.3. When a note is first created, the element is contained in the requested state for the client that created it and a creation request is sent to the server. The server processes the message, sends an acknowledgement to the client that requested the creation, and sends a broadcast to all other clients. When the clients receive these messages they add the element to their confirmed states.	57
Figure 3.4. When the text of a note is changed, the text property is changed in the requested state for the client that initiates the change and a property change request is sent to the server. The server processes the message, sends an acknowledgement to the client that requested the change, and sends a broadcast to all other clients. When the clients receive these messages they change the property value in their confirmed states.	58
Figure 3.5. Two users simultaneously attempt to drag a note in different directions.	60
Figure 3.6. Database clients use a separate stream connection to upload large data objects to the server. The database server writes the data to a file and sets a property in the database to a URL. The GUI layer can then use the URL to download the file via HTTP.	64
Figure 3.7. When the user clicks on the history slider, the database client requests history data from the server to construct the past state.	66
Figure 3.8. The Java run-time environment displays an unusual and alarming dialog for applications and applets that request unrestricted access to the hard drive, even if they are signed with a digital certificate from a major certificate authority.	69
Figure 3.9. A web server launches Python scripts that interact with server managers and generate dynamic HTML content in order to allow the user to create a new MessyBoard space and sign a consent form.	71

Figure 3.10. A router makes two physical computers appear as one from the outside and it directs traffic to the appropriate computer based on the destination port number.....	74
--	----

Related Work

Figure 4.1. MessyBoard shares some features in common with several similar systems but its combination of features is unique.....	81
Figure 4.2. Teamwave Workspace. Taken from http://www.markroseman.com/teamwave/overview_interface.gif	82
Figure 4.3. A Muswiki. Taken from http://www.cc.gatech.edu/~lex/muswiki/	84
Figure 4.4. Designers' Outpost. Taken from talk slides at http://guir.cs.berkeley.edu/projects/outpost/	85
Figure 4.5. Notification Collage. Image taken from http://grouplab.cpsc.ucalgary.ca/software/notification_collage/	86
Figure 4.6. iCom. Images taken from http://web.media.mit.edu/~stefan/hc/projects/icom/	88
Figure 4.7. A collage generated by What's Happening. Image taken from http://www.cc.gatech.edu/gvu/ii/community/	89
Figure 4.8. Scanboard. Image taken from (Hindus et al., 2001).	90
Figure 4.9. Semi-Public Displays. Image taken from http://www.cc.gatech.edu/fce/ecl/projects/semiPublic/	91
Figure 4.10. The Groove Pinboard tool. Image taken from http://www.cabecal.com/Products/PinBoard/pinboardscreenshot.gif	92
Figure 4.11. Kansas. Image taken from http://research.sun.com/ics/kansas.html	93
Figure 4.12. Netomat. Image taken from http://www.netomat.com/	94
Figure 4.13. Two public boards.	100
Figure 4.14. Three departmental boards.....	101
Figure 4.15. Three public whiteboards.....	101
Figure 4.16. Three group boards.	102
Figure 4.17. Three locked boards.....	103
Figure 4.18. Two dormitory boards.....	103
Figure 4.19. The floor plan provides common ground for interpreting the purple note about the door.....	106
Figure 4.20. The screen shots provide common ground for interpreting the surrounding comments.	106

Observations of MessyBoard Use

Figure 5.1. Summary of MessyBoard deployments and methodologies for observation.....	120
Figure 5.2. The Info Cockpits MessyBoard space.....	122
Figure 5.3. The Alice MessyBoard space.....	122
Figure 5.4. The Stage3 lab with the Alice (left) and Info Cockpits (right) MessyBoard spaces projected on the walls.	124
Figure 5.5. Users schedule a teleconference on the Info Cockpits MessyBoard using a single note.....	125
Figure 5.6. Users schedule a teleconference on the Info Cockpits MessyBoard by placing notes over an image pasted from a spreadsheet.....	126
Figure 5.7. A technical conversation on the Info Cockpits MessyBoard space.	127
Figure 5.8. Users play a game of Pong in the Info Cockpits MessyBoard space.	128
Figure 5.9. The Info Cockpits MessyBoard spaces becomes a dedicated project space for creating a video demonstration on short notice.	129
Figure 5.10. MessyBoard usage and events for group A.....	139
Figure 5.11. Group A's MessyBoard space.	139
Figure 5.12. Two pictures on Group A's MessyBoard space. Both are related to the group's research, at least tangentially, and both are also posted for the enjoyment of the group members.	141
Figure 5.13. Group A uses MessyBoard to sign up for a table tennis tournament (at left).	141
Figure 5.14. Group A uses MessyBoard to sign up to give talks at their upcoming retreat (at left).....	142
Figure 5.15. MessyBoard usage and events for group B.	143
Figure 5.16. Group B uses their MessyBoard space to post humorous pictures and announce social events.....	144
Figure 5.17. Group B has a discussion about how to fairly assign people to offices in their new office space.....	145
Figure 5.18. Group B uses their MessyBoard space to select new offices. Fictitious names are used to maintain users' anonymity.	146
Figure 5.19. Members of Group B make jokes about the office selection process on MessyBoard. The pixelated images are pictures of two faculty members who were involved with the process, edited into a screen shot from a video game. The pictures and the phrase "all your	

space are belong to us" refer to a poorly translated video game that became a cult phenomenon on the Internet (Sega, 2005).	147
Figure 5.20. Group B uses a similar process to choose offices a second time.	148
Figure 5.21. MessyBoard usage and events for group C.	150
Figure 5.22. Group C's MessyBoard space.	150
Figure 5.23. Group C's MessyBoard space.	151
Figure 5.24. Group C uses MessyBoard to organize extracurricular activities.....	151
Figure 5.25. Group C posts humorous pictures in their MessyBoard space.	152
Figure 5.26. Group C coordinates the creation of a historical timeline.	153
Figure 5.27. MessyBoard usage for groups A, B and C.	155
Figure 5.28. Percentage of transactions from individual static IP addresses.	156
Figure 5.29. MessyBoard usage for groups D, E and F.	159
Figure 5.30. MessyBoard usage and events for group D.	160
Figure 5.31. MessyBoard usage and events for group E.	161
Figure 5.32. MessyBoard usage and events for group F.	162
Figure 5.33. Group D's MessyBoard space.	163
Figure 5.34. Group E's MessyBoard space.	164
Figure 5.35. Group F's MessyBoard space.	165
Figure 5.36. MessyBoard usage for students in course Beta.	167
Figure 5.37. Students in Course Beta use MessyBoard to introduce themselves and to share contact information.	168
Figure 5.38. One of the groups in Course Beta uses MessyBoard to create an affinity diagram.	168
Figure 5.39. MessyBoard usage for two project groups that approached me and requested MessyBoard spaces.	171
Figure 5.40. MessyBoard usage for group G.	172
Figure 5.41. Group G uses MessyBoard to build a shared representation of the dates that they will be out of town.	173
Figure 5.42. MessyBoard usage for group H.	174
Figure 5.43. MessyBoard usage for group I.	175
Figure 5.44. MessyBoard usage for the small groups that used it the most.	176
Figure 5.45. MessyBoard projected opposite the registration table at UIST 2004.	182

Figure 5.46. The UIST 2004 MessyBoard space.....	185
Figure 5.47. The CSCW 2004 MessyBoard space.	185
Figure 5.48. MessyBoard usage for the UIST and CSCW conferences.....	186
Figure 5.49. MessyBoard spaces used by class project groups working on short-term projects.	187
Figure 5.50. The Google AdWords program displays a simple MessyBoard advertisement on the Google search results page.	189
Figure 5.51. Number of MessyBoard spaces created each day, page views for the main MessyBoard web page per day, and cumulative number of MessyBoard spaces created for each day of the public Internet deployment.	190
Figure 5.52. The number of clicks on the AdWords advertisement shown in Google's search results for each keyword. The total number of clicks is 135. Keywords marked with a * were disabled early on because too few users clicked on them relative to the number of times that the advertisement was shown.....	192
Figure 5.53. The number of spaces created for each intended purpose (54 in total). All of the spaces created for synchronous photo sharing appear to be created by the same person.	194
Figure 5.54. The number of spaces intended for group communication for each kind of group (33 in total). Three of the four spaces for entire classes appear to be for different kinds of discussion in the same class.	195
Figure 5.55. Cumulative number of MessyBoard spaces vs. minimum life span. For each life span on the X axis, the bar indicates how many spaces were used for at least that many weeks. All 124 spaces were used for zero or more weeks, 35 were used for one or more weeks, etc.	196
Figure 5.56. Cumulative number of MessyBoard spaces vs. minimum trimmed life span.	197
Figure 5.57. The cumulative number of spaces for each minimum number of users. All 124 spaces had at least one user, 65 spaces had two or more users, etc.	198
Figure 5.58. The average number of times that various features were used for the 20 MessyBoard spaces with a lifespan greater than four weeks.	200
Figure 5.59. The median number of times that various features were used for the 20 MessyBoard spaces with a lifespan greater than four weeks. (The median for odd-numbered data sets is the average of the two values in the center.)	201

Figure 5.60. Cumulative number of MessyBoard spaces vs. minimum life span for spaces that logged at least one connection from the screen saver.....	202
Figure 5.61. A couple uses MessyBoard to manage to-do lists and to send messages to each other. The image has been blurred to maintain their anonymity.	203
Figure 5.62. A group of female college students use MessyBoard to share comments and pictures. The images have been blurred to maintain their anonymity.	203
Figure 5.63. Members of a band use this space to chat and find a time to rehearse. The images have been blurred to maintain their anonymity.	204
Figure 5.64. A short-term project team uses MessyBoard to share files while creating an interactive entertainment experience.	205
Figure 5.65. A single user uses MessyBoard as an all-purpose storage space.	206

Chapter 1

Introduction

1.1 Motivation

Projects often demand the resources and attention of multiple people. Some projects, like the development of a large software system, require too much work for one person to do in the allotted time. Others might be too complex for a single person to understand every detail, such as planning a large event. Finally, some projects require a diverse set of skills that a single person is not likely to have. For example, designing a building requires artistic vision, engineering skill and knowledge of human factors.

People who are working together on a project must communicate with each other, and the act of communication incurs a cost in time and effort. Talking face-to-face or on the phone, writing an e-mail message, using an instant messaging program, and leaving a note on someone's desk all take time and effort. People need to set aside time every day to read e-mail messages and listen to voice mail messages. If people want to meet in person, they need to find a time when all of them are free and perhaps reserve a room. These costs can be extremely high for groups that are geographically distributed, but even when a group of people works in the same location the cost can be substantial.

Some costs are incurred when a person initiates communication, and I refer to these as **authoring costs**. For example, it takes time and effort to compose an e-mail message that precisely expresses complex thoughts.

Other costs are incurred when a person receives a communication, and I refer to these as **receiving costs**. Incoming communications contain information and consuming information incurs a cost. As Herb Simon notes:

What information consumes is rather obvious: it consumes the attention of its recipients. Hence a wealth of information creates a poverty of attention,

and a need to allocate that attention efficiently among the overabundance of information sources that might consume it. (Simon, 1971)

In general, two forms of receiving costs are common: **interruption** (as in an unexpected telephone call) and **polling** (as in checking e-mail or voice mail). An interruption takes time away from ongoing work and causes workers to forget things, requiring them to expend more total time to complete tasks (Cutrell, Czerwinski, & Horvitz, 2001). Polling requires workers to set aside time that could be spent on other tasks and to employ mechanisms to help them remember to poll regularly.

Many people enjoy communicating in a variety of situations, but work-related communication is often boring, inefficient and distracting. Many workers complain about having to read e-mail and attend meetings (Poole & DeSanctis, 1990; Walker, 2004).

All else being equal, if the cost of an activity is high and the activity is not enjoyable then people will not do that activity as often. For example, research findings suggest that long-distance collaborators communicate less frequently and are less productive than collocated collaborators because the cost of scheduling a meeting or teleconference is much higher than the cost of initiating a conversation with a colleague in the hallway (Kraut, Fish, Root, & Chalfonte, 1990). If a home computer is located in an out-of-the-way room, users may be less likely to send e-mail or check for new messages (Frohlich & Kraut, 2003). Greenberg and Roseman argue that the need to move data between applications on the same computer can be a high enough cost to discourage communication (Greenberg & Roseman, 1998). Taken together, these results and arguments suggest that there are many times when a worker might communicate but does not because the cost is too high.

The goals of the present research are to lower the cost of communication and make it more enjoyable. Lowering the cost of communication is important because if the cost is lower, people will be more likely to communicate and this will lead to more effective collaboration.

One might argue that lowering communication costs could lead to a communication glut that will lower productivity. For example, spontaneous conversations have low cost compared to pre-scheduled meetings, but the opportunity for spontaneous conversation may

lead workers to waste time around the water cooler. Research results refute this claim: increased physical proximity leads to increased productivity because workers can communicate at a lower cost by walking into each others' offices or meeting by chance in the hallway (Herbsleb & Mockus, 2003; Kraut et al., 1990). Interestingly, the collaborators themselves may view these interactions as unproductive, but they clearly contribute to the success of the collaboration, either by accomplishing work or by strengthening social bonds (Kraut et al., 1990). One could imagine a hypothetical medium that lowers the cost of communication below the cost of a spontaneous conversation in the hallway, and perhaps this medium would lead to a harmful communication glut, but the communication tool that I introduce does not reduce costs to this level.

One might also argue that lowering the cost and increasing the amount of communication will lead to "communication overload" for individuals. For example, sending a mass e-mail message has a lower cost than distributing a hard copy of a memo, perhaps leading to more broadcast announcements and in turn causing employees to spend precious time sifting through hundreds of irrelevant announcements. If e-mail does in fact hurt productivity in this way, it is because the medium reduces the authoring cost of a broadcast announcement without a comparable reduction in the receiving cost. In my work, I pay close attention to both the authoring cost and the receiving cost of communication.

Making communication more enjoyable is important for three reasons. First, if it is more enjoyable then people are more likely to engage in work-related communication. Second, people are more likely to engage in casual communication and research has shown that this can lead to increased productivity, so long as the communication is not overly distracting (Kraut et al., 1990). Third, enjoyment is important in its own right. It is not my goal to encourage enjoyment at the expense of productivity. However, all else being equal, more pleasure in a person's life is desirable. A study by Morkes et al. suggests that humor can make a collaborative task more enjoyable without a loss in efficiency (Morkes, Kernal, & Nass, 1998).

This research focuses on improving communication between relatively small groups of people (groups of less than one hundred members, as opposed to entire organizations)

who work together and trust one another. This work does not address communication for entire organizations, nor does it address issues brought about by malicious users such as security and privacy.

My approach to lowering the cost of communication and making it more enjoyable has two main components:

1. I have built **MessyBoard**: a software communication medium that provides a networked, two-dimensional, freeform, finite, What-You-See-Is-What-I-See (WYSIWIS) space with a simple user interface.
2. I have integrated the medium into people's lives in a natural way by projecting it on the wall and displaying it as a screen saver on idle workstations. The medium runs in a web browser with no software installation so that people will be able to start using it with a minimal investment of time and effort.

1.2 The Medium

The communication medium is a networked 2D bulletin board. I built the initial prototype as a part of the "Information Cockpits" project (Tan, Stefanucci, Proffitt, & Pausch, 2001) with a different goal in mind: my colleagues and I wanted an easy way to share pictures and project them on the wall of our lab so that they could act as memory cues in the future. For instance, if I wanted my colleague to remember a conversation that happened in the lab several months ago, I might say to her: "remember, there were pictures of fighter planes projected on the wall." Psychology studies have shown that these kinds of "context" cues can be helpful (S. M. Smith & Vela, 2001).

It takes a lot of time and effort for one person to find many suitable pictures to project on the wall, but I thought that if it were easy enough then everyone in the lab would contribute pictures, perhaps just for fun. I built a networked bulletin board system called MessyBoard in order to allow anyone to decorate the wall instantly (Fass, Forlizzi, & Pausch, 2002). The MessyBoard client runs in a window on any user's computer, and the same client runs on a computer that is connected to a projector. Putting pictures or text on

MessyBoard is as simple as dragging and dropping or cutting and pasting from any other Windows application.

When I deployed MessyBoard, I observed some interesting collaborative behaviors, such as long-running asynchronous exchanges of text and pictures (Figure 1.1), and the playful creation of visual humor (Figure 1.2). I believe that these behaviors are enabled by specific properties of a 2D mixed-media shared space. Specifically, persistent notes and pictures can be used to keep a conversation grounded and spatial proximity and overlap can be used to quickly establish that a new object refers to an existing note or picture. Thus, MessyBoard **reduces the authoring cost** for any communication involving these activities. Further, the ability to put anything anywhere is conducive to humor and creative expression, **increasing the enjoyableness** of communication.

1.3 Natural, Low-Cost Integration Into People's Lives

Some communication methods, such as telephone calls or dropping by a colleague's office unannounced, allow an initiator to interrupt a receiver. Other communications media require that the receivers set aside time every day to poll their messages, as in e-mail or voice mail. Both of these receiving costs are high. One field study finds that software developers are interrupted an average of 3 to 5 times a day and that they spend an average of 20 minutes dealing with an interruption, for a total of 1 to 1.5 hours or 15-20% of their total time spent dealing with interruptions (van Solingen, Berghout, & van Latum, 1998). A separate field study finds that for real-world tasks and interruptions, over 40% of interruptions result in the worker not returning to the original task (O'Conaill & Frohlich, 1995).

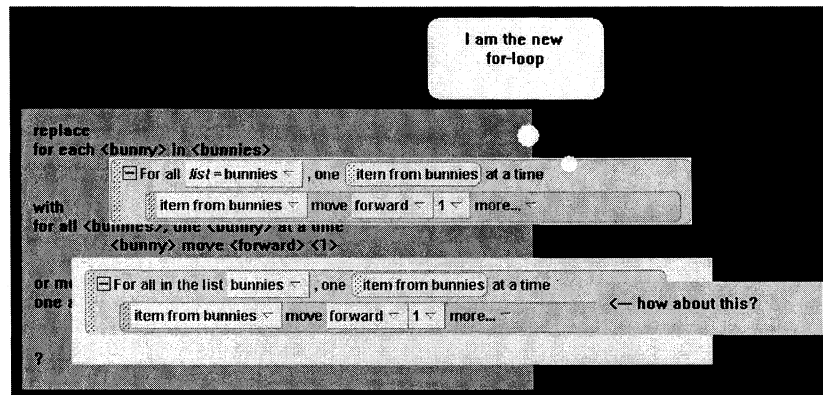


Figure 1.1. A mixed media conversation on MessyBoard about the design of a drag and drop programming interface.

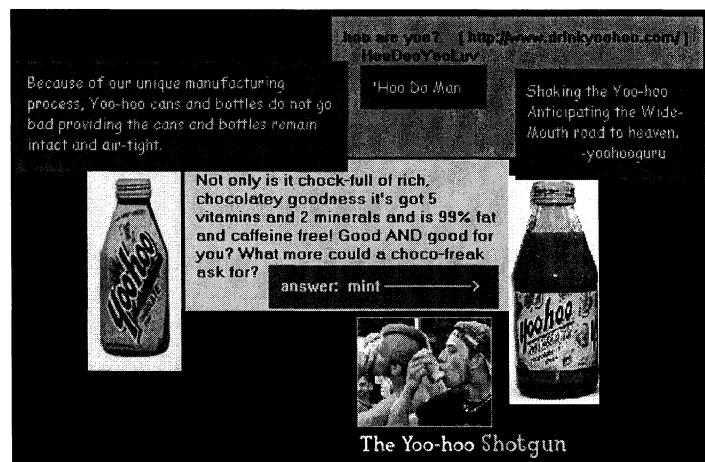


Figure 1.2. A collaborative humorous collage on MessyBoard.

However, there are usually times in a persons' schedule when they are able to process a little bit of information without losing productivity or being annoyed. For example, people often do not mind if someone asks a question on the way to lunch or during a stretch break. My goal is to **eliminate the costs of interruption and polling by getting people to pay attention to MessyBoard during these convenient times**. My approach is to display the medium in two places where users will observe it naturally, unavoidably and without much effort: on the wall of a shared work space and as a screen saver on idle workstations.

1.4 Observations and Results

I have deployed MessyBoard to a total of 67 groups at Carnegie Mellon University and I have analyzed the use of 124 MessyBoard spaces created by anonymous users on the Internet at www.messyboard.org. I began with a few groups that I observed directly and as I expanded my user base I relied on online surveys and logs of recorded activity on MessyBoard.

My main result is that large groups (25 or more members) are more likely to adopt MessyBoard and find value in it than small groups (15 or fewer members). I deployed MessyBoard to four large groups at Carnegie Mellon University and three of those groups adopted it, using it for 7 to 12 months. Two of those groups are still using MessyBoard at the time of this dissertation's completion. I deployed MessyBoard to 37 small groups and only 6 of those groups adopted it.

Large groups used MessyBoard for a combination of playful and goal-directed behavior. Group members frequently post inside jokes or humorous pictures on MessyBoard. Every so often, group members use MessyBoard to accomplish a task that requires communication. In the most striking example, a large group created a process for choosing new offices by placing a floor plan on MessyBoard as shown in Figure 1.3. Other examples include organizing movie outings and collecting a list of birthdays.

The small groups that used MessyBoard and found value in it tended to use it for specific work-related functions. The most common pattern was for groups to use MessyBoard as an annotated file repository as shown in Figure 1.4. Group members posted files on MessyBoard, clustered them spatially into meaningful groups and annotated the files and groups with notes. Other less common patterns include using MessyBoard to create collages of concept art and using it to collaboratively and synchronously edit lists.

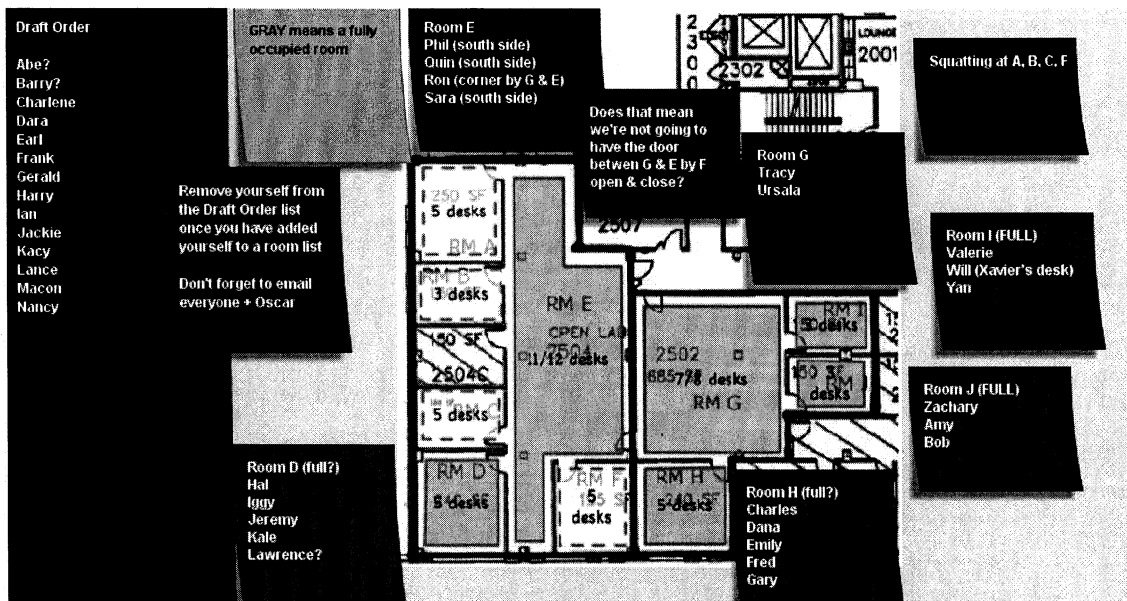


Figure 1.3. A large group uses MessyBoard to choose offices.

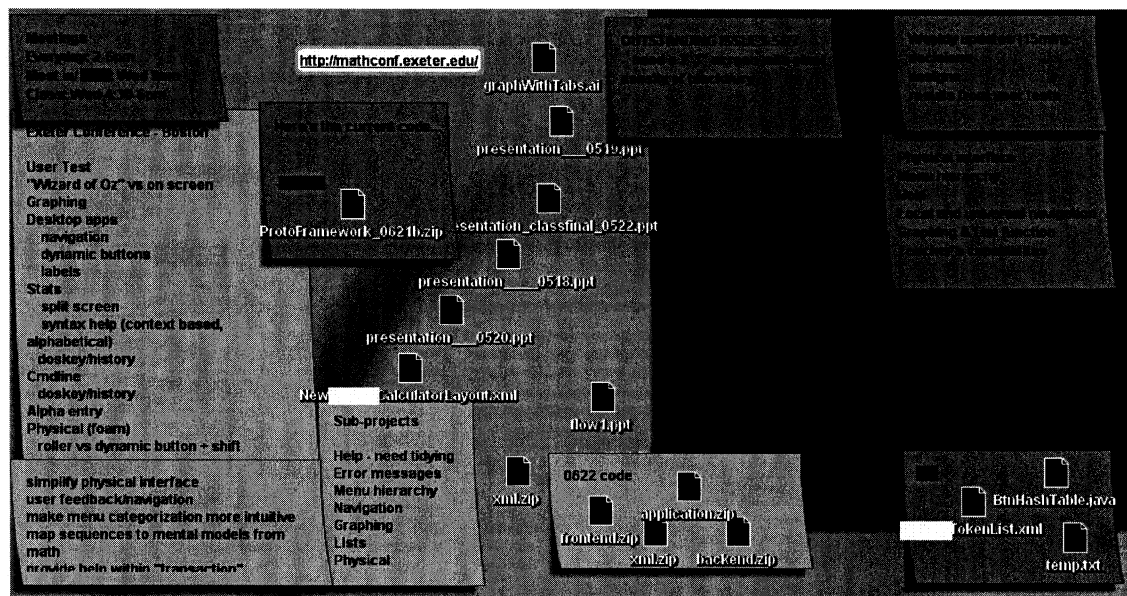


Figure 1.4. A small group uses MessyBoard as an annotated file repository.

1.5 Thesis Statement

In my work, I seek to show that:

A communication medium based on the metaphor of a bulletin board with free-form layout of text and pictures lowers the cost of communication and makes it more enjoyable. Projecting the medium on the wall and/or displaying it as a screen saver decreases the cost of communication and increases enjoyableness. A combination of iterative software design and testing, ethnographic observations, surveys and activity logs provides evidence to support these claims.

1.6 Contributions

The main contributions of this work are the findings that large groups are more likely to adopt MessyBoard than small groups and that large groups use MessyBoard for a mixture of playful and goal-directed behaviors while small groups tend to use it as an annotated file repository. These findings are based on observations of a large number of diverse groups and have important implications for practitioners that might wish to productize and market a tool like MessyBoard.

Another contribution of this work is MessyBoard, a tool that is designed with the explicit goals of reducing the cost of communication and making it more enjoyable. These goals lead to a unique combination of features and design tradeoffs that set MessyBoard apart from other communication tools. In particular, MessyBoard provides a single finite space to ensure that users can view it at a glance, whereas many other communication tools provide an infinite space or multiple spaces. The combination of a freeform 2D networked shared space with a screen saver is also unique.

A robust implementation of MessyBoard is publicly available at www.messyboard.org. Users can create their own MessyBoard spaces and invite their friends and coworkers to use them. The MessyBoard server is hosted and maintained by Carnegie Mellon University. At the time of this dissertation's completion, users have created 205 MessyBoard spaces.

1.7 Dissertation Organization

Chapter 2 discusses the design of MessyBoard and the underlying design rationale. In Chapter 3, I discuss the software architecture and implementation of MessyBoard and the rationale behind the major software design decisions. Chapter 4 reviews related systems and literature. In Chapter 5, I present my observations of MessyBoard usage in the field. Finally, 0 summarizes my work and discusses future directions for this research.

Chapter 2

MessyBoard Description and Design Rationale

2.1 A Networked Bulletin Board

MessyBoard is a networked bulletin board that allows a small group of people to share notes, pictures and other content. Everyone who looks at a MessyBoard sees exactly the same thing, and all users see changes in real time. Users view MessyBoard using client software on their own workstations and a central server keeps all of the clients synchronized in real time. Users add content to the board by using a menu or by dragging and dropping or cutting and pasting from other applications. Users modify existing content using direct manipulation. Figure 2.1 provides an overview of the MessyBoard user interface and depicts each of the objects that can exist on the board.

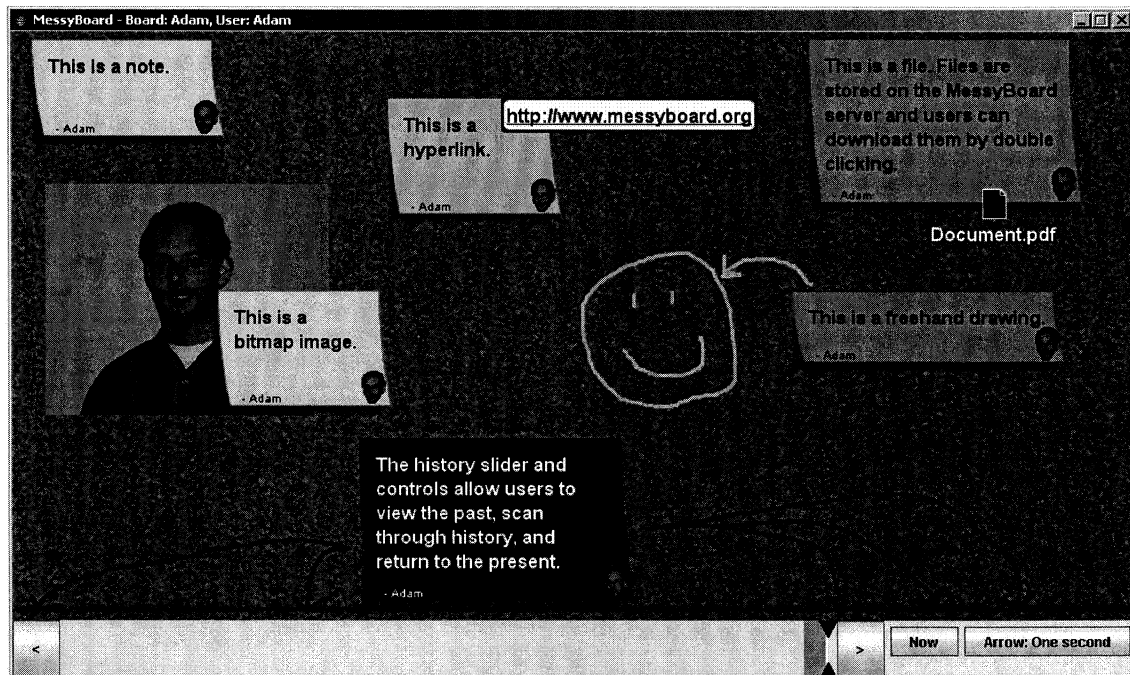


Figure 2.1. The MessyBoard interface, running in a window as a stand-alone application.

MessyBoard is primarily intended for use by a small group of people. Different groups have distinct MessyBoard spaces and each space has a name. Users can access the MessyBoard space for their groups by navigating a web browser to a URL of the form www.messyboard.org/MyGroupName. (URLs for MessyBoard spaces are not case sensitive.) The web page at that URL contains an embedded Java applet that runs the MessyBoard client software. From the user's perspective the MessyBoard space exists on the web at that URL, as shown in Figure 2.2.

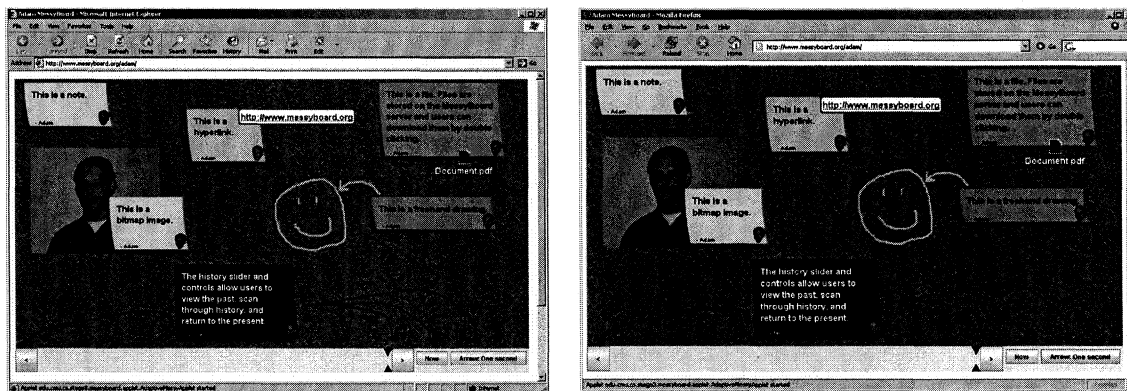


Figure 2.2. The MessyBoard Java applet running in Microsoft Internet Explorer (left) and Mozilla Firefox (right). The applet runs in a variety of web browsers on the Windows, Macintosh and Linux platforms.

The characteristics of MessyBoard were carefully chosen in order to create a medium that reduces the costs and increases the enjoyableness of communication. In this section, I discuss the characteristics and features of MessyBoard in detail and explain how they are meant to reduce costs and increase enjoyment.

2.2 Startup Cost

Some communication media require users to invest time and effort up front, before they derive any of the benefits that the medium has to offer. I refer to this time and effort as the **startup cost** for the medium. For example, consider a Yahoo! Group (Yahoo, 2005), which is similar to an e-mail list with some additional features. In order to create a Yahoo! Group, one must first create a Yahoo! account and then create the group itself. The entire process requires the user to fill out several lengthy forms as shown in Figure 2.3.

YAHOO! GROUPS

Create Your Yahoo! ID

Username:

Password:

Confirm Password:

If You Forget Your Password:

Secret Question:

Answer:

Customizing Yahoo!

Search:

Verify Your Registration

Terms of Service

1. Create a Yahoo! Account

YAHOO! GROUPS

Start a Group

Step 1 of 3: Select a Yahoo! Groups Category

Please select the category that best describes your group. Please be as specific as possible when selecting the category. Groups containing sexual content must be placed in the Sexuality and Relationships / Adult category.

Search Group Categories:

OR

Browse Group Categories

- Business & Finance
- Computer & Science
- Culture & Community
- Entertainment & Arts
- Family & Home
- Food & Drink
- Health & Wellness
- Hobbies & Crafts
- Music
- Pets
- Relationships & Social
- Religion
- Science & Technology
- Sports & Recreation
- Travel & Adventure
- Other

2. Select a category for the group (use a search engine or drill down through the category hierarchy)

YAHOO! GROUPS

Start a Group

Step 2 of 3: Describe Your Group

Group Name

This name will appear on your Group page and in search results.

Example: The Great Group

Enter Your Group Email Address

This email address is for your group. When you send a message to this address, all members of your group will receive a copy. (All characters must be in lower case or space characters allowed)

Example: contactgroup@earthlink.net

Describe Your Group

This description will appear at the top of your group page (2000 characters maximum)

3. Type a name and short and long descriptions of the group

YAHOO! GROUPS

Start a Group

Step 3 of 3: Select your Yahoo! Profile and Email Address

Email Address

If you choose to receive group messages by email, they will be sent to the email address you select below.

Example: user@earthlink.net

The following email addresses from Account Info cannot be used to start this group.

Example: user@earthlink.net (use earthlink)

Yahoo! Profile

The profile you select will be used when other members want to learn more about you, or communicate with you via Yahoo! Messenger.

Member: Real Name: Gender: Age:

Example: user@earthlink.net (Not Provided) M (Not Provided)

Word Verification

Enter the text as it is shown in the box below. This step helps Yahoo! prevent automated registrations.

Example: Fw38

4. Select an e-mail address and user profile to associate with the group and pass a final security test

Figure 2.3. A user must fill out several forms to create a Yahoo! group.

The startup cost is a one-time cost, and one might imagine that users will choose to invest time up front in order to reap the benefits of a new communication medium in the near future. However, by requiring users to invest time and effort up front, startup cost can delay use of a communication medium or dissuade users for using it at all.

Consider the following scenario: An organization has just formed an ad-hoc committee and they are having their first meeting. The group decides that they need to discuss some things over e-mail before the next meeting and they agree on the need for a Yahoo! Group. Since the startup cost of creating a Yahoo! Group is high, they most likely will not create the group during the meeting while everyone is waiting. They are more likely to assign the task to someone. Once the meeting is adjourned and the excitement and urgency dissipates, that person is not as likely to create the Yahoo! Group immediately. If that group member is distracted or procrastinates, other group members will not be able to communicate and precious time will be lost.

The following scenario demonstrates another danger of startup cost. Imagine that a person wants to facilitate communication between members of her family. She has a vague idea that a Yahoo! Group helps people keep in touch, but she is not sure exactly how it works or what it does. She decides to create a group in order to find out, but when she sees the long form that she needs to fill out she loses interest and decides not to continue.

The purpose of these scenarios is not to point out that people act irrationally. On the contrary, the ad-hoc committee wisely decides not to waste everyone's time waiting for the creation of a list when one person can do it outside the meeting. For the woman who wants to foster communication in her family, the rewards may outweigh the startup cost, but she has no way of knowing this until the group is created. She is making a decision in the face of uncertainty, and investing time and effort up front is risky.

It is also not my intent to criticize Yahoo!'s decision to require users to invest time and effort in order to create a Yahoo! Group. Yahoo! has made a business decision to require users to fill out lengthy forms and they use the information to prevent abuse and for mar-

keting purposes. They have designed the startup process with other considerations in mind besides improving collaboration.

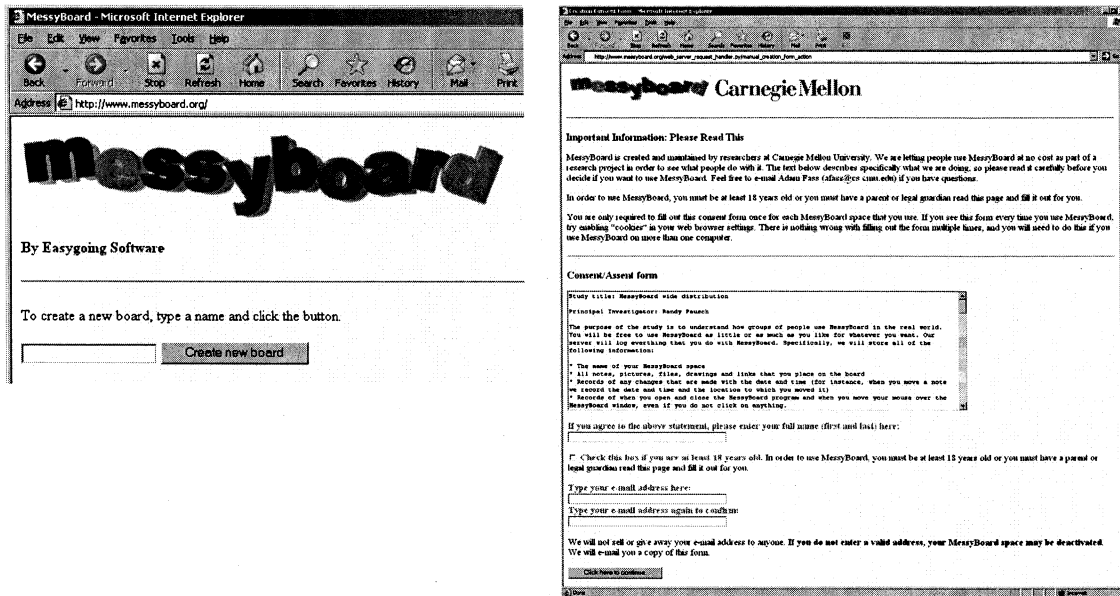
The purpose of these scenarios is simply to point out the potential danger of a seemingly small startup cost. Such costs are common and often taken for granted. When people consider these costs in the abstract, they often assume an ideal situation in which a single user weighs the costs and benefits and immediately makes the correct decision. Instead of an ideal global cost-benefit tradeoff, I propose the following two questions as guidelines for judging startup cost:

- Would a newly formed group of people find it acceptable to incur the startup cost at their first meeting?
- Would a user incur the startup cost without a clear idea of the benefits they will gain from using the medium?

The forms required to create a Yahoo! Group are only one example of the startup cost for a communication medium. Other kinds of startup costs include software installation and learning to use a new user interface. It is my goal to create a communication medium in which all forms of startup cost are reduced to a bare minimum.

2.2.1 Reducing the Startup Cost: Creating a MessyBoard Space

To create a new MessyBoard space, a user can simply type a name into the text field on the MessyBoard home page (www.messyboard.org) and click on the button as shown in Figure 2.4.



1. Type a name and click the button

2. Fill out the consent form

Figure 2.4. To create a new MessyBoard space, a user types a name, clicks a button and then fills out a consent form. On completing the consent form, the user is taken directly to the MessyBoard space. Other users will see a similar consent form the first time they view the space.

The consent form is responsible for the bulk of the startup cost, and it is only necessary because MessyBoard is a research project and the server is logging users' actions. If this were not the case, the new space would be created instantly.

I have reduced the startup cost further by providing an alternate method for creating a new MessyBoard space. If the user types a URL of the form www.messyboard.org/mynewboard, the server will inform her that the space does not exist and ask if she wishes to create it. The prompt is necessary to prevent confusion, since the user may have made a typographical error while trying to access an existing MessyBoard space.

2.2.2 Reducing the Startup Cost: Accessing a MessyBoard Space

Users can access their groups' MessyBoard space simply by typing a URL into their web browser. This seems an obvious idea, and indeed several popular e-mail services already make it possible for users to access their e-mail without any software installation (Yahoo!, 2005). A survey conducted by Wheeler et al. finds that the ability to use applications without installing client software is a major advantage for web-based groupware (Wheeler, Dennis, & Press, 1999). However, it is still the case that some of the most popular and widely known groupware applications still require users to download and install software before (Groove, 2005). This startup cost can reduce the value of a group communication tool and have a crippling effect on adoption (Wheeler et al., 1999). In many circumstances in both corporations and academia, users may not even have the capability to install new software.

Easy access to MessyBoard through existing software and an easy-to-remember URL enables the following scenarios. A person can e-mail the URL to a coworker and the recipient can simply click the URL to start using MessyBoard right away. Because the URL is easy to remember, a user can visit a coworker's office and tell her to type it in. An audience member at a small meeting or presentation might ask the presenter to type in the URL and display it on the screen for all to see. These scenarios may be merely plausible for a tool that only requires a web browser, but they are almost impossible for a tool that requires software installation.

2.3 Authoring Cost

Sometimes it takes a great deal of time and effort to express a complex thought in a way that is precise and unambiguous using a given medium. For example, it takes time and effort to write an e-mail message with clear and unambiguous automobile driving directions from one location to another. I refer to this time and effort as **authoring cost**.

Note that the authoring cost depends on the medium and the nature of the information to be communicated. For example, it may take very little time and effort to communicate driving directions by drawing a line on top of a map.

2.3.1 A Spatial Medium Reduces Authoring Cost

MessyBoard provides a persistent 2D space where users can easily arrange objects with direct manipulation. These characteristics lower the authoring cost of two important communication activities: referring to an object and grounding a conversation (Clark & Brennan, 1991). For example, in Figure 2.5 an author has used overlap and proximity to establish that he is referring to an existing picture. The entire conversation is grounded because the objects are persistent. In other words, it is always clear what the conversation is about because the relevant objects are near the discussion.

I have observed that users naturally group objects in space on MessyBoard in order to communicate that the objects are related to each other and/or unrelated to other items. For example, Figure 2.6 shows how users have used proximity and overlap to communicate the fact that the files on their board are clustered into groups. In Figure 2.7, it is obvious which comments refer to each of the pictures.

Figure 2.8 shows a more structured spatial organization in which lists of occupants for offices are placed on top of a floor plan. Details on this group's usage can be found in Section 5.5.1.1.2.

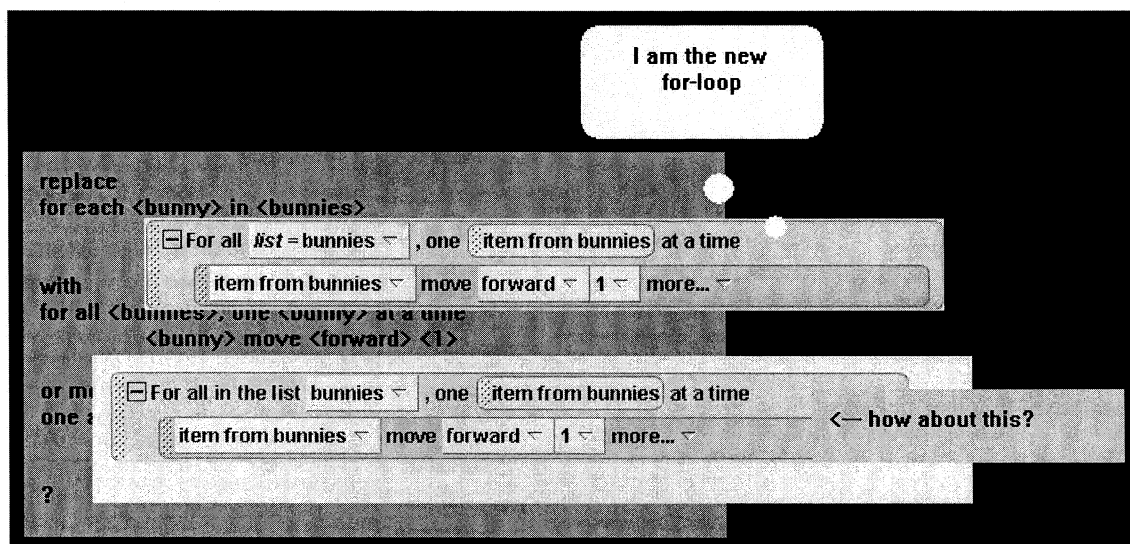


Figure 2.5. A design conversation in a prototype version of MessyBoard. The authors of the note that reads "how about this?" and the thought bubble that reads "I am the new for-loop" both use proximity and overlap to establish a referent.

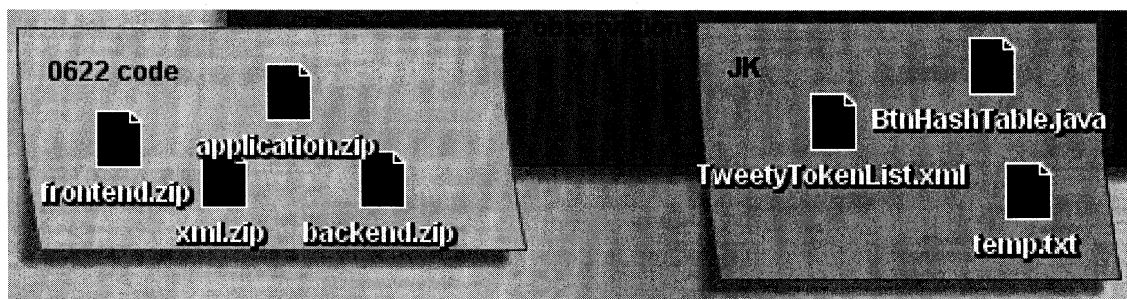


Figure 2.6. MessyBoard users arrange files in clusters on top of notes to indicate that there are two separate groups.

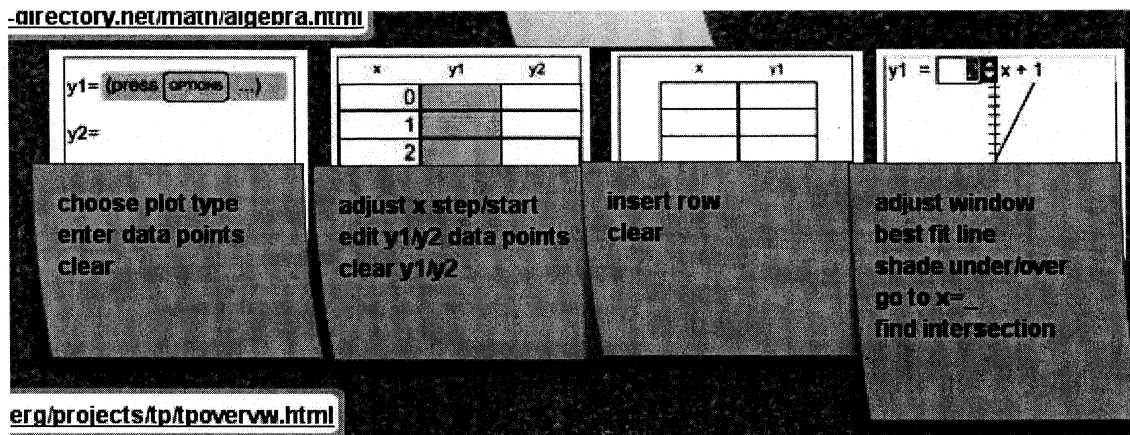


Figure 2.7. A MessyBoard user places comments over the pictures to which they refer.

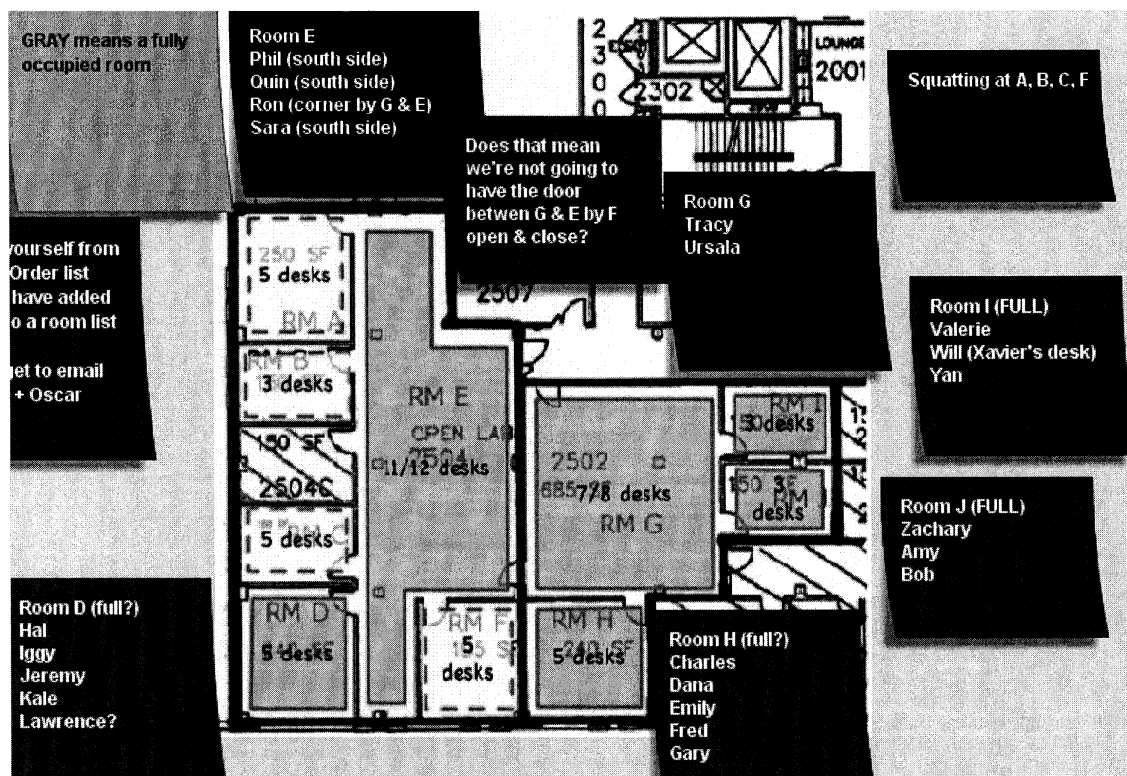


Figure 2.8. Users place notes over a floor plan in order to assign occupants to individual rooms.

2.3.2 The MessyBoard Interface Reduces Authoring Cost

The MessyBoard interface is designed to reduce authoring cost to a minimum. Creating a new note is one of the most common MessyBoard operations, and this is accomplished by double-clicking in an empty spot on the board. Double-clicking on an existing note positions the cursor in that note, allowing the user to edit the text.

Users can add content to the MessyBoard space by dragging and dropping or copying and pasting from other applications or from a graphical file browser such as Windows Explorer or Macintosh Finder. MessyBoard attempts to handle dropped or pasted data in such a way that the user will not have to modify it any further. When the user drops or pastes text MessyBoard creates a note with that text and the note is sized to fit the text exactly. If the text is a URL then MessyBoard creates a clickable hyperlink. Dropping or pasting an image creates an image on MessyBoard, and the image is automatically scaled so that the top left corner appears at the drop point and the entire picture fits within the bounds of the space. (The top left corner was an arbitrary choice. A better choice may be the center of the image or the point where the drag operation was initiated.)

MessyBoard also allows users to add objects to the space using a menu (Figure 2.9) that appears when the user clicks the right mouse button anywhere in the space. (Ctrl-click for Macintosh computers.) However, the interface is designed so that the menu is rarely necessary. One notable exception to this is the drawing feature, which adds freeform pen strokes to the board. This feature is buried in a submenu and most users are not aware of the keyboard shortcut (Ctrl-D). I found that many users were unaware of the existence of the drawing feature, even after I had demonstrated it to them a few weeks earlier. This may be due to the fact that the lightweight interface using only double-clicking, dragging and dropping made it unnecessary for users to ever explore the menu.

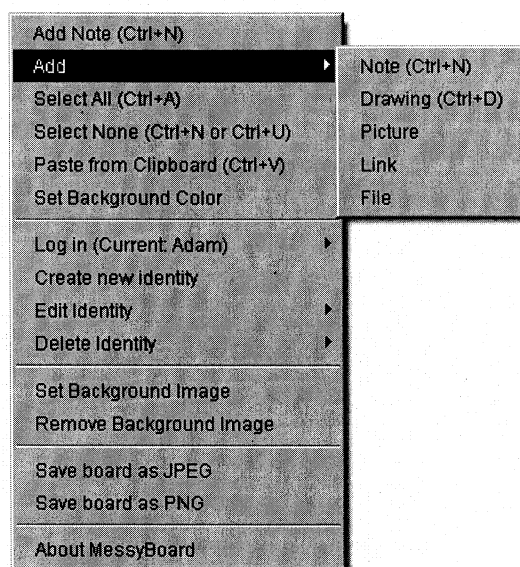


Figure 2.9. The MessyBoard main menu.

Users often need to move and resize objects on MessyBoard to make more room or to position objects in a meaningful way. Users do this with direct manipulation: Dragging the corners or edges of an object makes it larger or smaller and dragging the middle of an object moves it. MessyBoard provides invisible resize “halos” so that the user can initiate a resize operation even if the cursor is a few pixels outside the edge or corner of an object. The mouse cursor changes shape to provide feedback to the user when the cursor enters a halo. Making targets “oversized” and therefore easy to hit is part of the MessyBoard design philosophy of making things pragmatically as easy as possible.

Objects in MessyBoard have a Z ordering, and new objects are always created at the top of the order so that they sit on top of existing objects. This allows users to easily annotate existing objects with overlapping notes or pen strokes.

Users often wish to move several objects at once. For example, the groups of files in Figure 2.6 are arranged in meaningful patterns and the user might wish to move an entire group off to the side to make room for new content. Drawing interfaces such as Microsoft PowerPoint provide a grouping feature, allowing the user to explicitly tag a set of objects as a group so that they will all move together when any one of them is dragged. This style of grouping has several problems. First and foremost, the user is required to explicitly

designate a set of objects as a group, which adds to the authoring cost. In addition, not all users may be aware of the feature, and if one user creates a group then other users may be confused when they try to move a single object and the entire group moves.

MessyBoard provides a lightweight grouping interface that leverages the natural tendency of users to place related objects so that they overlap one another. When a user moves an object, any objects that are on top of that object and completely within the bounding rectangle of that object are also moved. Thus, a small drawing on top of an image automatically moves with the image, and the files in Figure 2.6 move whenever a user moves the note. Since there is no explicit “group” of objects, the user can move an individual file simply by dragging it off of the note.

2.3.3 Identification

For most communication media, an essential requirement is to make the identity of the sender known to the receivers. Most electronic media do this in way that imposes substantial startup cost. For example, a user must create an account and fill out forms in order to use e-mail or a chat program.

MessyBoard users can make their identities known simply by typing their names in their notes, and many casual users choose to do this. There is no startup cost, and typing their names takes very little time.

For more frequent users, MessyBoard provides an identity feature so notes are automatically decorated with a custom color, font, picture and a screen name as shown in Figure 2.10. A user creates a MessyBoard identity using the main menu. Once she has created an identity, MessyBoard records her preferred identity on her computer and logs her in automatically whenever she uses MessyBoard. She can quickly log in from any computer by selecting her identity from a menu as shown in Figure 2.11. MessyBoard identities do not require a password to log in. They are not intended to provide authentication. Rather, they are merely intended to allow users to easily identify themselves in an environment where group members are assumed to know and trust one another.



Figure 2.10. A note with the author's screen name and picture.

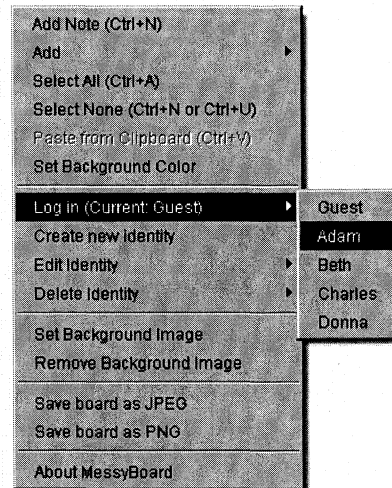


Figure 2.11. The login menu is a submenu on the main MessyBoard menu.

2.4 Receiving Cost

Receiving communication requires time and effort. At a minimum, a receiver must spend time and effort reading or listening to the content of a message. Some media require additional effort. For example, an e-mail client displays a list of unread messages and the user might have to search through them to find one that she is expecting. I refer to this time and effort as **receiving cost**.

2.4.1 Finite Space Reduces Receiving Cost

The user interface of a communication tool often imposes a receiving cost on the user. For example, in order to read e-mail messages, the user may be required to scroll through a list, double-click on messages and scroll through individual messages in order to read all of the content.

MessyBoard reduces receiving cost by providing a finite space. MessyBoard fits on a standard computer screen with no scrolling or zooming so that users can see everything on the board at a glance. Users can read all of the information on MessyBoard without touching a mouse or keyboard.

This reduction in receiving cost relies on a shared understanding between group members about how MessyBoard is to be used. For example, a user could place a new message under an existing object and expect that other users will rearrange the objects in order to read the new message. If receivers are aware of this and the behavior becomes a convention, then the receiving cost is raised. Such conventions have not been observed in practice, and the interface is designed to discourage them.

A limited amount of space implies that MessyBoard can contain a limited amount of information. It is helpful to think of the space on MessyBoard in economic terms: space is a scarce resource, and therefore it has inherent value. When a sender posts information on MessyBoard, she implicitly makes a judgment: Is the information she is about to post worth the space that it will occupy? Is it as valuable as the information that they are about to remove or cover up in order to make room? Thus, the information on MessyBoard at any given time represents an implicit group consensus: Given a small amount of time to view some information, the information in this valuable space is the most important.

Note that finite space actually increases the authoring cost, since a sender needs to rearrange or delete existing items based on judgments about the value of the information. The section below describes how a shared history mitigates this cost, but it is still the case that a finite space reduces receiving cost at the expense of increased authoring cost. This tradeoff is desirable, since for any given communication there is a single sender and many receivers.

2.4.2 History

The discussion above omits one serious problem: Not everyone in the group is certain of the value of any given piece of information to other group members. This uncertainty, combined with social norms such as politeness and respect for authority, leads to unwillingness to delete information from MessyBoard, especially if it was posted by another person. When I deployed an early prototype, I anecdotally observed that old information remained on MessyBoard long after most group members knew that it was obsolete. Thus, the prototype was often cluttered with old information and would-be senders complained that they could not post new information for lack of empty space.

My approach to the problem of clutter was to give users confidence that they could delete old material without disastrous consequences. MessyBoard provides a shared history mechanism that allows any user to go back in time in order to view and recover old material. The MessyBoard server automatically stores a complete history of every change to the MessyBoard space, and users can access this history using a simple slider interface as shown in Figure 2.1.

The main intent of the history feature is accomplished even if users do not actually use it. A sender must simply know that deleted information will not be lost forever and that anyone can recover it, and this gives her the confidence to delete it. This is similar to providing an "undo" button in a simple user interface, even if it is seldom used, for the purpose of stress reduction. Users are comforted just by knowing it is there if they ever do need it.

History adds an infinite third dimension of time to the existing finite 2D space, and one could argue that the addition of history eliminates the scarcity and hence the implicit value of the space that is so important to reducing the receiving cost of communication. Once again, this depends on the shared understanding of the group about how the history is to be used. A sender could create a note and then delete it, expecting that receivers will browse the history and see it. In reality, such conventions have not arisen and the history interface does not appear to encourage this sort of behavior.

2.4.3 A Poverty of Attention

Up to this point, I have discussed the receiving cost of communication assuming that a person is able to receive a communication in the first place. In reality, just because a communication medium exists does not mean that users will receive communications through that medium. For example, a user must remember to check her e-mail in order to read it and she must be present when the phone rings in order to answer the call.

In one way or another, a user must focus attention on a communication medium in order to receive communications. Much has been written about the poverty of attention in modern life, due in part to the proliferation of new modes of communication (Davenport &

Beck, 2002). In general, every communication medium uses one of three mechanisms to ensure that users devote attention to it: interruption, polling and advance scheduling. Telephones and some chat programs interrupt the user unexpectedly in order to get their attention, the former by ringing and the latter by opening a new window on the computer display. Polling refers to a habit of regularly checking for new messages. For example, many people remember to check their e-mail and voice mail messages at least once a day. Scheduling in advance is common for face-to-face meetings and teleconferences. Figure 2.12 lists some common communication methods and the techniques that they employ to guarantee that the receivers pay attention.

	Interruption	Polling	Scheduling
Go to office to talk	✓		
Telephone	✓		
Instant message	✓		
E-mail		✓	
Voice mail		✓	
Meeting			✓

Figure 2.12. Common communication methods and the ways that they guarantee that receivers will pay attention.

The cost of interruption, polling and scheduling measured in time and effort are quite high and almost everyone uses at least one medium that employs each technique. A new communication medium that employs any of these techniques faces overt competition, as people are aware of these costs, they find them unpleasant, and they are not willing or able to give up existing modes of communication.

2.4.4 Salvaging Attention with Projectors and Screen Savers

While modern communication media demand a lot of attention, there are periods of time in most peoples' days in which they could process a little bit of extra information at a very low cost and without feeling overwhelmed. For example, most people do not mind if a colleague asks them a question on the way to lunch or during a stretch break. My design goal is to display MessyBoard such that people see it at these convenient times.

One solution is to display MessyBoard on a large public display in a shared workspace as shown in Figure 2.13. People unavoidably see MessyBoard when they arrive and they glance at it before they leave or when they are taking a break. Just as with traditional, physical bulletin boards, the exact location is crucial. Ideally, it should be visible in a common area, such as a coffee lounge, or in the line of sight of a commonly walked path. A large public display can also encourage conversation since people can tell when others are looking at the board and several people can stand around the board and point to it.

An alternative solution is the MessyBoard screen saver, shown in Figure 2.14. When the screen saver is installed, the computer displays MessyBoard when the computer is idle for a period of time. This means that the user will see MessyBoard when she arrives at work or when she returns from a break. The screen saver does not encourage conversation in the way that a large public display does, but it is a good alternative for environments where a large public display is not feasible. Large displays at present are also expensive, and the screen saver utilizes existing display hardware at no extra cost. (Even with projector prices dropping, the cost of keeping a projector on for 8 hours a day, 5 days a week is approximately \$450 per year for a modestly bright projector. The electricity costs approximately \$40 per year and the bulbs cost approximately \$410 per year.)

The MessyBoard screen saver is interactive and fully functional, allowing the user to post and edit material on the board. Instead of disappearing when the mouse is moved, as most screen savers do, the MessyBoard screen saver is dismissed by pressing the Escape key. This allows users to quickly reply to a message in the moment when they are already paying attention to it without paying the unnecessary price of exiting the screen saver, opening a web browser and typing a URL.

The interactive nature of the screen saver opens up an interesting possibility in an open-plan lab: Idle workstations can become public MessyBoard terminals. Users can configure the standard password protection in the Windows operating system so that others can interact with the MessyBoard screen saver on their workstation but they cannot use the computer for anything else.

I have chosen to combat the problem of limited attention by displaying MessyBoard in such a way that people naturally view it at convenient times. An alternate approach is to sense the user's attentional state in real time in order to deliver messages only when the benefits outweigh the costs of disruption (Fogarty, Hudson, & Lai, 2004; Horvitz, Kadie, Paek, & Hovel, 2003). This is a very difficult problem and the research is still in its early stages. In that sense, the present research is similar to the "Office of Real Soon Now," (Bishop & Welch, 2000) in that both are driven by short-term pragmatic goals, as op-

posed to the long-term goals of the "Office of the Future" project (Welch, Fuchs, Raskar, Towles, & Brown, 2000).



Figure 2.13. MessyBoard projected on the wall in a shared work space.

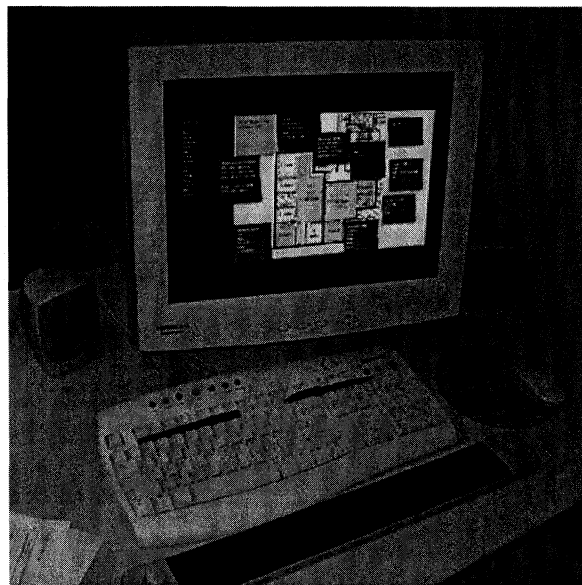


Figure 2.14. The MessyBoard screen saver.

2.5 A Spatial Medium Makes Communication Enjoyable

A freeform 2D space may encourage certain kinds of playful and creative behavior. For example, the Stage3 lab at Carnegie Mellon University and Professor Dennis Proffitt's Perception Lab at the University of Virginia played a game of Pong (Figure 2.15). The paddles are actually small notes, and a bitmap of a white circle serves as the ball. There were no strict rules, since the ball did not move on its own, any user was able to move either of the paddles or the ball, and anyone could update the scoreboard. Nevertheless, people seemed to enjoy this game for a short while, and they did a lot of work to create it.

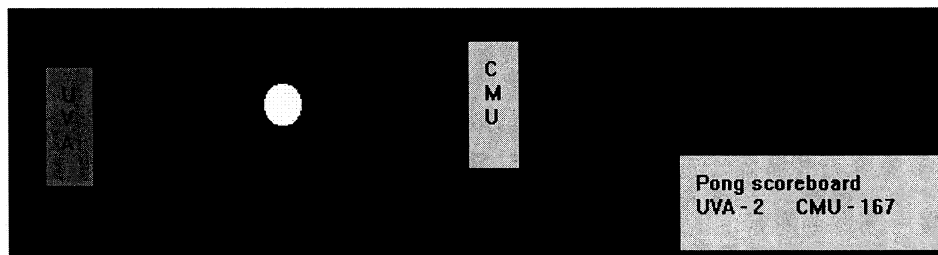


Figure 2.15. Two groups of users in different locations use an early MessyBoard prototype to play a game of pong.

Figure 2.16 shows another example: a group is using MessyBoard to schedule a ping pong tournament. Next to the ping pong tournament roster is the sign up sheet for the weekly lab meeting, and this underscores a critical point. A freeform 2D space allows work and play to occur in the same medium, right next to each other, and closely intertwined. Playful behavior may encourage people to pay more attention to MessyBoard and use it more often. If this is the case, they may also take part in some work-related exchanges that they otherwise would have ignored.

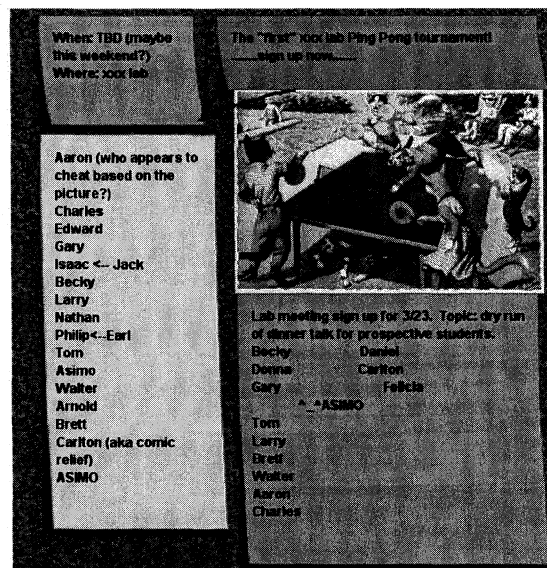


Figure 2.16. MessyBoard users sign up for a table tennis tournament (left) and for their next lab meeting (right).

Chapter 3

Software Design

3.1 Architecture Overview

MessyBoard is designed to be persistent: content on the board must continue to exist even if nobody looks at it for a period of time. MessyBoard must also be accessible from any computer on the Internet at any time. The simplest way to satisfy both of these constraints was to use a client-server model where the server is always online, similar to a web server.

MessyBoard employs a straightforward client/server architecture. The system state is stored in a networked database so that updates originating from the GUI of one client are automatically sent to the server and broadcast to all of the other clients.

Figure 3.1 shows the major components of MessyBoard. The GUI layer defines the appearance and interactive behavior of the MessyBoard space and objects such as notes and pictures that inhabit the space. The database layer provides a distributed database ensuring that data is automatically synchronized between remote clients. The network layer allows clients to send and receive messages asynchronously so as to keep the user interface responsive.

Shared data in the database is organized as a flat set of *elements*, each of which has a set of *properties*. Each property has a name (a string) and some data (a byte array). MessyBoard uses a single element to represent each object in the MessyBoard space such as a note or picture. The properties of the element represent attributes of the object such as the position and size and the text that appears in a note.

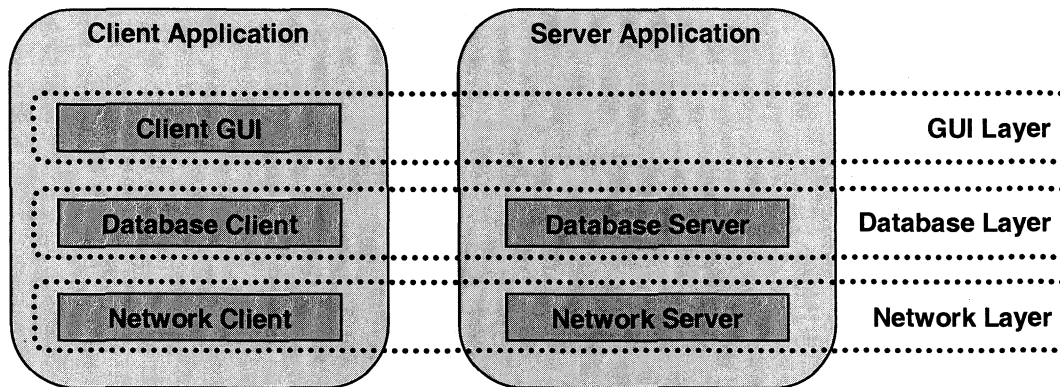


Figure 3.1. The MessyBoard architecture comprises three layers: GUI, database and network. The database and network layers each have a client and a server component.

The most complex part of the system is the database client and its interaction with the GUI layer. The GUI layer uses the Model-View-Controller pattern (Burbeck, 1992) and the database client serves as the underlying model. The database client is structured in order to allow the GUI code to be as simple as possible while at the same time ensuring that the user always gets immediate feedback for interactive operations, even if those operations require sending and receiving information over the network.

3.2 The GUI Layer

I began work on the new version of MessyBoard with many ideas about different kinds of objects that could go on the board and uncertainty about exactly how those objects should appear and behave. Clearly, some amount of rapid prototyping and iterative design would be necessary. It was important to be able to quickly develop new MessyBoard objects and focus on the GUI appearance and behavior without having to worry about the network messages required to keep different users' views synchronized.

To accomplish this goal, I use the Model-View-Controller pattern (Burbeck, 1992) to implement objects on MessyBoard. For each kind of object on MessyBoard, the GUI layer contains a Model class and a ViewController class. For example, the implementation for the note object comprises the NoteModel class and the NoteViewControl-

ler class. The `NoteModel` class defines and encapsulates the data that are associated with the note, such as its position and text, and it synchronizes the data with other clients using the database layer. The `NoteViewController` class specifies the rendering and the interactive behavior of the note. The relationship between objects that the user sees on `MessyBoard`, `ViewController` objects, `Model` objects and database elements is illustrated in Figure 3.2.

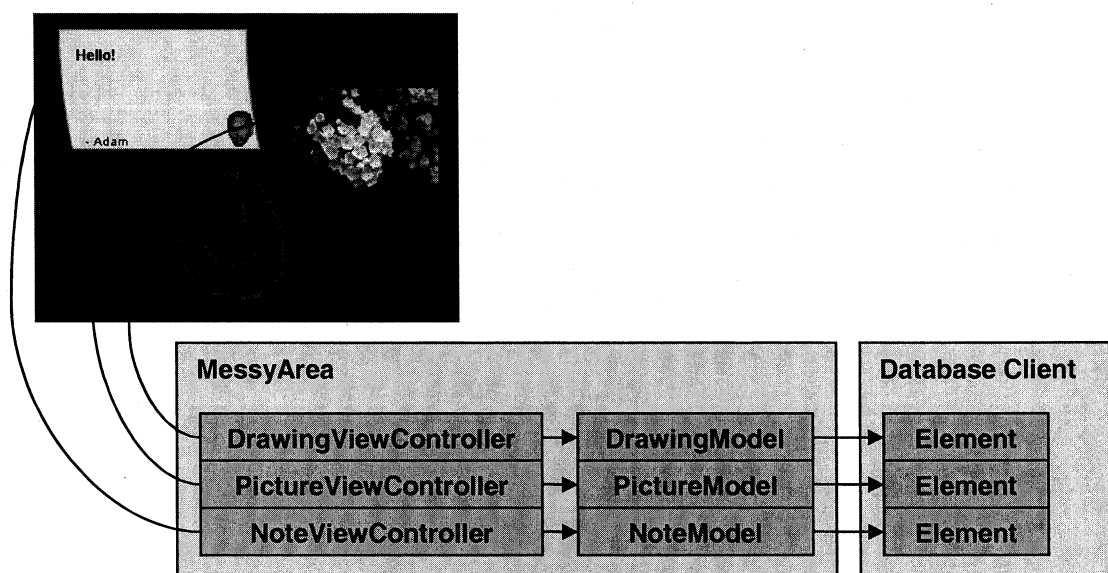


Figure 3.2. The `MessyBoard` space is represented by a `MessyArea` object. Each object that the user sees on `MessyBoard` is represented by a `ViewController` object which specifies the rendering and interactive behavior. A `Model` object encapsulates the shared data for each object and stores it as properties of a database element.

Once the developer has defined a `Model` class, she can change the rendering and interactive behavior by modifying only the `ViewController` class. Since the model handles the interaction with the networked database, modifying the `ViewController` class is as simple as building a component in any GUI framework. The developer can focus on the look and feel without thinking about network synchronization.

The `Model` class acts as a wrapper for an element in the network database, and its two main purposes are to provide type safety and convenience. For example, the `Note-`

Model class provides a `position` field and the `ViewController` class can change the note's position by calling `myModel.position.set(point)`. For the `ViewController` developer, this is more convenient than converting the point into a byte array and then setting the property on a database element. If the developer accidentally sets the position to an integer instead of a point, this error is caught by the compiler. If the `ViewController` class were to interact with the database element directly, such errors would result in corrupt database states at run time.

The `MessyBoard` space is represented by a `MessyArea` class. The `MessyArea` class is an AWT (Sun, 2005a) component that can be placed in a window frame, applet or any other AWT container. `ViewController` objects representing notes, pictures etc. are children of the `MessyArea` in the GUI component hierarchy as shown in Figure 3.2. When `MessyBoard` begins execution, the `MessyArea` initializes the database connection and creates a `Model` and `ViewController` object for every element in the database. The `MessyArea` also listens to the database for object creation and deletion events and creates and deletes `Model` and `ViewController` objects accordingly.

I chose AWT over the more advanced JFC/Swing library (Sun, 2005c) because it was important for the `MessyBoard` client to be able to run in a Java 1.1 virtual machine. However, AWT components are inadequate for creating irregularly shaped objects with partial transparency such as curved notes, and AWT controls like the `TextArea` class are too inflexible to create a custom look and feel. Thus, the `MessyArea` and `ViewController` classes extend the AWT `Container` class so that they can contain subcomponents be contained in an AWT GUI hierarchy, and they implement their own lightweight rendering scheme similar to JFC/Swing components. For editing and displaying text in notes, I use the lightweight text control provided in the KFC library (Yasumatsu, 2005) with minor modifications.

3.3 The Database Layer

The core of the `MessyBoard` implementation is a networked database. The database stores a collection of *elements* and each element stores a set of *properties*. Each property

has a name, stored as a string, and a value, stored as an array of bytes. The fundamental operations provided by the database are element creation and deletion, creating a property and changing a property value, and locking and unlocking an element. MessyBoard uses one database element to represent the state of each object (a single note or picture, for example) on the board. The element's properties represent the object's location, size, and object-specific information such as the text in a note.

Two basic requirements drove the design of the database:

1. Immediate feedback to the user for local operations
2. Real-time synchronization of remote clients

3.3.1 Immediate feedback

Immediate feedback is an important part of lowering the perceived and real costs of communication. A sluggish interface may be enough to dissuade a user from posting a quick comment. For this reason, the MessyBoard client gives the user real-time feedback for operations such as moving objects interactively (dragging with the mouse) before it can guarantee that these operations will succeed.

MessyBoard operations can potentially fail, for example if two people attempt to move a note at the same time. However, these conflicts are rare and there is little cost in notifying the user a few milliseconds after the fact that someone else is moving the note.

An alternative design would be to give immediate feedback to the user during an interactive operation while indicating explicitly whether the operation is pending or if it is guaranteed to be successful (Pausch, 1988). For example, when a user starts to move a note the system could leave the original note in place and create a transparent copy of the note that the user can manipulate. When the operation is acknowledged by the server, the transparent copy would disappear and the real note would move to the current location.

While the MessyBoard architecture supports this kind of interaction, such an interface may be distracting and unsettling to novice users. Many users may not understand why notes become transparent for a short while and then become solid, and since the likeli-

hood of a lock collision is so rare, it does not justify the distraction to the user in the common case.

3.3.2 Real-time synchronization

People may be reluctant to expend time and effort on a task if they have reason to believe that their work may be in vain. For example, a user would not want to spend time updating a shared "to do" list if she had reason to believe that someone else had already updated the list. Therefore, users must be confident that the information they see on MessyBoard is completely up to date. For this reason, all MessyBoard clients are updated in real time within the limits imposed by network latency (usually less than two seconds).

Another reason for real-time synchronization is that MessyBoard may be used for synchronous remote collaboration. Though this is not MessyBoard's primary intended use, if the tool is adopted by remote collaborators then it is possible that they will start to look at it during teleconferences or while they are sending Instant Messages to one another. Synchronous use requires fine-grained coordination where real-time feedback on the actions of other users is important.

3.3.3 Requested and Confirmed States

The need for real-time synchronization led to a design in which GUI elements make changes to objects in a database and the database client automatically and immediately updates the server. The server processes the request, sends an acknowledgement message to the client that requested the change, and sends a broadcast message to any other database clients that are connected. The database client uses the observer pattern (Gamma, 1995) to notify GUI elements immediately through a callback when a broadcast is received.

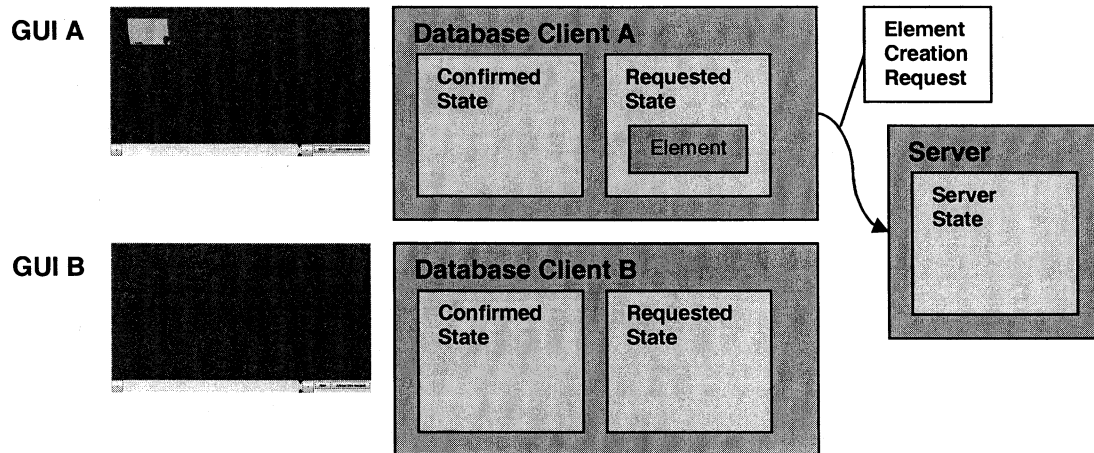
The need for immediate feedback dictates that when the user moves a note, the note is shown in a new position before the database client receives an acknowledgement from the server. However, for the sake of simplicity in the GUI implementation, GUI elements are always rendered to reflect the state of the database. This led to a design in which the database client stores two different states:

- The *requested state* takes into account requests that have not yet been acknowledged.
- The *confirmed state* only includes modifications that have been acknowledged by the server.

Figure 3.3 and Figure 3.4 illustrate the interaction between the GUI and the requested and confirmed states during creation of an object and modification of a property. Conceptually, the requested and confirmed states are both complete descriptions of the database and they are treated as such in the examples. The two states may include different elements and corresponding properties in the two states may have different values. In practice, the two states are either identical or very similar and the requested state is encoded as a set of pending changes to the confirmed state.

By default, a database query returns the requested state. This means that when a GUI element is painted on the screen the position, size and other attributes automatically reflect the requested state. If the database client receives an acknowledgement indicating that a request is denied, it sets the requested state back to the confirmed state. This means that for a denied request, the GUI will temporarily reflect the requested state until the acknowledgement arrives, and then the GUI will automatically “snap back” to the confirmed state.

1. The user of GUI A double clicks to create a note. When the GUI layer instructs the database client to create an element, the requested state is updated immediately so the new note appears in the GUI right away. A creation request is sent to the server.



2. The server processes the creation request and adds an element to its own internal state. The server sends an acknowledgement to database client A and a broadcast to database client B. When they receive the respective messages, the confirmed states for both A and B contain the new element and the note appears in both GUIs.

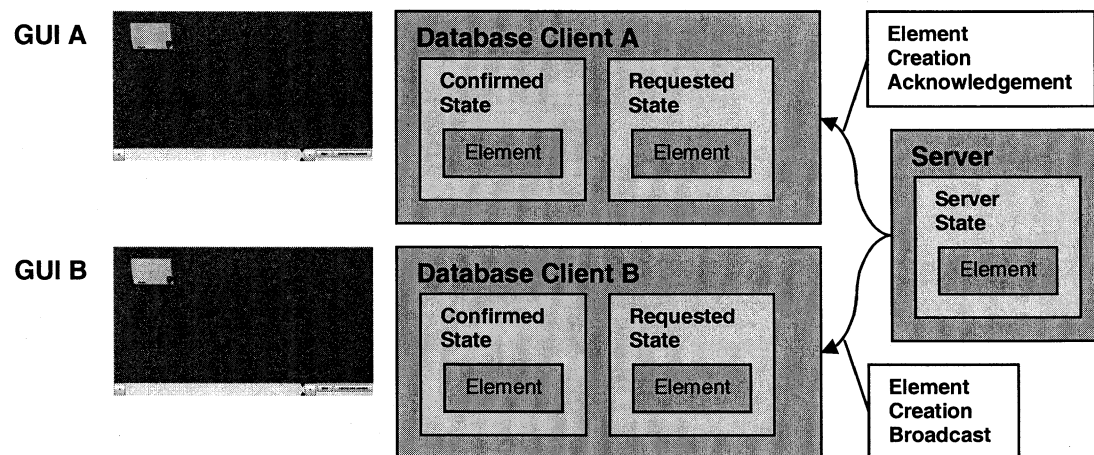
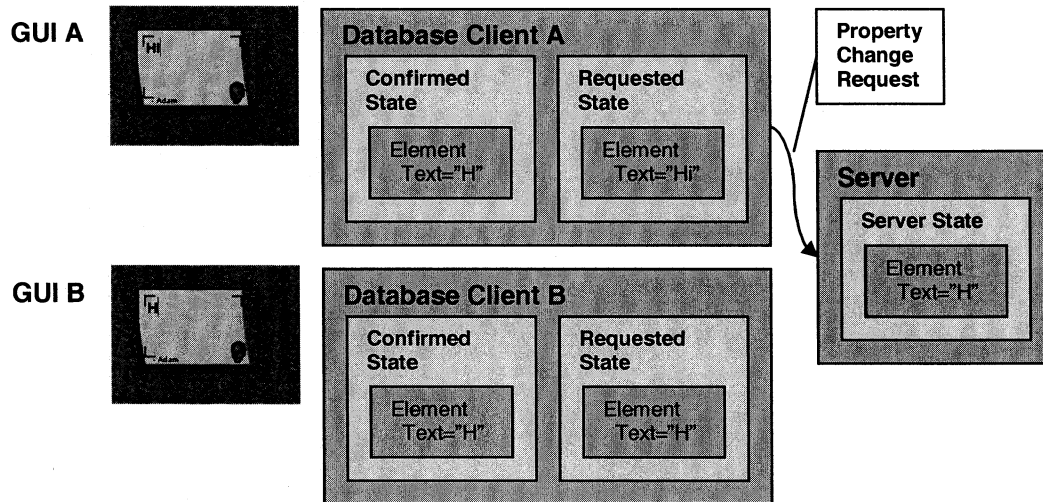


Figure 3.3. When a note is first created, the element is contained in the requested state for the client that created it and a creation request is sent to the server. The server processes the message, sends an acknowledgement to the client that requested the creation, and sends a broadcast to all other clients. When the clients receive these messages they add the element to their confirmed states.

1. The user of GUI A types a letter in a note. When the GUI layer instructs the database client to change the text property, the requested state is updated immediately so the new text appears in the GUI right away. A property change request is sent to the server.



2. The server processes the property change request and changes the property value in its own internal state. The server sends an acknowledgement to database client A and a broadcast to database client B. When they receive the respective messages, the confirmed states for both A and B contain the new property value and the new text appears in both GUIs.

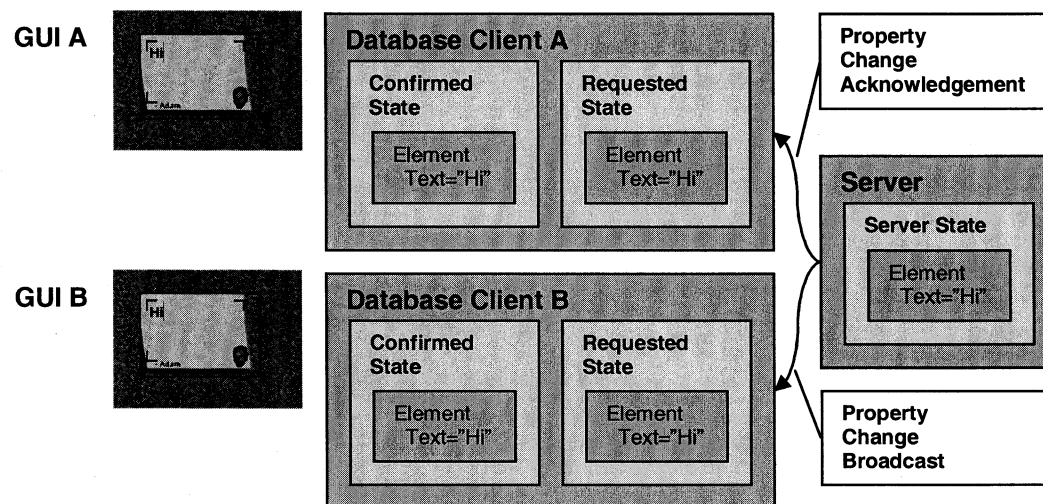


Figure 3.4. When the text of a note is changed, the text property is changed in the requested state for the client that initiates the change and a property change request is sent to the server. The server processes the message, sends an acknowledgement to the client that requested the change, and sends a broadcast to all other clients. When the clients receive these messages they change the property value in their confirmed states.

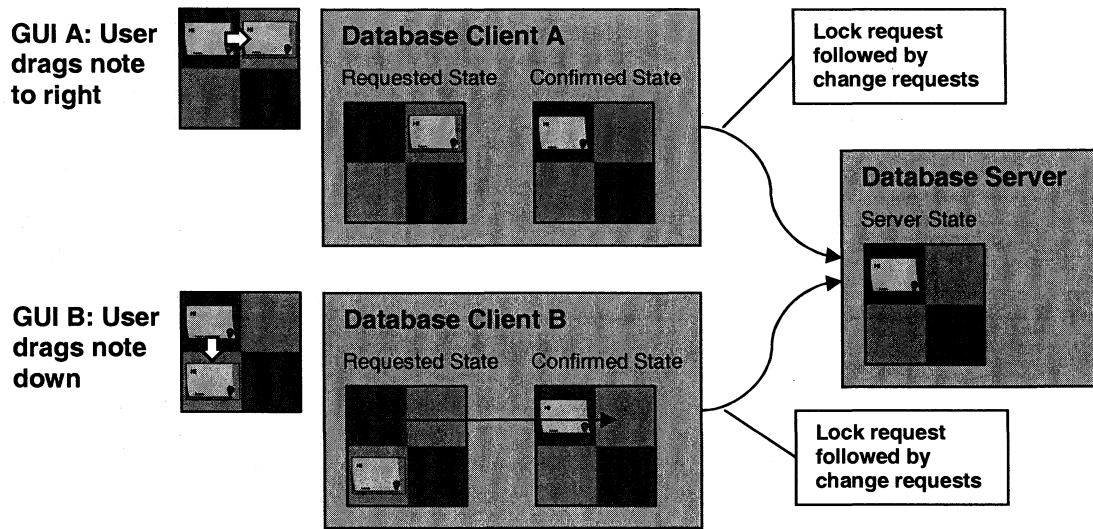
3.3.4 Locks and Simultaneous Operations

Though MessyBoard is primarily intended for asynchronous use, users may attempt to simultaneously drag or resize an object. It is possible for two users to begin an operation on the same object before either of their clients receives notification that their operation has succeeded or failed. Thus, the system is required to provide immediate feedback to both users that their operation has begun, decide which operation will succeed and which will fail, and provide feedback as soon as possible to the user whose operation failed. In addition, as little of this complexity as possible should be exposed to the GUI layer in order to allow for rapid prototyping of interactive behavior without worrying about possible failures in every state.

These requirements are satisfied by a locking mechanism in the database layer. When the user begins an interactive operation, such as dragging a note, the client application requests a lock for the element representing that note. The note's position is updated in real time as the user moves the mouse. When the user releases the mouse button, the client unlocks the element. The GUI layer determines when locks are requested and released and it is responsible for ensuring that all locked elements are eventually unlocked.

Semantically, the lock and unlock requests serve to designate all of the changes in between as a single operation that cannot be interrupted by another client. If the lock request succeeds then all of the requests before the unlock request will also succeed. Likewise, if the lock request fails then all requests before the unlock request will also fail. These semantics allow the GUI to lock, make changes and unlock without any special cases for lock failure during or after an operation. Notification of a lock failure updates the requested and committed states of the client database so that user gets the desired feedback as illustrated in Figure 3.5.

1. The users A and B drag a note in different directions simultaneously. Each GUI renders immediate feedback. Each database client changes its requested state and sends a lock request followed by property change requests to the server.



2. The server processes the lock request from A first and indicates success in the acknowledgement. The lock request from B is denied and failure is indicated in the acknowledgement. When the clients receive the acknowledgements, their requested and confirmed states are changed to the state requested by A. Property changes requested by A are now implicitly accepted and property changes requested by B are implicitly rejected. Note that GUI B will continue to request property changes until the user releases the mouse button, even while it renders the changes requested by client A. This happens naturally without any special code in the GUI to handle lock rejection.

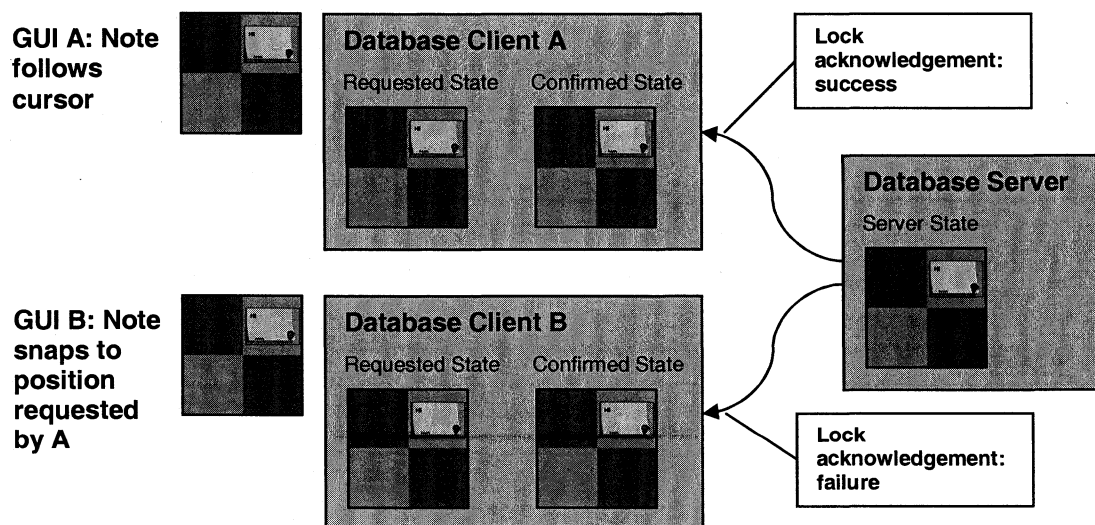


Figure 3.5. Two users simultaneously attempt to drag a note in different directions.

Locking is used for interactive dragging because this operation involves a stream of property changes over time and intuitively it makes no sense for two users to be able to simultaneously move an object to different locations. Other operations, such as changing the color of a note, do not use locking because they are instantaneous operations that involve only a single property change. Typing text in a note is an interesting case: multiple keystrokes can be thought of as parts of a single interactive operation, but there is no reason in principle why two users should not be able to simultaneously insert text into a note. Thus, locking is not used when the user types text in a note. If two users type text in a note simultaneously, the server will determine the order in which their individual keystrokes arrive and both clients will eventually be updated to reflect that order. Though the current GUI implementation behaves badly when this occurs, causing the cursor to jump to unexpected locations, there is no reason in principle why two users cannot edit the same note simultaneously with two separate cursor positions.

3.3.5 Real-time Updates and Un-guaranteed Requests

When the user drags a note to move it, the GUI layer requests a property change every time it processes a mouse movement event. This is desirable because the requests change the position of the note in the requested state, thereby causing the GUI to immediately render the note in a new position. However, if the database client were to send a message to the server for each mouse movement event the network would be flooded with packets and the server would fall behind in processing them.

In order to conserve bandwidth, some updates are labeled as *un-guaranteed* by the client making the request. Un-guaranteed updates are used during interactive operations such as updating the position of a note while the user is dragging it. The database client automatically filters these updates, discarding some of them so that a network message is not sent for each individual mouse movement event. The GUI layer ensures that each interactive operation ends with a guaranteed update so that the server is notified of the final state.

3.3.6 Large Data Objects

MessyBoard allows users to post pictures on the board and to share arbitrary files, represented on the board as icons. These large data objects can take seconds or even minutes to upload to the server and then download to an arbitrary number of other clients. It is important that the interface remains responsive during these transfers, as the user who posted the picture or file will be focusing an unusual amount of attention on the medium at that very moment. This is a likely time for that user to post quick responses to other content or engage in playful behavior. Other users might occasionally be using MessyBoard when the large objects are posted, and they should be able to continue their activities while the data is downloaded from the server.

When transferring large data objects, it is also important to provide feedback to the user about the progress of the transfer. Users may become anxious if they are not sure how long they need to wait (Myers, 1985), and a user may be waiting for an upload to complete before terminating a modem connection.

Property values are not designed to meet these requirements, so the database layer provides an alternative mechanism called *streaming data transfer* for associating large data objects with properties in the database. The definition of “large data object” might sound like an arbitrary threshold, but in practice there is a marked difference in the size of objects that are stored as property values and objects such as files and pictures that use the alternative mechanism. For example, an unusually large note that covers the entire MessyBoard space contains approximately 5,700 characters and is encoded in the database as 5,700 bytes. A relatively small 320 x 240 image of an outdoor scene with aggressive JPEG compression occupies 15,000 bytes. In practice, notes are usually much smaller and images can be much larger.

The URL data transfer mechanism is illustrated in Figure 3.6. The database client maintains two separate connections to the server in order to keep the interface responsive while data is uploading. The database connection carries database update requests and acknowledgements from the server as discussed previously. The stream connection is used by the client only for transmitting large data objects. Thus, a user can create and

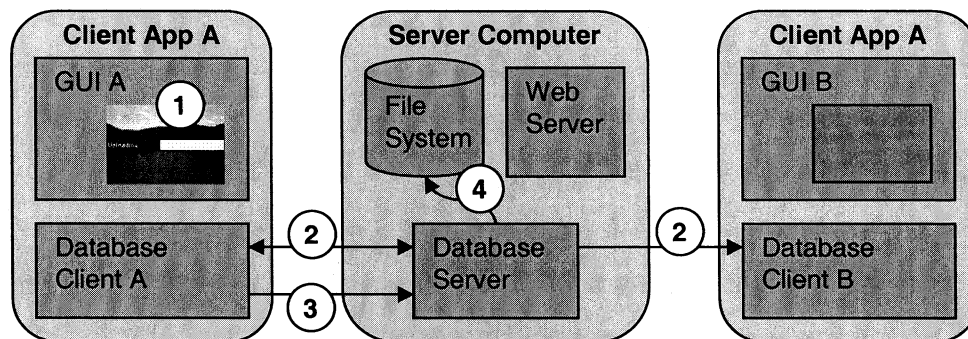
modify a note (using only the database connection) while a large file is uploaded over the stream connection by a separate thread.

When the database server receives a large data object it stores it as a file accessible through a web server and it sets a property in the database to the URL of the file. The client waits for the URL property to be set and then uses the http protocol to retrieve the file. This approach allows the client to leverage the rich functionality provided in the Java standard libraries to download and display images asynchronously. The Java library also provides extensive support for downloading files and displaying progress indicators without blocking the GUI thread.

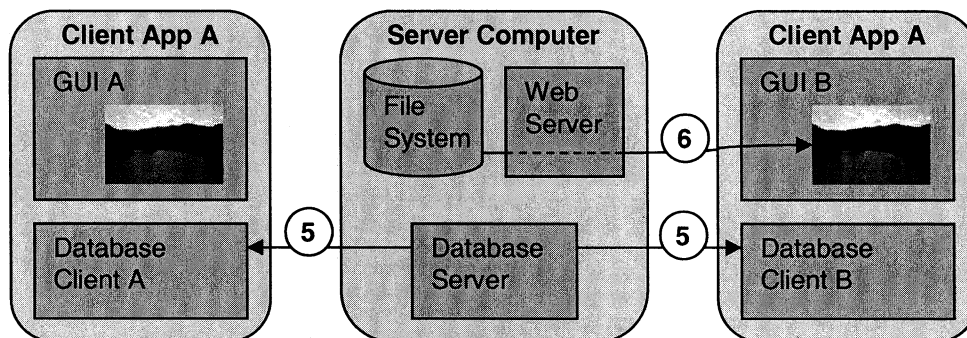
3.3.7 History

MessyBoard users must be confident that any information that they store on MessyBoard can be retrieved. The history architecture is designed to capture every change that is made to the database in real time. The history is stored as a list of *changes* interspersed with *snapshots*. A snapshot specifies the entire state of the database (the set of elements and their properties) at an instant in time. A change specifies a single modification to the database state, such as a property value change or the creation or deletion of an element. The database state at any time t can be reconstructed by finding a snapshot before time t , making a copy of the state specified by that snapshot, and then applying all of the changes that occur between the time of the snapshot and time t .

The history must be persistent so it is stored on the server. I assume that a user typically wants to look at a portion of the history without waiting for the entire history to download. Thus, a client fetches parts of the history on demand in response to the user clicking on the history slider. The client sends a request for a history interval over a separate network connection reserved for history requests and responses. The server responds by sending a dynamically created snapshot specifying the database state at the beginning of the interval followed by all of the changes up to the end of the interval. (The server can apply changes to an existing snapshot to create a new snapshot representing the state of the database at any point in time.) This sequence is illustrated in Figure 3.7.



1. The user drags an image into the MessyBoard space.
2. Database client A creates an element and receives an acknowledgement. Database client B receives a broadcast and GUI B renders an empty rectangle.
3. Database client A uses the stream connection to upload the image data.
4. Upon receiving all the data, the database server writes it to a file.



5. The database sets a property on the image element to a URL and broadcasts the property value to all clients.
6. On receiving the new property value, GUI B uses HTTP to retrieve the image data from the server file system through a public web server that runs on the server computer.

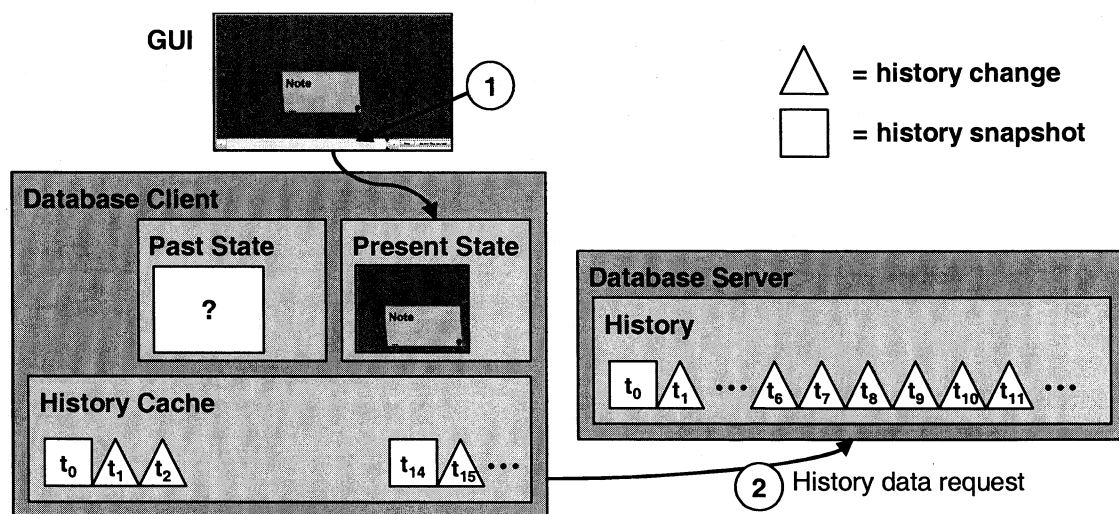
Figure 3.6. Database clients use a separate stream connection to upload large data objects to the server. The database server writes the data to a file and sets a property in the database to a URL. The GUI layer can then use the URL to download the file via HTTP.

It is interesting to note that the history data structure allows for the dynamic creation of snapshots on demand in order to improve future performance. The client's history cache will contain snapshots that correspond to the intervals that the user has requested as demonstrated in Figure 3.7. This policy is intuitively sound, assuming that the user's history requests will be localized with respect to the times in history that they request. Though the new snapshots are dynamically created by the server, the server currently discards them after sending them to the client and the server's history representation always contains only a single snapshot at the beginning. As an optimization, the server could keep some of these snapshots in order to improve performance on future requests.

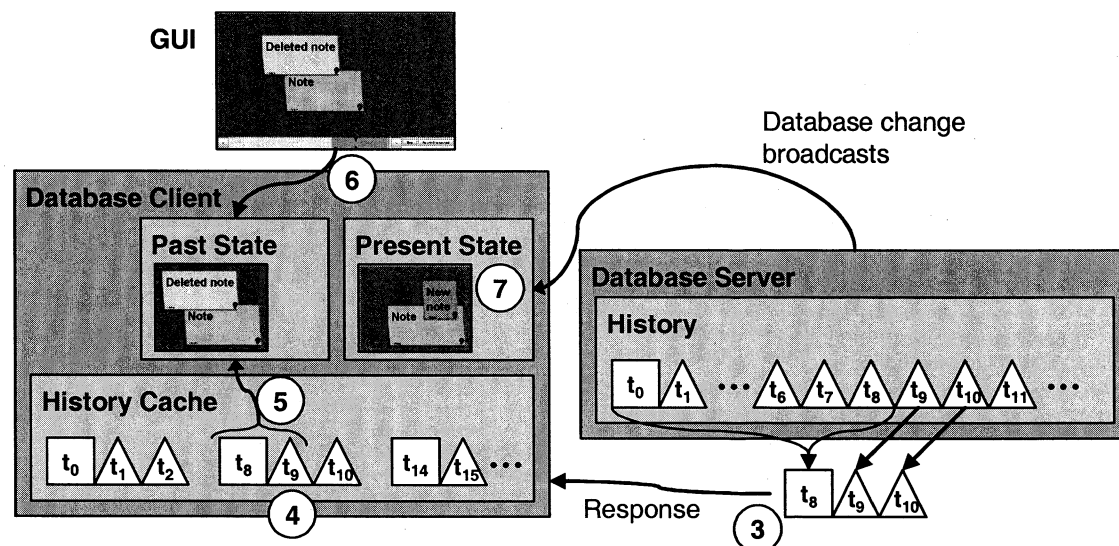
Currently, when the user clicks a time on the history slider the client requests a two-week interval centered at that time. A more sophisticated caching heuristic might improve the user's experience by more frequently having the desired portion of history in local memory before the user views it.

3.3.7.1 History and the GUI

In order to avoid duplicate implementations and preserve consistency in the interface, it is important to leverage a single GUI framework to display both the present and past states of the database. I accomplish this by loading past database states into the database client so that the `MessyArea`, `Model` and `ViewController` objects interact with objects in the past and present through the same interface. When the time changes, the `MessyArea` receives deletion and creation notifications similar to when elements are created or deleted in the present.



1. The user clicks the history slider and attempts to view time t_9 .
2. The database client cannot construct the state at t_9 from the cache, so it requests t_8 - t_{10} from the server.



3. The server applies changes to the snapshot at t_0 to construct a snapshot at t_8 . That snapshot along with changes at t_8 and t_9 constitute the response to the client's request.
4. The client inserts the response into the cache.
5. The client reconstructs the state at t_9 by applying the change at t_9 to the snapshot at t_8 .
6. The GUI views the newly constructed past state.
7. The client silently maintains the present state in response to broadcast messages from the server.

Figure 3.7. When the user clicks on the history slider, the database client requests history data from the server to construct the past state.

While leveraging a single GUI framework to display past and present states avoids duplicate implementations, some interface behaviors must be different when viewing the past. For example, users can copy objects out of the past but they cannot create a new note as they can in the present. The current time is stored in the database client, allowing the MessyArea to determine whether it is presenting the past or the present and to behave accordingly when it processes input events. In theory, ViewController objects could also query the time in their paint methods so as to change their appearance based on the time.

3.4 The Network Layer

The entire MessyBoard client-side architecture is structured to keep the user interface responsive and mask network latency while providing real-time updates for actions that are carried out by remote users. The network layer facilitates this by providing an event-based abstraction on top of Java's TCP/IP sockets. The network client encapsulates a worker thread that waits for incoming messages and puts them in a queue. Another worker thread waits for messages to be placed in the queue and then updates the database client state and posts a repaint event to the AWT event queue. Thus, the GUI thread is notified in a timely manner and it never blocks while waiting for network messages.

3.5 Implementation and Deployment

It is important that users are able to use MessyBoard instantly without installing any software. Currently, the best way to ensure this is to build an applet using Java 1.1. However, drag-and-drop and copy-and-paste require Java 1.2 or greater. My solution is to build the core of the client with Java 1.1 and build extensions using a more recent version of Java. The main program queries the system at runtime to determine the version of the Java Runtime Environment and loads the appropriate classes. This allows MessyBoard to run in any Java virtual machine, though features such as drag-and-drop and history will not always be available.

Users can always upgrade their Java VM to get these features and they may be more likely to do so after they have used MessyBoard and seen some value in it. MessyBoard currently displays a message where the history slider ought to be if it detects that it is

running in a Java 1.1 VM. The message informs users that the history feature will be enabled if they upgrade. It would also be desirable to detect when users are attempting to use drag-and-drop and prompt them to upgrade at that moment, but this is not currently done.

Some MessyBoard users may want to run MessyBoard as a stand-alone application rather than as an applet in a web browser. A stand-alone application can start faster and take up less screen real-estate. I deployed MessyBoard both as an applet and as a stand-alone application using Java Web Start (Sun, 2005b). The Java Web Start client automatically downloads a new version of the program if it has changed, otherwise it uses a cached local copy. It also allows the user to create a shortcut to the application on the desktop.

The MessyBoard screen saver is coded mostly in Java for future portability across platforms. I have implemented a working screen saver for one platform, Windows 2000/XP. The MessyBoard screen saver for Windows is a small executable that launches Java Web Start and directs it to the JAR file hosted on www.messyboard.org. This scheme ensures that the screen saver will be upgraded automatically in the same fashion as the stand-alone application. Similar schemes should be possible on other platforms.

The MessyBoard implementation uses several third-party libraries and components. These components and their uses are listed in Appendix A.

3.5.1 Technical Hurdles to Reducing Startup Cost

In most respects, Java technology is a tremendous boon to any developer who is concerned with reducing startup cost. A Java applet can be embedded in a web page and most web browsers will download and run the applet without any software installation. A developer can set up a Java Web Start application such that end users can install it by clicking on a link in a web page. In either case, the developer can upgrade the application simply by placing new files on the Web and end users will automatically get the new code the next time they run the application.

For applications and applets that require unrestricted access to the local computer's file system, the Java run-time environment displays warning dialog boxes as shown in Figure

3.8. For users that understand these dialog boxes, the startup cost is minimal: just a single mouse click. The Java Web Start dialog box only appears once, and the applet dialog box will only be shown once if the user clicks the “Always” button. For the vast majority of users who are not familiar with the relationship between applets and the local computer, these dialog boxes can increase startup cost substantially by adding worry or annoyance that can translate into delay or complete failure to proceed. Even if the applet/application has a proper digital signature verified by one of the major certificate authorities, the wording in the dialog boxes is serious and alarming and users who do not understand what it means may refuse to run the program. Care-free users who are used to clicking through dialog boxes without reading them may pay extra attention to the Java warnings because the look and feel is different than typical Windows dialog boxes. Users who do find the “Yes” or “Ok” button and click it immediately without reading the rest of the dialog will miss the “Always” button in the applet dialog so they will see the warning every time they run the applet. I have observed this behavior a number of times. One user in particular listed this as her biggest complaint with MessyBoard, and she was astonished when I pointed out the “Always” button.

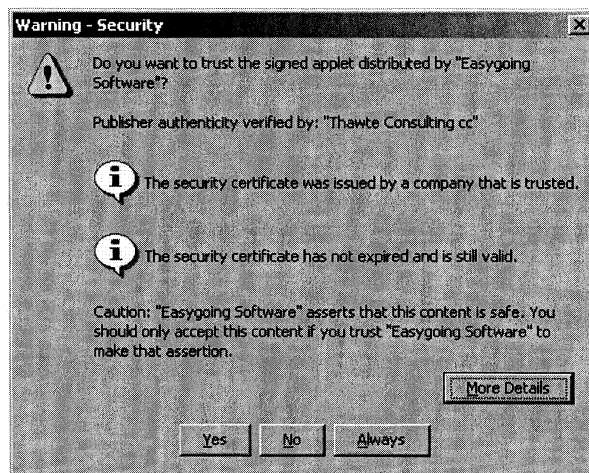


Figure 3.8. The Java run-time environment displays an unusual and alarming dialog for applications and applets that request unrestricted access to the hard drive, even if they are signed with a digital certificate from a major certificate authority.

Solving this problem for a global audience of MessyBoard users is not possible with the current Java security framework. A digitally signed applet or application verified by a Certificate Authority only assures the end user that the code comes from a particular source and that it has not been altered in transit. The user must decide whether or not they trust the source.

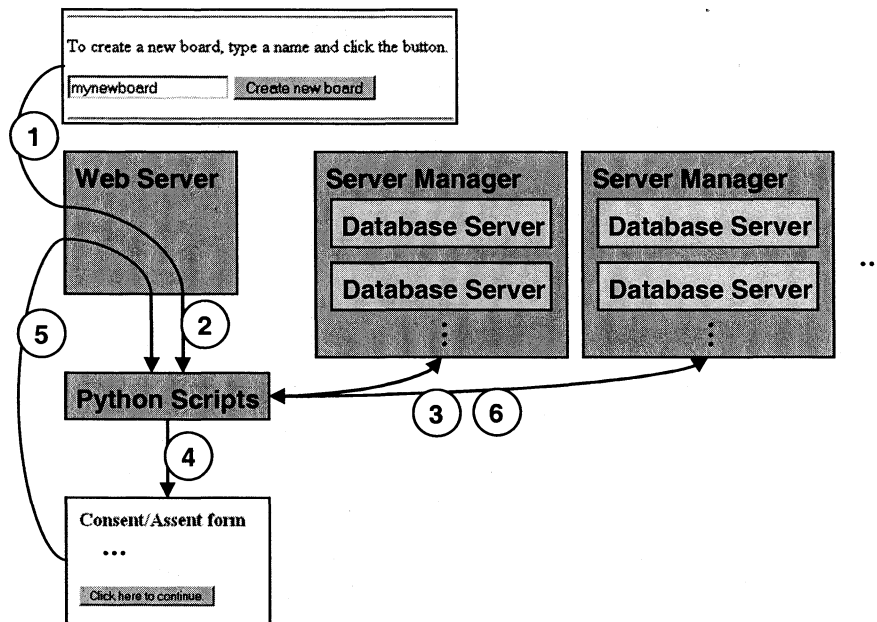
The problem may be more tractable for an enterprise deployment within an organization. The web browser and Java VM for each user's workstation could be configured (perhaps with some modifications) to accept digitally signed applets from certain sources automatically, and the organization could sign the MessyBoard applet and application with its own certificate. However, this requires that the organization is able to easily change the configuration of each individual workstation.

3.6 The MessyBoard Web Site

The MessyBoard web site, hosted at www.messyboard.org, allows users to create their own MessyBoard spaces and access them. The main software components of the server are the MessyBoard *server manager* and the Apache HTTP server (Apache Software Foundation, 2005a). The server manager encapsulates servers for several individual MessyBoard spaces in a single process and it can create additional servers. The web server uses the `mod_python` module (Apache Software Foundation, 2005b) to run Python (van Rossum & de Boer, 1991) scripts in response to URL requests. These Python scripts perform server-side processing of CGI form data in a similar manner to CGI scripts. `Mod_python` provides added flexibility by allowing all URL requests to be processed by a script that takes the requested URL as input and generates arbitrary output dynamically. The Python scripts use TCP/IP sockets to communicate with the MessyBoard server manager.

3.6.1 Creating a MessyBoard Space

The most common way to create a new MessyBoard space is to go to the main page at www.messyboard.org, type a name in the box and click the button. Figure 3.9 explains how the server components interact to create a new MessyBoard space and provide feedback to the user.



1. The user types a name for a new space and clicks the button, sending a CGI request to the web server.
2. The web server invokes a Python script to process the CGI request
3. The Python script queries server managers to check that the name is not already in use
4. The Python script returns a consent form with hidden state information
5. The user completes the form and clicks the button, sending another CGI request to the web server
6. The Python script handling the CGI request instructs one of the server managers to create the new server with the requested name

Figure 3.9. A web server launches Python scripts that interact with server managers and generate dynamic HTML content in order to allow the user to create a new MessyBoard space and sign a consent form.

When the user types a name and clicks the button, the name is sent to the web server which invokes a server-side Python script. The script opens sockets to all MessyBoard

server managers and queries them to determine whether or not the requested name is already in use. If it is not, the script returns a consent form that is displayed in the user's browser. The user fills out the consent form and clicks the "submit" button which sends another URL request to the server. Upon receiving the request, the server runs another Python script that queries the MessyBoard server managers to determine which one is currently running the fewest MessyBoard servers. The least loaded manager is instructed to create a new server with the given name. When the server is created, the user's web browser is redirected to the new MessyBoard space automatically.

The user may also initiate the creation of a new MessyBoard space by typing the URL for a space that does not yet exist. All URL requests are handled by a Python script which checks to see if the requested space exists or not, using the same mechanism described above. Thus, it is a simple matter for the script to inform the user that the space does not exist and enter the creation sequence described above.

The MessyBoard server must ensure that all users fill out the consent form once but it would be highly undesirable to require that users fill out the consent form each time they access a MessyBoard space. The server-side Python scripts read and write cookies in order to determine whether or not a user has already filled out a consent form. Consent forms are tracked separately for each MessyBoard space, so a user that uses two MessyBoard spaces will have to fill out two forms. All of this is tracked in a single browser cookie encoded with fields for each MessyBoard space.

For this project, the consent forms are an unfortunate aspect of the ethical requirements on research involving human participants, since this project places a huge premium on lowering the effort threshold to adoption and use, and the consent form itself is a huge disincentive for users to proceed.

3.6.2 Scalability

At present, the public MessyBoard deployment hosts a modest number of MessyBoard spaces (under 300) and the load is distributed over two computers. This section describes

how this is accomplished and how the current strategy might scale to host thousands of MessyBoard spaces.

The MessyBoard server manager uses memory to store the current state for each MessyBoard space. The Java virtual machine imposes a limit on the amount of heap space available to a program, so it is necessary to distribute all of the servers over several different server managers. The CPU demands also grow as more MessyBoard spaces are added and more users access the space, since each new server launches a few threads that listen for incoming connections and additional threads are launched each time a user connects. For this reason, it is desirable to spread MessyBoard server managers over several different physical machines.

Figure 3.10 shows how MessyBoard server managers are currently distributed over a cluster of two computers on a private subnet. A router using Network Address Translation (NAT) makes the server cluster appear as a single machine at a fixed IP address to the outside world. This is important for the Java applet, since the client-side Java sandbox will only allow Java 1.1 applets to open a socket to the IP address from which the applet was served.

The router directs incoming packets to different machines in the cluster depending on the port that they are bound for. All HTTP requests are directed to the machine that runs the web server. The router directs a fixed range of port numbers to each computer and server managers are configured to listen on ports in the appropriate range. Each individual MessyBoard space has a fixed port. The MessyBoard client gets this port from the HTML page if it is running as an applet or from the `jnlp` file if it is running as an application through Java Web Start (Sun, 2005b). Users are never required to know the port numbers for their MessyBoard spaces.

The server cluster can easily accommodate new physical machines for running additional server managers. Adding a new physical machine requires some configuration of the router, changes to a master list of server managers used by the Python scripts, and configuration of the new machine. Existing machines need not be reconfigured. One could

also imagine different schemes for scalability, such as a single physical machine with a large amount of memory and multiple processors.

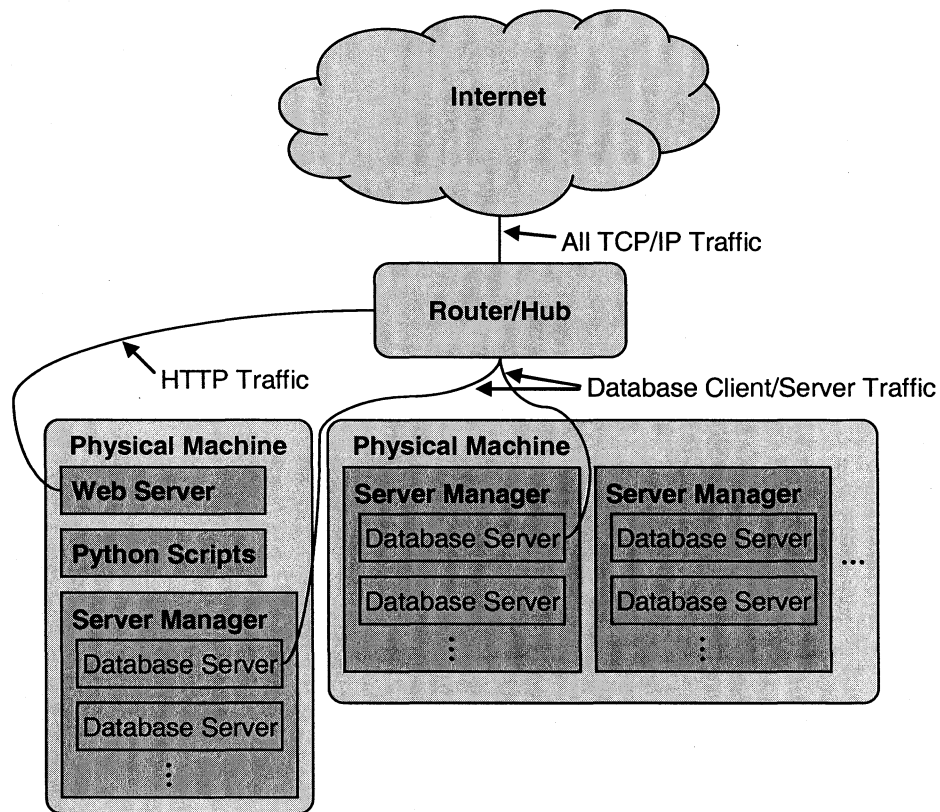


Figure 3.10. A router makes two physical computers appear as one from the outside and it directs traffic to the appropriate computer based on the destination port number.

The only fundamental limit in this scheme is the number of available ports. Since all traffic is redirected by a single router based on the port number, adding new physical machines to the server cluster does not result in additional ports for new MessyBoard spaces. Each MessyBoard space requires three ports (database connection, stream connection and history connection). There are 65536 ports, so no more than 21845 MessyBoard spaces can exist on a server cluster at a given IP address. This restriction could be lifted if all end users are required to use a modern version of Java, since signed applets in a Java 1.2 and later can open sockets to any computer if the user grants permission. One could also imagine a custom router that redirects traffic based on the content of the packets, thereby removing the limit on the number of MessyBoard servers while maintaining

the outward appearance that the server cluster is a single physical machine at a fixed IP address.

3.6.3 Practical Problems with a Monolithic MessyBoard Server

The current MessyBoard deployment hosted at www.messyboard.org has several practical problems. The name of each MessyBoard space must be unique, but inevitably two groups will want the same name. One group will be forced to choose a less intuitive or memorable name, thereby increasing the startup cost. Worse yet, if a MessyBoard space with a simple and popular name has no password protection, many users may view it by accident.

All MessyBoard traffic is sent over the public Internet to the central server, introducing security risks. This traffic could be encrypted to obscure the content, but an attacker could still learn when different people are using the MessyBoard space by intercepting messages sent between the client and server. If group members work at a single site or campus with a firewall, it is desirable to allow them to use MessyBoard without sending traffic over the public Internet.

A MessyBoard space is publicly accessible by default to anyone who knows the URL and MessyBoard has no mechanism for user authentication. Part of the reason for this is reduced startup cost: Users should be able to start using MessyBoard without creating an account and deciding on a password. However, many groups need to keep the contents of their MessyBoard space hidden from the public even if they trust other members of their group. As a stop-gap measure I have implemented an optional protection scheme that protects a MessyBoard space with a single password. This leads to other problems: Users will be tempted to e-mail the password to each other, and if the password is compromised then all of the users will have to agree on a new password and learn it. A pragmatic approach is for users to choose an obscure name that is difficult to guess or stumble upon, effectively acting as a password. However, this approach eliminates all of the benefits of an intuitive and memorable name and URL for the MessyBoard space.

MessyBoard users have no physical access to the MessyBoard server, which would raise security and legal concerns in many organizations. Given the choice, many organizations would rather hire an IT staff and pay them to maintain such a server than trust a third party with the storage and protection of sensitive data.

The solution to all of these problems is an enterprise MessyBoard server. If members of a large organization are convinced of the value of MessyBoard, they could set up their own MessyBoard server on site and allow members of the organization to use it. Corporations and other organizations already manage their own enterprise servers for e-mail and they are beginning to do this for instant messaging programs (Lawton, 2003). An enterprise MessyBoard server could offer all of the functionality currently provided by messy-board.org including low-cost methods for creating a new MessyBoard space.

Separate servers would allow groups in different organizations to use the same names for their MessyBoard spaces. For example, if the design groups at companies Alpha and Beta both wanted a MessyBoard space named “design”, the two spaces could both exist at the following URLs:

- www.alpha.com/messyboard/design
- www.beta.com/messyboard/design

Both URLs are simple and memorable for the members of their respective organizations.

An enterprise MessyBoard server could reside on a private subnet behind a firewall so that an in-house IT staff could administer and maintain it. For users working at the same site or campus, all MessyBoard traffic would be confined to the organization’s subnet. An enterprise MessyBoard server could leverage existing infrastructure for authentication and access control. For example, users could sign on to a MessyBoard server using their UNIX or Windows domain user name and password, and existing permissions and groups could be used to control access to MessyBoard spaces.

Chapter 4

Related Work

4.1 Shared Drawing Applications

Shared drawing applications are networked programs designed primarily to support synchronous collaboration between either co-located or remote participants (Centra, 2005; Greenberg, Bohnet, Roseman, & Webster, 1992; Groupboard, 2005; HelpMeeting.com, 2005; Hindus, Mainwaring, Leduc, Hagström, & Bayley, 2001; JDH Technologies, 2005; Microsoft, 2005a; Moran, McCall, van Melle, Pederson, & Halasz, 1995; Netscape, 2005; RTZ Software, 2005; Stefik, Bobrow, Foster, Lanning, & Tatar, 1987; Trimble, Wales, & Gossweiler, 2002; WebEx, 2005; Wolf, Rhyne, & Briggs, 1995). Typically, the tools provide a 2D WYSIWIS space and support some subset of these features: freeform sketching, bitmaps, text, shared telepointers, scrolling over a large area, multiple pages, and marking up an existing web page or document.

These tools are designed to support synchronous meetings where people are in the same room or communicating by voice using a telephone or the Internet. Their interfaces are optimized for rapid visual communication of ideas in real time, augmented by voice or text chat, though one could conceivably set up a dedicated server for one of these programs in order to allow asynchronous use. It is possible to run any of these programs on a public display, but none of them function as a screen saver. Only one of these programs (We-Met) allows users to browse a complete history of activity (Wolf et al., 1995).

4.2 Shared Text Editors

Shared text editors such as ShrEdit (Dourish & Bellotti, 1992) and MoonEdit (Dobrowolski, 2005) allow multiple users to edit a single document, each on their own networked computer. These tools are designed to support only synchronous collaboration and the only medium they support is text.

4.3 Chat (Instant Messaging)

Chat programs such as AOL Instant Messenger (AOL, 2005) and RVM (Herbsleb, Atkins, Boyer, Handel, & Finholt, 2002) typically allow two users to type messages back and forth to each other synchronously. Some chat programs allow users to exchange pictures and files. Although some chat programs allow more than two users to participate in a discussion, the programs are typically used for two-way communication. Chat sessions are not persistent and the medium does not provide a 2D space.

4.4 Large Public Displays

A few researchers have built systems that use large public displays to support focused, time-critical collaboration. The MERBoard system (Trimble et al., 2002) is designed to help NASA scientists analyze data from a rover on Mars, and the eWhiteBoard system (Bercowicz, Barnett, & Chueh, 1999) supports scheduling in a cardiac catheterization center. Neither of these systems provides a networked freeform 2D space designed for asynchronous collaboration (MERBoard includes a shared drawing application).

The Portfolio Wall system allows automobile designers to display works in progress on large public displays (Buxton, Fitzmaurice, Balakrishnan, & Kurtenbach, 2000). Rather than providing a WYSIWIS space, the Portfolio Wall display automatically cycles through content in a repository and allows users to add or access content using special scanners and tags.

Dynamo allows users in a single room to quickly share content, and users can leave notes to communicate asynchronously (Izadi, Brignull, Rodden, Rogers, & Underwood, 2003). Dynamo is designed primarily for synchronous collaboration between colocated users. Users must go to the room where Dynamo is displayed in order to interact with it..

Many researchers have attempted to design software systems that facilitate informal information sharing, scheduling, planning and casual conversation. Two key issues are: 1) making sure that the users do not forget about the systems, and 2) making it very little work for users to view and author shared content. A number of systems use large displays (projectors, plasma screens or augmented whiteboards and tack boards) in public settings

to accomplish these goals (Churchill, Nelson, Denoue, & Girgensohn, 2003; Grasso et al., 2002; Houde, Bellamy, & Leahy, 1998; McCarthy, 2002; McCarthy, Costa, & Liongosari, 2001; Moran et al., 1999; Rogers & Brignull, 2002; Russell, 2002; Russell, Trimble, & Wales, 2002). Other systems use smaller displays in key locations, such as outside of office doors or meeting rooms (Cheverst, Fitton, Dix, & Rouncefield, 2002; McCarthy et al., 2001; Nichols, Wobbrock, Gergle, & Forlizzi, 2002; O'Hara & Perry, 2002). None of these systems provide a networked freeform 2D space.

4.5 Screen Savers

Researchers and companies have created several different kinds of screen savers that deliver interesting information to the user. Some screen savers support peer-to-peer collaboration, meaning that a user can contribute content that appears on another user's screen saver (MyCorkboard, 2005; Netpresenter, 2005; Oceanview Consultancy, 2005; Spreitzer & Theimer, 1993). Other screen savers are designed to broadcast information from a central source (Birch Grove Software, 2005; e-motional.com, 2005; Jackson & Johansson, 1997; Netpresenter, 2005). Another category of screen savers acts as message boards, displaying away messages and idle times and in some cases allowing visiting coworkers to leave messages on a person's computer when the person is not in her office (Bellotti & Bly, 1996; Birch Grove Software, 2005). None of these screen savers provide a freeform 2D WYSIWIS shared space.

Shiozawa et al. have built a screen saver that displays a shared XWindows desktop (Shiozawa, Okada, & Matsushita, 1999). Their architecture is based on VNC (AT&T Laboratories Cambridge, 1999) and the shared desktop can run any UNIX application. This is a clever and powerful approach, but the XWindows desktop is clearly not designed as a lightweight interface for sharing information. In order to display a given picture in the shared space and post comments nearby, users need to run multiple applications and move the windows around. It is not possible to draw freeform ink strokes over existing content, and the system does not maintain a history of user activity.

4.6 Lightweight Asynchronous Collaboration Tools

Wiki Wiki Web (Cunningham, 2005) and CoWeb (Rick, Guzdial, Carroll, Hollaway-Attaway, & Walker, 2002) are systems that allow multiple authors to edit a shared web page using only a web browser and no additional software.

A number of commercial and research systems aim to support collaboration over the World Wide Web (Akiva, 2005a, 2005b; Caucus, 2005; Centra, 2005; Cybozu, 2005; Documentum, 2005; Facilitate.com, 2005; HelpMeeting.com, 2005; Intranets.com, 2005; JDH Technologies, 2005; Klöckner, 2000; Lachoff, 2001; Opentext Corporation, 2005; RTZ Software, 2005; Smart Technologies, 2005; Thruport, 2005; WebEx, 2005). These systems typically comprise a subset of the following features:

- Document repository with version control
- Asynchronous and synchronous messaging
- Real-time application sharing similar to VNC (AT&T Laboratories Cambridge, 1999)
- Support for votes or polls
- Shared calendar
- Conferencing (real-time voice chat, video, shared drawing program, shared presentation with real-time annotation)

4.7 Closely Related Systems

A number of systems have several features in common with MessyBoard. This section gives a brief description of each of those systems and explains how MessyBoard differs. Unless stated otherwise, these systems are displayed on a traditional desktop workstation with a standard monitor, keyboard and mouse. Figure 4.1 shows how each of these systems compares to MessyBoard in terms of a few critical features. The remainder of this section discusses each system in turn and compares it to MessyBoard.

	Large Space	Multiple Spaces	WYSIWIS	Runs on any PC	History	Screen Saver	Public Display
MessyBoard			✓	✓	✓	✓	✓
TeamWave Workplace	✓	✓	✓	✓			✓
MuSwikis		✓	✓	✓			
Designers' Outpost			✓		✓		✓
Notification Collage				✓	✓		✓
iCom			✓				✓
What's Happening				✓		✓	✓
ScanBoard		✓	✓				✓
Semi-Public Displays		✓	✓				✓
Groove Pinboard	✓		✓	✓			
Kansas	✓		✓	✓			
netomat		✓	✓	✓	✓		

Figure 4.1. MessyBoard shares some features in common with several similar systems but its combination of features is unique.

4.7.1 TeamWave Workplace

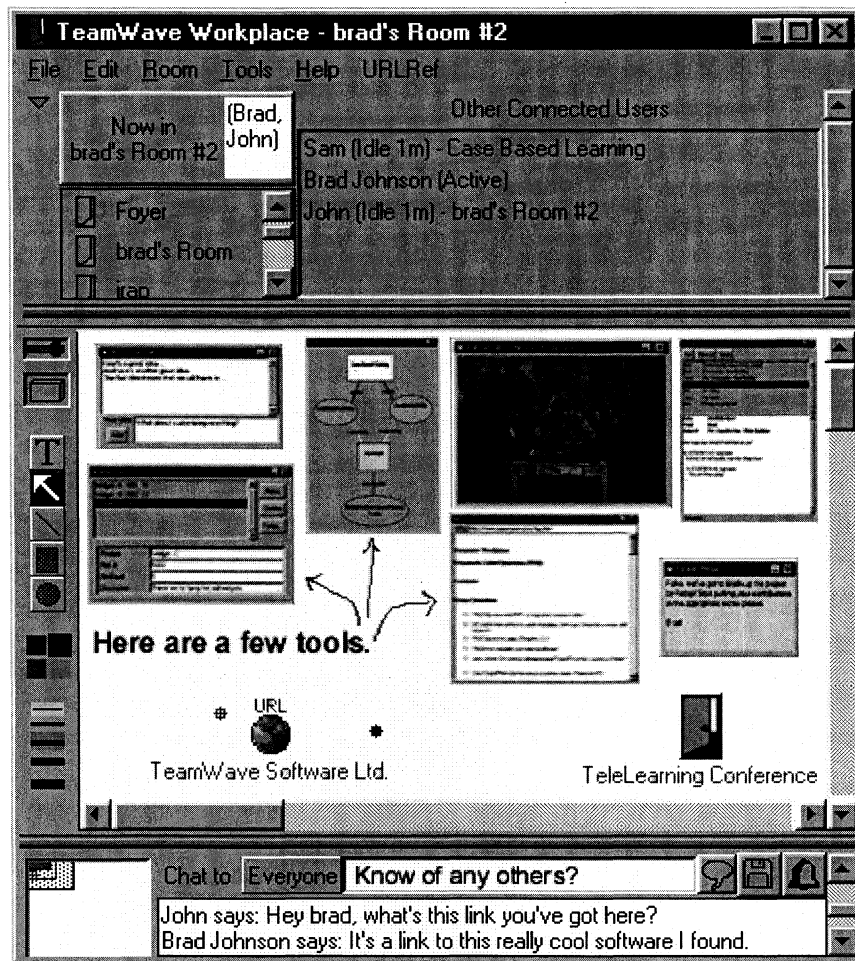


Figure 4.2. Teamwave Workplace. Taken from

http://www.markroseman.com/teamwave/overview_interface.gif

TeamWave Workplace (formerly known as TeamRooms) is a shared freeform 2D WYSIWIS space designed for synchronous and asynchronous collaboration (Greenberg & Roseman, 1998; Roseman & Greenberg, 1996). A group of people shares multiple “rooms”, and each room is a persistent shared space in which users can draw, chat, and manipulate special-purpose “applets” as shown in Figure 4.2. The available applets include simple tools like notes and pictures, as well as more complex tools such as the hierarchical outliner, a database interface, a shared web browser and several games. The system facilitates real-time collaboration by providing awareness information (a list of users

that are currently in the room) and shared telepointers. The TeamWave Workplace server can store an arbitrary number of versions of the shared space, allowing users to access the history of a room.

TeamWave Workplace allows users to spread information among multiple rooms, while MessyBoard users must put all of their information in a small space so that the entire state can be seen in a glance, without any interaction. This design choice may make MessyBoard better suited for public displays and screen savers. TeamWave Workplace has been used on large public displays, but never as a screen saver (Mark Roseman, electronic mail, March 13, 2003).

MessyBoard has a more advanced interface for browsing and visualizing the history of activity on the board. Though TeamWave has the capacity to save an arbitrary number of versions, the authors have not developed an interface that allows users to browse a large history by simply clicking on a time slider (Mark Roseman, electronic mail, May 2, 2003).

4.7.2 MuSwikis

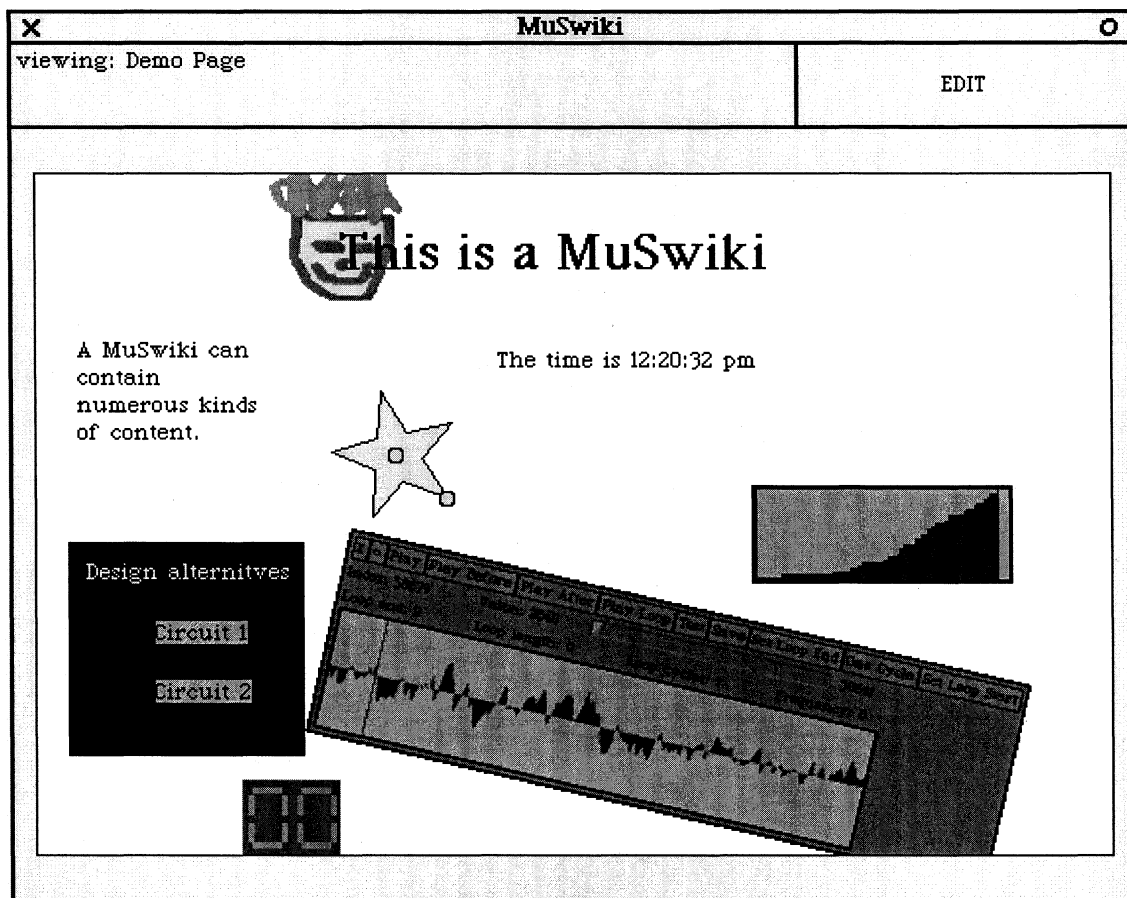


Figure 4.3. A Muswiki. Taken from <http://www.cc.gatech.edu/~lex/muswiki/>

A MuSwiki is a collection of 2D freeform pages designed for asynchronous collaborative learning (Spoon & Guzdial, 1999). Users can modify existing pages or create new pages, and MuSwiki pages are hyperlinked to each other much like HTML pages. Each page is a freeform WYSIWIS 2D space that supports text boxes, bitmap images, freeform drawings, and “computational elements” that support arbitrary kinds of computation and interaction as shown in Figure 4.3. Users do not see each others’ modifications in real time: an author makes some changes and then explicitly asks for them to be broadcast.

MessyBoard differs from MuSwikis in that it has no concept of multiple pages. As noted above, this makes MessyBoard better suited to display as a screen saver or on a large

Designers' Outpost is designed for synchronous collaboration on a specific task, whereas MessyBoard is designed primarily for asynchronous collaboration with a much broader audience in mind. The evaluation of Designers' Outpost has focuses exclusively on synchronous collaboration for short periods of time, and MessyBoard will be evaluated as a tool for asynchronous collaboration over weeks or months.

Designers' Outpost also differs from MessyBoard in that it does not run on a user's personal workstation. Users must go to a dedicated installation with special-purpose hardware to use Designers' Outpost. A user can use MessyBoard on any computer with an Internet connection.

4.7.4 Notification Collage

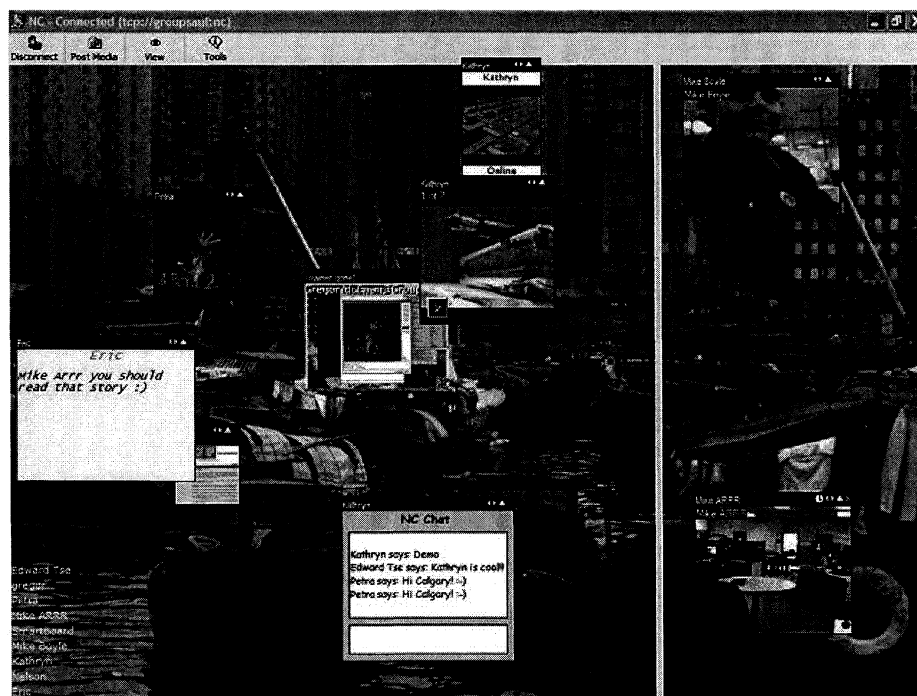


Figure 4.5. Notification Collage. Image taken from http://grouplab.cpsc.ucalgary.ca/software/notification_collage/

Notification Collage is a collaborative 2D bulletin board system for sharing notes, pictures, live video streams, and other media (Greenberg & Rounding, 2001) as shown in Figure 4.5. Notification Collage is not a strict WYSIWIS system: each user can have a

unique arrangement of items on her screen, and users can individually show and hide elements without affecting other users' displays. The authors have created a robust toolkit to facilitate the development of new media elements and modification of the medium itself, including a system to record the history and visualize it (Tang, McEwan, & Greenberg, 2003).

MessyBoard and Notification Collage are designed for similar audiences and to solve a similar set of problems, but the two systems are designed differently. The strict WYSIWIS nature of MessyBoard allows users to use spatial relationships to communicate information, while the relaxed nature of Notification Collage allows users to personalize their own displays and filter information as they see fit.

Notification Collage and MessyBoard are both shown on large public displays in addition to personal workstations, and largely for the same reasons. Notification Collage currently does not run as a screen saver.

4.7.5 iCom

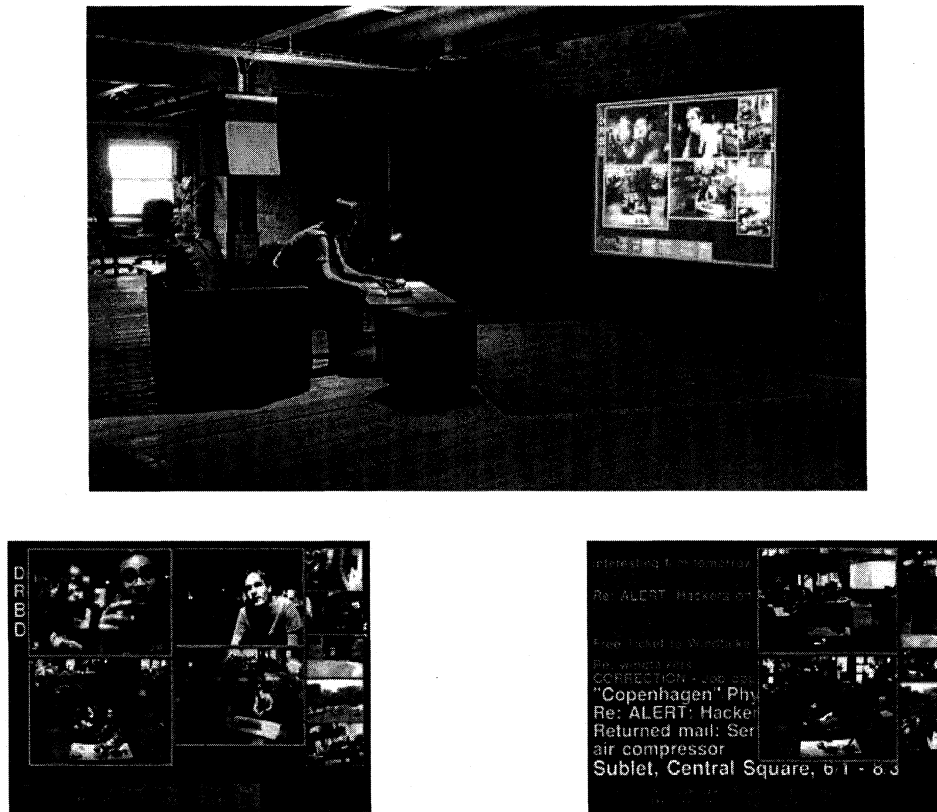


Figure 4.6. iCom. Images taken from <http://web.media.mit.edu/~stefan/hc/projects/icom/>

The iCom system is designed to foster awareness of activities, synchronous conferencing, and asynchronous messaging between different physical locations (Agamanolis, 2002a, 2002b). Each iCom station has a large projected display and a seating area with a trackball. The display contains video streams (one broadcasted from each station) and a community bulletin board as shown in Figure 4.6. The video windows and the bulletin board window can be moved with the trackball and each station shows the same configuration (strict WYSIWIS). Users post content on the bulletin board by sending e-mail messages to a designated address. Each station broadcasts video and audio, facilitating cross-site meetings and casual interaction.

The design of iCom is driven by the desire for a subjective sense of “connection” between different physical locations, whereas MessyBoard is designed more for explicit

communication and collaboration. iCom's always-on audio and video encourages synchronous interactions, but iCom does not allow freeform expression with a direct-manipulation interface. There is currently no way to add pictures or drawings of any kind except by placing them in front of the camera, and even then they are not persistent.

4.7.6 What's Happening

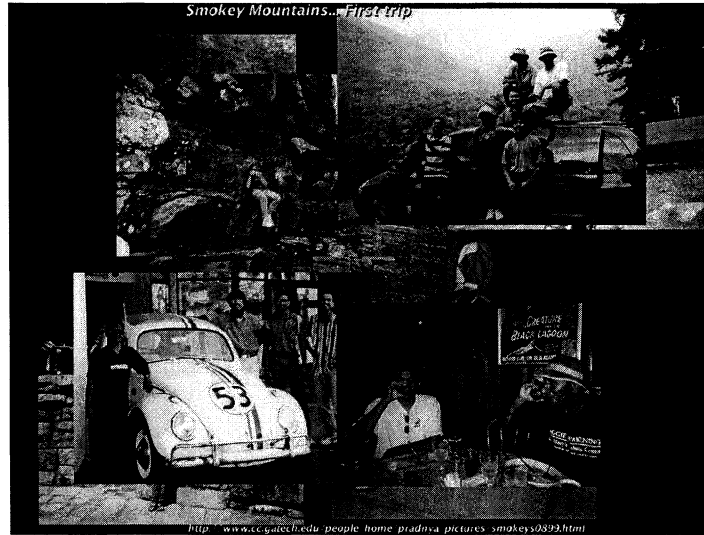


Figure 4.7. A collage generated by What's Happening. Image taken from <http://www.cc.gatech.edu/gvu/ii/community/>

The What's Happening system is designed to foster community awareness for a large organization (Zhao, 2001). The system is made up of two components: a communication bar that displays text messages and a screen saver that presents collages of images. The text messages on the communication bar are automatically taken from a variety of sources including announcements and popular web sites, and users can also submit their own messages. The screen saver collages are created automatically using images from public web sites in the community, as shown in Figure 4.7. Occasional “value added” collages show satellite images from weather sites, maps with traffic conditions, or a random text message from the same set of sources as the communication bar. All collages are assembled and laid out automatically. A single server creates the collages, so all clients see the same collage at any given time. The authors have run the collage screen saver on their personal workstations and on large plasma displays in public areas.

MessyBoard and the What's Happening screen saver are similar in that they both make use of large public displays and screen savers in order to get peoples' attention. What's Happening creates all the content automatically, without the users having to do any extra work. MessyBoard relies on the users to do all of the work involved in content creation, and the users have complete control.

4.7.7 ScanBoard

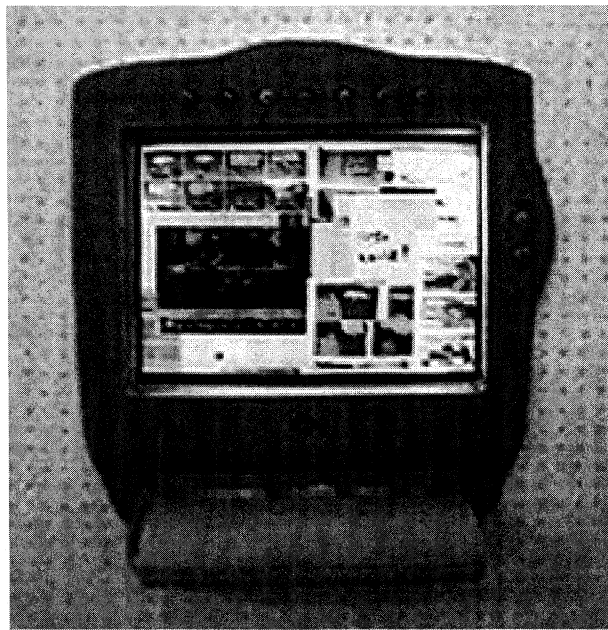


Figure 4.8. Scanboard. Image taken from (Hindus et al., 2001).

ScanBoard is a prototype system designed for communication between homes (Hindus et al., 2001). A single wall-mounted unit comprises a touch screen display and a scanner as shown in Figure 4.8. The display shows a shared WYSIWIS 2D space. Users add pictures or notes to the space by placing pieces of paper in the scanner, and they can move the notes around using the touch screen. The system allows multiple shared spaces, and the user switches between spaces using buttons on the display. Two buttons ("Forward" and "Back") allow the user to temporarily remove recently posted items in order to view the items underneath.

ScanBoard is designed for home use, whereas MessyBoard is designed primarily for collaboration in the workplace. ScanBoard's integrated scanner makes it easy to share existing paper artifacts such as photos and comic strips, while MessyBoard is optimized for composing new information and for importing existing digital information. The concept of an integrated public display and scanner is interesting, and support for low-effort scanning might make a very synergistic addition to MessyBoard.

4.7.8 Semi-Public Displays

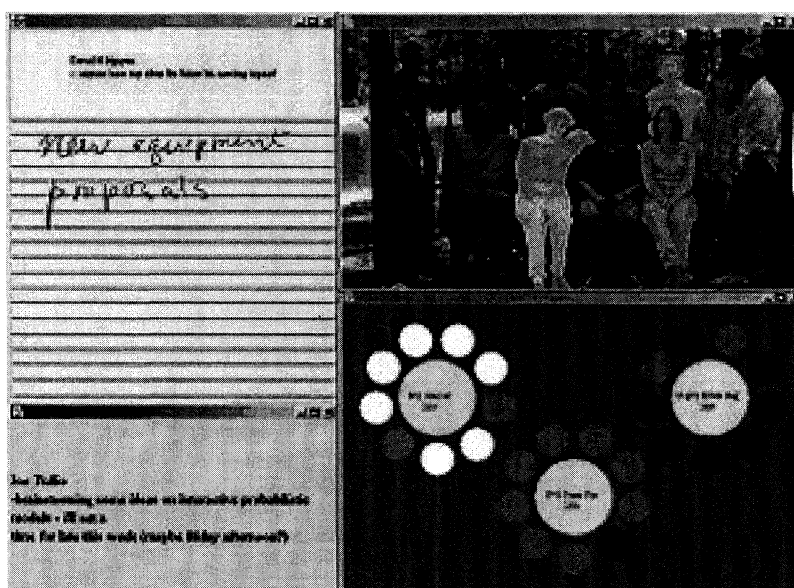


Figure 4.9. Semi-Public Displays. Image taken from <http://www.cc.gatech.edu/fce/ec1/projects/semiPublic/>

Semi-Public Displays are intended to aid collaboration amongst small co-located groups (Huang & Mynatt, 2003). The system displays four separate applications on a large public display in the authors' lab, and each application occupies about a quarter of the space as shown in Figure 4.9. The "Reminders" application displays a slide show of requests and reminders that are automatically parsed out of the group's weekly status reports. The "Collaboration Space" application also cycles through the help requests, and it provides a freeform space where lab members can scribble comments in digital ink using a stylus on the shard display. The "Active Portrait" provides awareness information by displaying a group photo. Each user's keyboard activity is logged, and when a user is not active her

image in the group photo slowly fades. The “Attendance Panel” presents an abstract representation of upcoming events, in the form of flowers, and how many people will be attending, represented by the colors of the petals on the flower. Users touch the petals with the stylus to indicate that they are planning to attend, and that explicit action changes the color of the petal.

MessyBoard and Semi-Public Displays are designed for a similar purpose: to aid small group collaboration. One important difference is that Semi-Public displays run only on a single shared public display, and MessyBoard runs on any number of clients simultaneously. Another difference is that Semi-Public Displays are designed to facilitate specific aspects of collaboration (awareness, help requests, event attendance) and provide content automatically from existing sources, and MessyBoard is a freeform space that relies entirely on the users for content.

4.7.9 Groove Pinboard

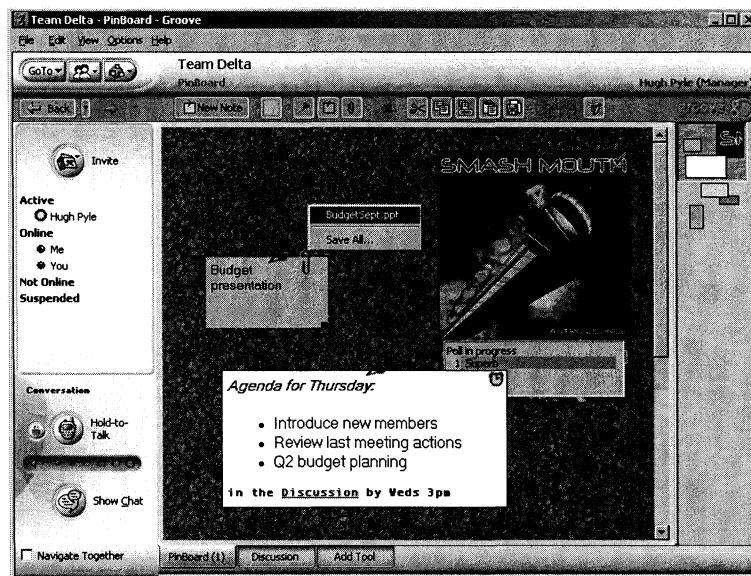


Figure 4.10. The Groove Pinboard tool. Image taken from <http://www.cabecal.com/Products/PinBoard/pinboardscreenshot.gif>

The PinBoard tool (Cabecal, 2005) is a component of Groove Workspace (Groove, 2005), an all-purpose collaboration tool. Groove Workspace integrates asynchronous and

synchronous text messaging, voice messaging, and a number of other collaboration tools in a single application. The PinBoard tool allows users to share and decorate an infinite 2D WYSIWIS space as shown in Figure 4.10. Users create notes, which can have background images and attached files. Notes can also be turned into polls, allowing members to vote on issues. The viewport shows a portion of the whole space, and a miniature view shows the entire board and the current location of the viewport.

MessyBoard provides a small finite 2D space, and PinBoard provides an infinite space. The infinite amount of space in PinBoard may encourage users to spread notes out over a vast area, increasing the receiving cost and making it less suitable than MessyBoard for public displays.

Though an entire Groove workspace can be saved as an archive at any time, this is an explicit and fairly heavyweight operation. The PinBoard tool has no lightweight history mechanism for browsing and recovering old content.

4.7.10 Kansas

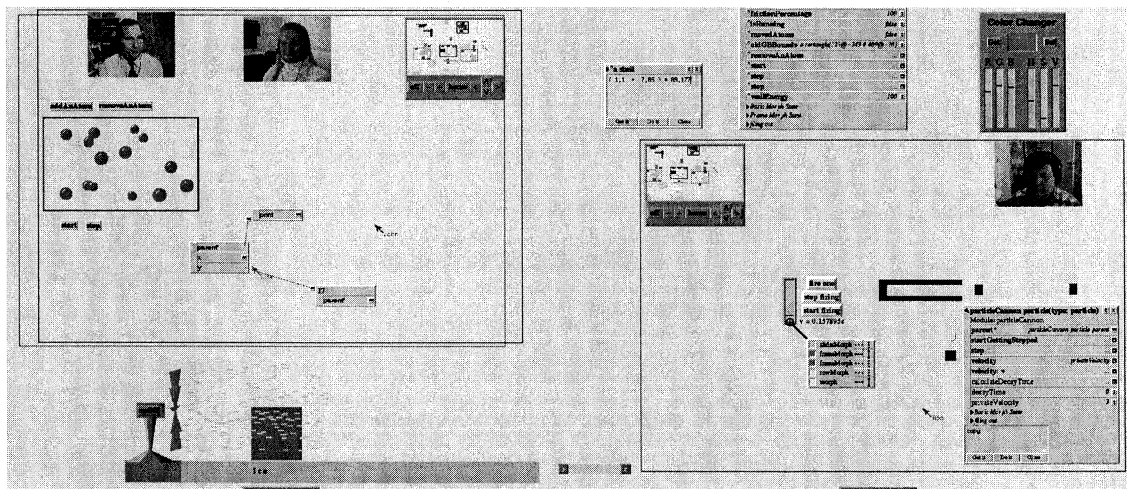


Figure 4.11. Kansas. Image taken from <http://research.sun.com/ics/kansas.html>

Kansas is a collaborative programming environment designed for education (R. B. Smith, Hixon, & Horan, 1998) shown in Figure 4.11. The name refers to the creation of a large, flat space. A very large shared 2D space allows students to work separately and collabo-

ratively, both synchronously and asynchronously, sometimes under the supervision of an instructor. The fully programmable environment is capable of supporting a variety of tasks, including building physics simulations and writing collaborative notes on a shared video presentation.

As with other large 2D spaces, Kansas is not well suited to projection on a single public display. Kansas aims to provide a complete learning environment for students, and the large amount of space allows students to work on their own before merging their work into a finished product. MessyBoard is focused solely on collaboration, and the design assumes that users will continue to do most of their individual work in separate applications. Kansas also differs from MessyBoard in that it provides no interface for the user to browse through the history of activity.

4.7.11 Netomat

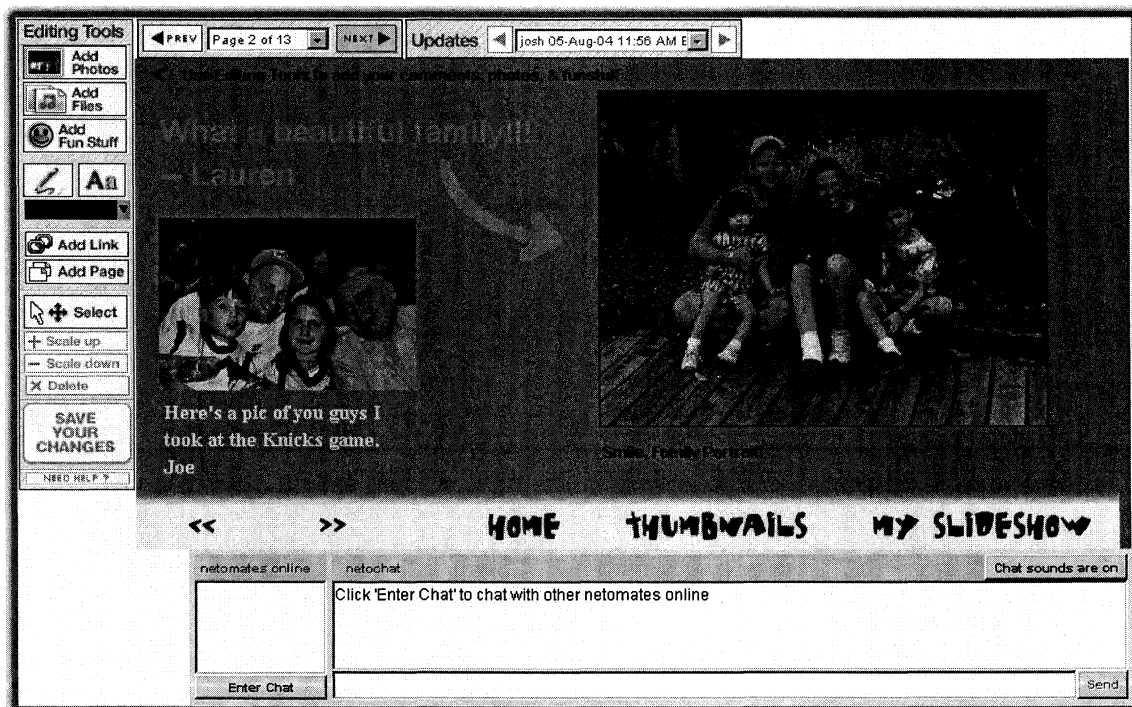


Figure 4.12. Netomat. Image taken from <http://www.netomat.com/>

Netomat provides a shared 2D space with a WYSIWIS paradigm and a Java applet implementation (netomat, 2005) as shown in Figure 4.12. Though the two systems are very

similar, Netomat differs from MessyBoard in two important ways. First, Netomat provides multiple pages and MessyBoard provides only a single finite space. Second, the Netomat space does not show real-time updates as users modify the space, relying on users to explicitly post changes and retrieve the latest content. MessyBoard shows all changes in real time. This is important because users can have confidence that they are seeing the latest content and that their changes will not be overwritten by someone else who is making changes at the same time.

4.8 Physical Corkboards, Whiteboards and Other Public Artifacts

Several researchers have looked at the use of corkboards, whiteboards and other public artifacts in specific contexts such as family coordination (Nässla & Carr, 2003), hospitals (Lasome & Xiao, 2001), or highly focused collaboration in the office (Covi, Olsen, Rocco, Miller, & Allie, 1998). I begin this section by reviewing the literature on corkboards, whiteboards and other public artifacts.

To my knowledge there has been no general survey of the different uses of public corkboards and other kinds of display spaces that people regularly encounter. As a supplement to the literature, I present my own observations of several bulletin boards on the Carnegie Mellon University campus. My observations are neither rigorous nor comprehensive, but they do serve to provide a rough picture of the use of bulletin boards and whiteboards in contexts not covered in the literature.

4.8.1 Trauma Center Operating Room Boards

Lasome and Xiao (Lasome & Xiao, 2001) report on the use of a large magnetic whiteboard to coordinate the use of operating rooms at a trauma center. They observe that "The entire operative process, from initial identification and planning for a surgical procedure through disposition from the OR to the post-anesthesia care unit (PACU), is captured on a large public display board." They describe a frenetic but highly structured use of the board in which there is a designated area for each operating room (OR) and nurses create magnetic "case strips" to represent each patient. When a case is scheduled for a particular OR, the case strip is moved to the corresponding area of the board. Case strips are moved around if plans change due to unforeseen circumstances, which is often the case.

In a separate case study, Bardram reports a similar use of a "wallboard" to schedule and coordinate activity in an unnamed surgical department for urinary surgery (Bardram, 2000). In this case the wallboard contains a written schedule for the day and head nurse is responsible for maintaining the wallboard while surgeons look at it frequently. As in the previous study, unforeseen circumstances lead to changes in the schedule and the hospital staff relies on the wallboard for current information.

In both of these cases, the use of public artifacts is highly structured and the artifacts are critical to the day-to-day operation of the hospitals. The hospital workers have taken an inherently versatile medium and imposed a rigid structure on its use. This is somewhat similar to my observations of MessyBoard use by Group B, in which students used MessyBoard to implement a fair procedure for choosing offices. However, in general the use of MessyBoard that I describe here is far less structured and more varied than the use of whiteboards by hospital staff. Further, groups that I have observed do not, in general, rely on MessyBoard for up-to-the-minute information the way that the hospital staff members rely on their public artifacts.

4.8.2 Personal Office Whiteboards and Corkboards

Two case studies have found that whiteboards in personal offices are used both for personal activities and for supporting conversations. Brinck and Gomez report on the use of whiteboards in personal offices for administrative and technical employees at Bellcore (Brinck & Gomez, 1992). They find that although the whiteboards are located in personal offices, whiteboards are often used to support conversations and meetings. Text was the most common kind of content on these whiteboards, and many participants stated that some of the content on their whiteboards should be left in place for future reference or to support a future conversation. Mynatt reports that whiteboards are used for a variety of tasks including to-do list and reminders, organizing thoughts and illustrating points in a discussion (Mynatt, 1999). Participants in this study reported using their whiteboards intensely for short periods with long lulls in between.

A case study by Malone looks at how ten different workers organize their entire offices (Malone, 1983). Though the study does not focus specifically on whiteboards and bulletin

boards, it does mention that they are used for to-do lists, phone numbers and addresses and for supporting conversations.

Based on these case studies, it seems clear that the use of personal corkboards and whiteboards is quite different from the intended and actual use of MessyBoard. The only reported collaborative use of these personal information artifacts is synchronous face-to-face conversation.

4.8.3 Public Artifacts for Collaboration

Covi et al. provide an overview of the use of dedicated collaborative spaces in 14 Fortune 500 companies. Much of their report is focused on the use of "shared visual displays" including flip charts, whiteboards and tack boards. They report that shared visual displays are often used for three purposes: making work visible, coordination, and motivation. For example, in one group flip charts made software requirements visible to the developers and another group used a large wall display consisting of magnets connected by thin strands of tape to visualize relationships between employees in a company. As an example of coordination, several groups created shared to-do lists. For motivation, groups posted company logos and motivational posters where everyone could see them.

Other case studies of public artifacts fit nicely into Covi's framework. Whittaker and Schwarz report on the use of large wallboards for scheduling interdependent tasks in software development teams (Whittaker & Schwarz, 1995). Bellotti and Rogers report that web publishers used paper on string to visualize the links between pages on their web site (Bellotti & Rogers, 1997). Similarly to the software developers, they also used a large public whiteboard for scheduling and project management.

Of all of the studies reviewed in this chapter, perhaps the use of public artifacts discussed in this section bears the greatest resemblance to the observed and intended use of MessyBoard. Though lacking in detail, Covi et al. suggest that shared visual displays in dedicated project rooms are used asynchronously for a wide variety of collaborative tasks. Several groups used MessyBoard for the kinds of shared lists described by Covi et al. In a

few instances, MessyBoard was used for scheduling or for visualization as described in the studies cited above.

4.8.4 Family Bulletin Boards

Nässla and Carr report on the use of bulletin boards in the home as observed in a photo diary study of three families (Nässla & Carr, 2003). They observe that families use their bulletin boards for a combination of short-term items such as coupons, long-term items such as school schedules and friends' phone numbers, and archival items such as the phone number for the electric company in case of a power outage. Families added and removed several notes during the three-week study period, suggesting that they did actually use the bulletin board for important information.

Though their report does not state it explicitly, it appears that most or all of the content placed on the family bulletin boards was created by someone outside the family. Invitations, school schedules, coupons and other printed materials were received in the mail or given to the families by outsiders. The bulletin boards did not contain lists or collaborative representations of any kind created by the family members themselves. Indeed, one of the main conclusions of the study is that if the physical bulletin board is replaced by an electronic substitute, that substitute must somehow accommodate the paper items that families regularly receive.

4.8.5 Observations of Campus Bulletin Boards

During the Spring semester of 2004 I toured the Carnegie Mellon campus once a week with a digital camera and took photographs of bulletin boards and whiteboards. I located boards based on my own knowledge, by walking through buildings and finding them, and by asking bystanders where I might find them. The boards were used by a wide variety of groups including the entire student body, businesses on campus, academic departments and smaller clubs or research groups.

Near the end of the semester, I attempted to interview one person for each of the boards that I had photographed. I located the person by talking to people in the offices near the boards until I found a person who was either formally responsible for the content of the

board or, if there was no such person, a person who was familiar with the board. I was not able to find such a person for each board in the time available. The open-ended interview questions are listed in Appendix B.

In all, I photographed bulletin boards and whiteboards in 41 separate locations. Corkboards or whiteboards that were close together (on the same wall or in the same small room) are considered to be a single board. I conducted interviews for 23 of these locations.

Based on my observations and my knowledge of the contexts for each board, I have separated the boards into five categories. My analysis does not quantitatively support this categorization. I simply present it as a tool to aid understanding and ground further discussion. For each category, I list some defining characteristics and I present my observation of boards in that category.

3 out of 41 boards did not fit into my classification scheme. They are omitted from the following discussion.

4.8.5.1 Public Boards

A public bulletin board is typically in an area where many people pass by each day and see it. Many of the people who pass by do not have any relationship with one another besides their common affiliation with Carnegie Mellon, and the content on the boards has no particular focus aside from its focus on information relevant to the Carnegie Mellon community. In my sample, 6 bulletin boards are in this category and I conducted interviews for 5 of them. This category includes no whiteboards.

The public board is a cacophony of announcements for events, apartment vacancies and items for sale. They appear messy with a lot of overlap and the content changes frequently. Two examples are shown in Figure 4.13.

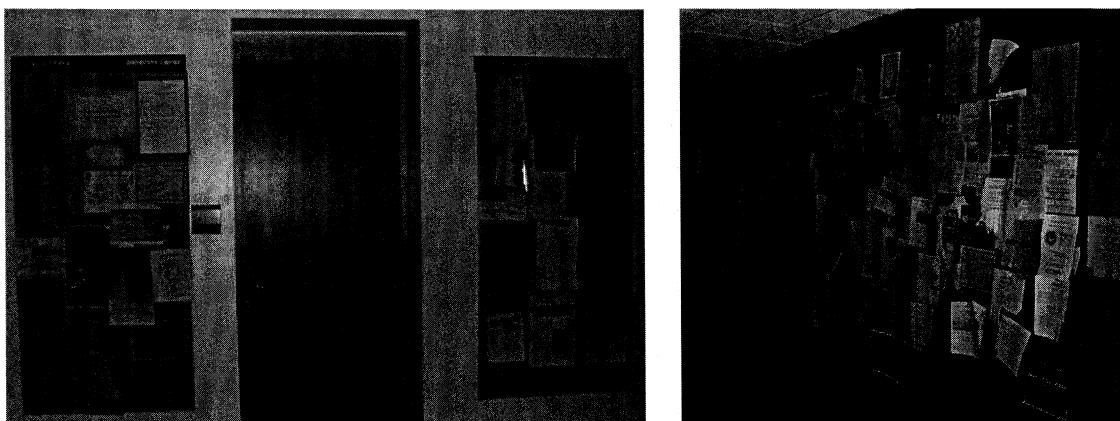


Figure 4.13. Two public boards.

For most of these boards, anyone at all can post information. If rules about access privileges or content exist, they are usually unknown or ignored. Only one of the five interviewees reported strict rules about posting and prompt enforcement of those rules. One interviewee in an academic building said that students were supposed to get approval for posting information but they often did not, and the organization responsible for maintaining the boards would only clean them once every semester. An interviewee in another academic building said that nobody was responsible for maintaining the board and that it was created primarily to keep students from posting flyers on the walls. In contrast, the boards in the University Center (a central building housing student club offices, sports facilities and public auditoriums) are more carefully maintained by a dedicated staff that worked in close proximity to the boards. The staff removes items that are old or "in bad taste". These boards appear less messy, though the content is largely the same as on other public boards in different buildings on campus.

4.8.5.2 Departmental Boards

A departmental board is in an area with limited pedestrian traffic and the people who pass by are likely to be affiliated with a common academic department. Compared to the public boards discussed above, departmental boards appear neater and the content is more relevant to a particular department. 11 of the boards that I observed fall in this category and I conducted interviews for 8 of them. Three examples are shown in Figure 4.14.

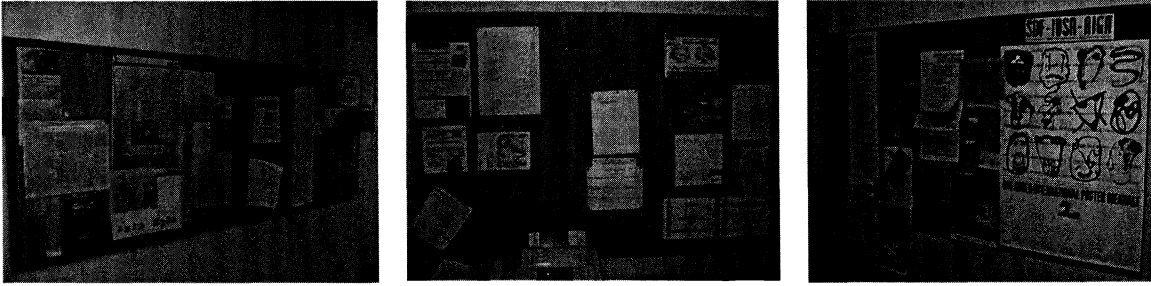


Figure 4.14. Three departmental boards.

A typical departmental board contains news items and printed announcements. Most of this material is relevant to the department. For example, newspaper stories featuring students and professors and posters for conferences in the department's field of study are common.

For some departmental boards, anyone is allowed to post information and in most cases a specific person is responsible for maintaining the board and removing old information. (5 out of 8 interviewees reported that anyone could post and 7 out of 8 reported that someone was responsible for removal.)

4.8.5.3 Public Whiteboards

Public whiteboards are located in hallways and, in one case, in a student lounge. These whiteboards are intended to support face-to-face conversations between researchers or students. My sample includes 5 public whiteboards, all in the same building. I did not conduct interviews for any of them. Three examples are shown in Figure 4.15.

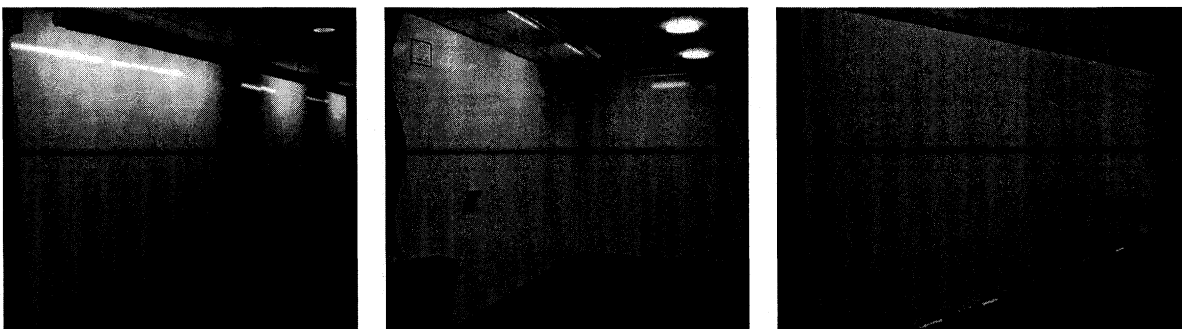


Figure 4.15. Three public whiteboards.

All of the public whiteboards seem to be used for their intended purpose: supporting synchronous collaboration. They are erased every night by a maintenance crew, presumably to keep them from being ruined. For most of my photos the boards were clear, though in a few cases I photographed equations and mathematical diagrams. There is no evidence that these boards are used for asynchronous communication of any sort.

4.8.5.4 Group Boards

A group board resides in a research lab, student organization office or a small business on campus. In all cases, the boards are owned and used by a small group of people who all know one another. My sample includes 9 such boards and I conducted interviews for 7 of them. Many of these boards are combinations of corkboards, whiteboards and wall space. Three examples are shown in Figure 4.16.

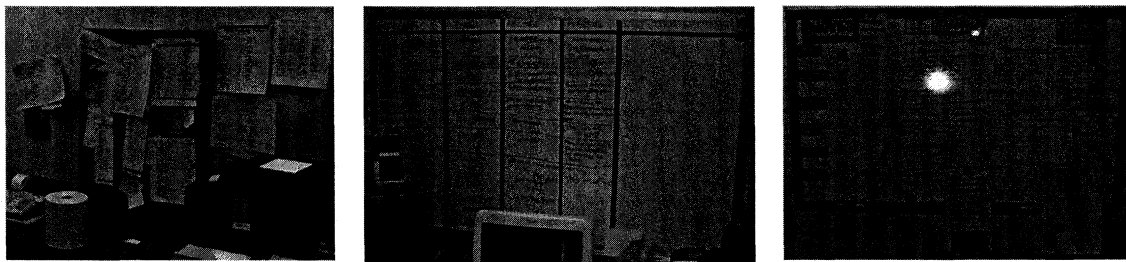


Figure 4.16. Three group boards.

Use of group boards varies widely across groups. Some groups had structured patterns of use that reflected the nature of their work. For example, a store on campus used a bulletin board to keep track of orders and deliveries. In general, one pervasive use of these boards is to make commonly used information such as phone numbers visible and available. Another common use is to pass information between workers on different shifts. Humor is also common, including hand-drawn cartoons and messages on whiteboards and printed material on walls and corkboards.

4.8.5.5 Locked Boards

Locked corkboards are literally locked in glass display cases. Though one could consider these boards to be public, departmental, etc. based on their audiences and location, the

dynamics of these boards are different than other bulletin boards. My sample includes 3 locked boards and I conducted interviews for two of them. All three are shown in Figure 4.17.

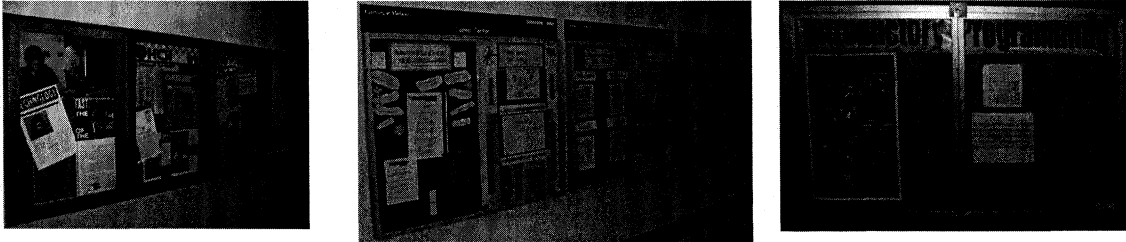


Figure 4.17. Three locked boards.

Locked boards are neatly arranged and the content is seldom updated. One of the boards never changed for an entire semester. A second one changed only once over the course of a month of observation. The third one was only photographed at one point in time. As one can easily guess, locked boards are maintained by a specific person or small group of people and nobody else is allowed to post or remove information.

4.8.5.6 Dormitory boards

I observed three boards located in a single student dormitory. I only observed them at one point in time and I did not conduct interviews for any of them. Two examples are shown in Figure 4.18. These boards are primarily used for printed announcements.

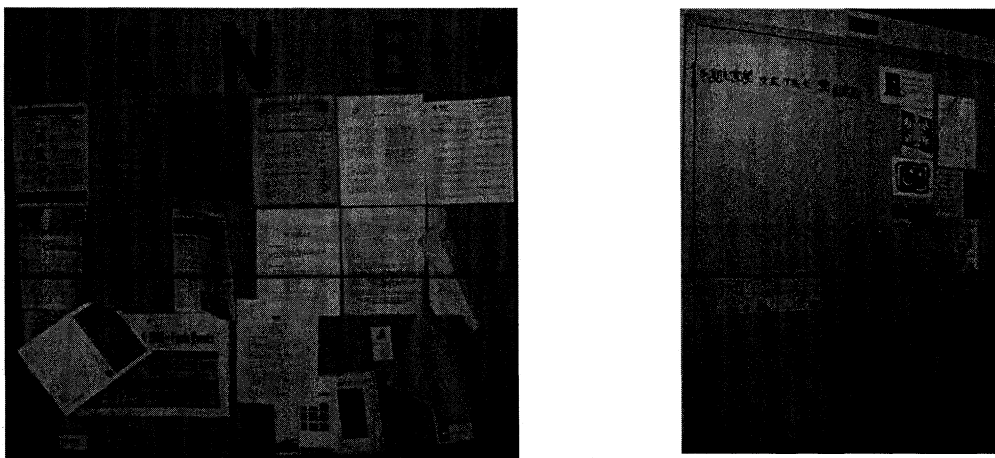


Figure 4.18. Two dormitory boards.

4.8.6 Discussion

Though I cannot draw definitive conclusions about the differences in use between MessyBoard and corkboards and whiteboards based on my observations and the existing literature, a few differences stand out.

MessyBoard use has a conversational aspect: users sometimes reply to an existing note by posting another note. I have not observed this kind of behavior on corkboards, though it sometimes occurs on whiteboards.

One important use of whiteboards seems to be coordination, for example divvying up tasks among members of a group. MessyBoard is also used for coordination by the groups that use it the most. However, there is very little evidence that corkboards are used for coordination.

A related observation is that MessyBoard and physical whiteboards both contain a large amount of user-created content that is designed specifically for that medium while corkboards are used largely for content created by a third party or for other purposes. Pen strokes on a whiteboard are user-created and intended for the medium by definition and users often type content into notes on MessyBoard. Corkboards are typically filled with newspaper articles and printed announcements that came from other organizations. Even content that was created by the group may not be specifically intended for that group's corkboard. For example, the group may have created posters and flyers to send to other groups or to place in mailboxes.

The use of MessyBoard that I have observed combines elements of both whiteboard and corkboard usage. Corkboards are primarily used for news articles and announcements created by third parties. Whiteboards are used for coordination and back-and-forth conversation. MessyBoard is used for both kinds of activities. The reason for these similarities and differences may be the relative costs of these activities in the different media. MessyBoard allows the posting of user-created content and third party digital content with relative ease. Corkboards make it relatively easy to post third party paper content, but creating new content requires a pen and a piece of paper, neither of which are typi-

cally provided near the corkboard. Whiteboards have trays for pens, encouraging the creation of original content, but typically provide no way to post third party content. I observed one magnetic whiteboard that facilitated the posting of third party content. This board was in fact used for a combination of user-created and third party content, as shown in the right-most board in Figure 4.16.

4.9 Grounding

Clark and Brennan state that communication requires mutual knowledge, beliefs and assumptions, which they refer to as *common ground* (Clark & Brennan, 1991). They assert that in order for communication to be successful, the common ground must be updated moment by moment through a process that they call *grounding*. They describe the process of grounding in terms of a face-to-face conversation. People participating in a conversation may repeat each others' utterances to indicate that they may have misheard them, nod and say "yeah" to indicate that they understand what the other person is saying, or give alternate descriptions of referents in order to make sure that both people are referring to the same thing.

MessyBoard can help users to establish common ground in a very straightforward and visible way. Information that is relevant to in a discussion can persist on MessyBoard alongside the discussion. For example, the floor plan in Figure 4.19 helps to establish common ground for interpreting the comment about the door. The screen shots in Figure 4.20 establish common ground for interpreting the surrounding comments. The common ground is explicit and persistent in the medium.

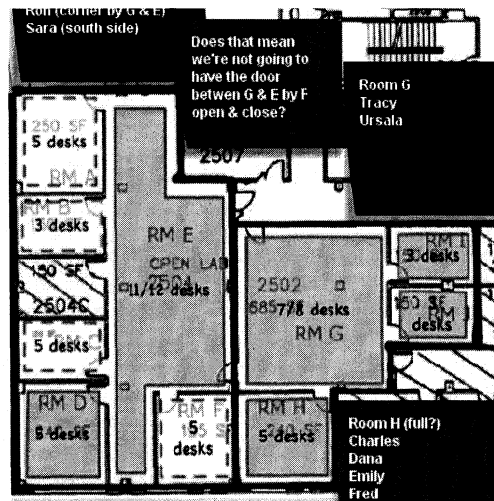


Figure 4.19. The floor plan provides common ground for interpreting the purple note about the door.

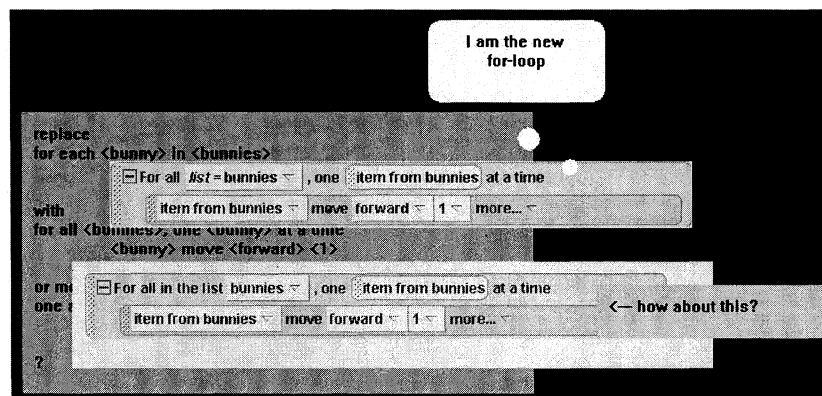


Figure 4.20. The screen shots provide common ground for interpreting the surrounding comments.

Clark and Brennan assert that communicators will use the grounding procedures that incur the smallest collaborative cost, where collaborative cost is the time and effort that all participants spend. For example, it often takes less collaborative effort for a person to utter an incomplete phrase and let another person complete it than it would take for that person to construct and utter a complete sentence. They list a number of different kinds of costs which vary depending on the communication medium. The cost most relevant to MessyBoard is the display cost, or the cost of presenting and referring to an object. MessyBoard differs from other asynchronous communication media in that it allows an

author to easily arrange objects and comments in space such that other participants will understand that the comments refer to the objects.

4.10 Awareness

A growing body of literature describes the concept of *awareness*. Different authors provide multiple definitions and taxonomies of different kinds of awareness (group awareness, task awareness, etc.) However, a consistent central theme in this body of literature is the notion that groups of people coordinate their activities in a seemingly effortless manner. That is, without explicit or intentional communication, people are often *aware* of the relevant activities of their coworkers. Researchers are interested both in studying awareness as it occurs naturally and in improving it through the design of digital tools for collaboration and communication.

I have claimed that MessyBoard allows people to see important information about their coworkers' activities at a glance. I have focused on keeping the receiving cost low by allowing a finite amount of space and by displaying the medium on a large public display and as a screen saver. These are ideal qualities for a medium that is designed to foster awareness.

However, it is also clear that MessyBoard requires explicit action on the part of its users. Users must post information on MessyBoard, and they must make decisions about what information will occupy the limited space. This runs contrary to the idea that coworkers can effortlessly remain aware of each others' activities.

In this section, I provide a brief overview of the awareness literature and I compare MessyBoard to systems that have been specifically designed to foster awareness. Though the concept of awareness is controversial and ill-defined in the literature, the pursuit of awareness has led to the development of two distinct classes of experimental systems: media spaces and event notification systems. Though some of these systems share characteristics with MessyBoard, they are quite different in many respects. The differences in the systems arise from two fundamental differences in design philosophy:

- MessyBoard assumes the existence of a stable group and awareness systems assume that groups and relationships between people are fluid.
- MessyBoard assumes that people will act explicitly to communicate important information and awareness systems assume that the best way to foster awareness is to make individual activity visible to others

More recently, the term "awareness" has been invoked to explain the design of a wide variety of systems including WYSIWIS 2D spaces (Roseman & Greenberg, 1996), large public displays (Huang & Mynatt, 2003) and collaborative virtual environments (Benford, Bowers, Fahl, & Greenhalgh, 1994). These more recent developments do not stand as a coherent body of work and the authors' use of the term "awareness" has become highly ambiguous (Schmidt, 2002). I do not address these more recent awareness systems in this section, though I address several of them in Section 4.1 since they share key features and characteristics with MessyBoard.

4.10.1 The Awareness Literature

The term "awareness" is used in many different contexts. The introduction to the CSCW special issue on awareness provides an overview (Schmidt, 2002). This section is concerned only with awareness as it relates to collaboration and coordination between people. Dourish and Bellotti define awareness in the context of collaborative writing, but their definition captures the general meaning of the term as it is used throughout the literature:

...awareness is an *understanding of the activities of others*, which provides a *context for your own activity*. This context is used to ensure that individual contributions are relevant to the group's activity as a whole, and to evaluate individual actions with respect to group goals and progress. The information, then, allows groups to manage the process of collaborative working. (Dourish & Bellotti, 1992) (Emphasis in original)

The term "awareness" is used to describe knowledge of relationships between people, task dependencies and the content of documents among other things. Gutwin et al. set forth a framework which describes four types of awareness in the context of collaborative learning:

Social awareness is the awareness that students have about the social connections within the group. *Task awareness* is the awareness of how the task will be completed. *Concept awareness* is the awareness of how a particular activity or piece of knowledge fits into the student's existing knowledge. Finally *workspace awareness* is the up-to-the-minute knowledge about other students' interactions with the shared workspace, such as where other students are working, what they are doing, and what they have already done in the workspace. (Gutwin, Stark, & Greenberg, 1995) (Emphasis in original)

For present purposes, the most important distinction is that between social awareness and task awareness. Concept awareness is specific to collaborative learning which is outside the scope of this dissertation. Workspace awareness assumes that collaborators perform all of their activities in a "shared workspace" that can share information such as who is looking at what information in real time. This may be true in the future, but the present work assumes the current reality: users perform most of their work using special-purpose applications that cannot automatically share information about the users' activities with coworkers. Though I discuss systems that are designed to promote workspace awareness, they are relevant only in so far as they promote task or social awareness as well.

Another crucial distinction in the awareness literature is that of short-term versus long-term collaboration. This is not entirely the same as the difference between synchronous and asynchronous collaboration, since some systems are designed to support long-term collaborations that are composed of many synchronous and asynchronous activities. In practice, many experimental systems focus on supporting only synchronous collaboration by providing real-time feedback about the actions of participants, for example the shared drawing applications mentioned in Section 4.2 and real-time shared text editors (Dourish & Bellotti, 1992). These systems are designed for short-term collaboration, since they do not do anything in particular to aid collaboration beyond a single synchronous meeting. Since MessyBoard is designed to be used over a long time span by a stable group, this discussion focuses on awareness systems that explicitly aim to provide awareness for long-term collaboration.

A final distinction in the awareness literature is that between active and passive mechanisms for communicating awareness information. Dourish and Bellotti explain this distinction:

A primary distinction between these mechanisms is whether the information is *explicitly generated*, directed and separate from the shared work object; or *passively collected and distributed*, and presented in the same shared work space as the object of collaboration. (Dourish & Bellotti, 1992) (Emphasis in original)

Though Schmidt attacks the validity of this dichotomy, he rightly points out that the bulk of the literature embraces it and focuses on passive mechanisms (Schmidt, 2002). Many of the systems that are designed to foster awareness attempt to do so without additional effort on the part of the user. In contrast, MessyBoard is an active mechanism by design. The MessyBoard design philosophy is to acknowledge the costs of using active mechanisms and minimize them.

In summary, Gutwin divides awareness into four categories of which two are relevant: task awareness and social awareness. Systems that support only short-term collaboration are not relevant. Thus, I compare MessyBoard to systems for task and social awareness in long-term collaboration. The use of passive mechanisms for gathering and distributing information is likely to be a major source of differences between these other systems and MessyBoard.

4.10.2 Media Spaces

Media spaces typically use audio and video links in order to foster collaboration. Contrary to commercial video conferencing systems, media spaces strive to make this kind of communication a natural part of workers' daily routines, for example by placing cameras and video displays in workers' personal offices or by placing always-on cameras and displays in the common areas of two remote sites. Bly et al. have described one of the most well-known media space experiments and provide a brief survey of similar efforts by others. (Bly, Harrison, & Irwin, 1993)

Media spaces are designed to improve collaboration across space and time based on the assumption that work is inherently social. That is, in order to complete tasks, people need to be aware of what their colleagues are doing and have opportunities for chance encounters. Media spaces are designed primarily for social awareness with task awareness emerging as a side effect.

Though the most salient use of media spaces is for synchronous conversation, media spaces are clearly designed to support long term collaboration. They do this by making it easy for workers to casually use the audio and video links to locate particular coworkers or just to see who is in the common area or who is still working late in the evening.

The clearest similarity between MessyBoard and media spaces is the emphasis on reducing the sending and receiving costs of communication. Though the media space literature does not tend to say this explicitly, it is clear that the ability to use a media space informally and casually is critical to its success. Thus, MessyBoard and media spaces converge in displaying the medium in public spaces and allowing access from personal offices.

The clearest difference between MessyBoard and media spaces is in the nature of the communication: a persistent bulletin board versus synchronous conversation over an audio/video link. Media spaces are inspired by the idea that informal conversations and chance encounters are critical to collaboration (Kraut et al., 1990), while MessyBoard simply allows people to easily share information with their group. The two approaches are both necessary and complementary.

4.10.3 Event Notification Services

Event notification services focus on the detection of events that people might not ordinarily perceive and delivering notifications of the events to people who are interested. For example, sensors may generate events when relevant files are modified or when motion is detected in a colleague's office (Fitzpatrick, Kaplan, Mansfield, David, & Segall, 2002; Prinz, 1999). It is assumed that these systems will be deployed to an entire organization, so the researchers in this field devote a great deal of effort to developing models for event

distribution based on user-generated profiles (Fitzpatrick et al., 2002; Prinz, 1999), semantic networks (Fuchs, Pankoke-Babatz, & Prinz, 1995) or other mechanisms.

Events are delivered to users in a variety of ways. The GroupDesk system provides a full-fledged work environment and presents event notifications associated with documents when a user views the document. Thus, the GroupDesk system provides information in context, but not in real time. By contrast, the Elvin system provides a "Tickertape" client that stays on the user's screen and displays notifications in real time.

Event notification services are designed to support both task and social awareness for long-term collaborations. They seek to do so by taking events and activities that are already present in the environment and making them visible by notifying interested parties. Thus, event notification services are primarily a passive awareness mechanism. (There are exceptions to this generalization, such as the Tickertape client provided by the Elvin system. (Fitzpatrick et al., 2002))

MessyBoard shares very little in common with event notification services. This is due in part to the fact that MessyBoard is an active medium and event notification services are largely passive. More fundamentally, the differences arise from the assumption that an event notification service will be used in an environment where patterns of relationships between people are complex and fluid. By contrast, MessyBoard assumes that people work in small stable groups. Thus, event notification services collect a plethora of events and employ computation to make sure that each individual only gets relevant notifications. MessyBoard relies on the users to act as their own "sensors" and "filters," assuming that the group has a shared understanding about what information is relevant and what is not.

4.10.4 MessyBoard as an Awareness System

Though MessyBoard differs from existing awareness systems, the design and use of MessyBoard is highly compatible with the abstract notion of awareness. MessyBoard has been used to make group members aware of everything from tasks to be accomplished to

upcoming birthdays. This happens via a simple and active mechanism: people post information on MessyBoard if they feel that others should see it.

MessyBoard may also improve awareness in a more subtle way: by encouraging broadcast communication to an entire group instead of bilateral communication between individuals. The clearest example of this may be Group D, described in Section 5.5.2.2.1, who used MessyBoard to share files on a regular basis. A file may be intended for a specific coworker and it may be the case that nobody else downloads it. However, if other group members look at MessyBoard and see that a file has been posted this gives them information about what their coworkers are doing. In this sense, MessyBoard can be viewed as a passive awareness system: person A uses MessyBoard to transfer a file to person B, and person C sees that this has occurred because she uses MessyBoard for other purposes.

MessyBoard is different from media spaces and event notification systems in that it assumes the existence of a stable group with common interests. MessyBoard's finite space and lack of access control mechanisms such as accounts and permissions make it unlikely that a single MessyBoard space would work for a large organization. Media spaces and event notification systems can be deployed to entire organizations and they have the potential to connect people within and between smaller groups. Users can make sure that they only get relevant information from media spaces and notification systems through their habits or by setting preferences in the software. However, the flip side of this is that users do not necessarily understand who sees what information. By contrast, MessyBoard users may have a clearer understanding of who will see their space due to the finite space and the WYSIWIS paradigm.

4.11 Groupware Adoption

Researchers and practitioners have found that it can be extraordinarily difficult to get a group of people to start using a groupware application, regardless of the theoretical utility that the application provides. The literature on groupware adoption discusses the causes of this problem and potential remedies. Authors focus on the initial period of use and the conditions that lead to adoption or rejection of a groupware application, often assuming

implicitly that the application will be useful to the group as a whole if they adopt it. This body of literature covers a diverse set of groupware applications, including e-mail, chat and group calendar systems.

The literature mentions three kinds of factors that often influence the adoption of groupware applications:

- The distribution of work and benefits among users of the system
- Whether or not the application attracts enough users to provide any value at all (critical mass)
- How well the application fits with existing social norms and political power structures

In this section, I briefly summarize each of these issues and discuss their bearing on the adoption of MessyBoard.

4.11.1 Work/Benefit disparity

One factor that often inhibits adoption of groupware is a disparity in the distribution of work and benefits among users. Even if a group as a whole would benefit from using a groupware application, the application faces an uphill battle for adoption if some individual users need to do work to support the system without realizing any personal benefit. Grudin cites two examples: electronic calendar systems and project management applications. In both cases, the tools are most useful to managers, but they require subordinates to do extra work in order to keep the information in the systems up to date (Grudin, 1994).

Since MessyBoard is a versatile and unstructured system, the distribution of work and benefits depends on the activities that it is used for. In most of the groups that I observed, there were no explicit rules requiring people to post information on MessyBoard. If a group does not set any explicit rules requiring people to use MessyBoard then any individual user can post information on MessyBoard when it benefits her for others to see it and there is no disparity.

The most beneficial uses of MessyBoard may impose a work/benefit disparity. Perhaps the most beneficial use of MessyBoard is to create a shared representation in order to solve a complicated problem. For example, in Section 5.5.1.1.3 I discussed how group B used MessyBoard to keep track of where people were moving in their new office space. In these situations, one user must do some initial "setup" work to start the process, such as creating a calendar grid or posting a floor plan and creating occupancy lists for each room. However, even in this case, the user who does the initial work still benefits from having the shared representation.

4.11.2 Critical mass

Critical Mass theory highlights a problem of particular importance in groupware adoption: even if work and benefits are distributed evenly among users, it may be the case that no individual benefits from the application until many people start doing the work (Markus & Connolly, 1990). This means that if all users act in their own self interest, initially nobody has any incentive to start using the system. This kind of situation is closely related to the "prisoner's dilemma" in game theory (Gallo & McClintock, 1965).

MessyBoard is a persistent asynchronous medium, so it is not sensitive to problems of critical mass in the same way as synchronous applications such as chat. For synchronous applications, multiple users must be using the application at the same time in order to benefit from it. MessyBoard users can benefit from the first time they view a MessyBoard space as long as someone else has posted some information at any time in the past. However, if users are slow to respond to requests on MessyBoard, users may still perceive that they are doing work without receiving any benefit by posting information on MessyBoard or even just by taking the time to look at it.

MessyBoard is a broadcast-only medium for a single group, and this characteristic may both solve and exacerbate the critical mass problem. On the one hand, any information on MessyBoard is immediately of potential benefit to all users of the same MessyBoard space. By contrast, in a medium for bilateral communication, only two users would benefit from any instance of use. On the other hand, bilateral communication tools such as chat and e-mail can achieve critical mass across an entire organization by connecting

people who are in different groups. MessyBoard cannot leverage between-group communications to achieve critical mass. The observations of Herbsleb et al. on the deployment of the RVM chat system suggest that even for a tool that facilitates bilateral communication, adoption is more likely to result by achieving critical mass in small groups than by connecting individuals throughout the organization. (Herbsleb et al., 2002)

Displaying MessyBoard on a large public display or as a screen saver may work to counteract the critical mass issue. If the threshold for viewing the MessyBoard space is low enough, users can begin to benefit from the software without running the client on their own workstations.

Another factor that may counteract the critical mass issue is the notion that broadcasting information to a wide audience can be perceived as a benefit in and of itself. For example, if a group of MessyBoard users contains a few extroverts who crave attention and their MessyBoard space is displayed in a prominent location, those extroverts may use it extensively to broadcast information to their colleagues. The critical mass perspective might be that the few extroverts are doing all of the work and everyone else receives the benefits, but this ignores the fact that the extroverts *enjoy* posting the information. This leads to the prediction that MessyBoard is more likely to be adopted by large groups, since a large group is more likely to contain a few extroverts and they get a larger benefit by having a larger audience. Two of the large groups described in Chapter 5 are known to have such members and log data suggests that a few members of these groups were responsible for the bulk of the material posted on the MessyBoard space.

4.11.3 Social and Political Factors

Groupware often runs into trouble when the prescribed use of the software violates social taboos or political power structures. Grudin cites two examples of this (Grudin, 1994). The first is an automatic meeting scheduling system that ignores managers' unspoken preferences for guarding their free time, leading managers to reject the system. The second is a work management system that instructs individual programmers to send progress reports directly to the chief executive officer when they report a problem. The program-

mers stopped reporting problems and sabotaged the system to so that it would not report their behavior to the administrator.

MessyBoard stands apart from these examples in that there are no explicit uses built into the system. Groups choose the tasks and activities for which they will use MessyBoard and they design protocols for using either explicitly or through their day to day use of the system.

The fact that MessyBoard allows only broadcast communication may cause violations of group norms. A user may, quite reasonably, choose not to use MessyBoard because it is socially appropriate to communicate certain information to another group member without making it available to the entire group.

Chapter 5

Observations of MessyBoard Use

5.1 Research Questions

MessyBoard may be most practical as a tool for groups of workers if it is deployed across an entire organization. In such a setting, a dedicated IT staff can maintain the MessyBoard server in order to ensure low-overhead use for other members of the organization. In addition, MessyBoard could leverage existing systems for creating accounts and setting permissions (Windows domains or UNIX accounts, for example) in order to control access to different MessyBoard spaces without imposing the extra startup cost of account creation when a new MessyBoard space is created.

Organizations pay a price in time, money and human resources in order to deploy and maintain enterprise applications. In order to decide whether or not to make this investment, decision-makers in organizations must have some idea how the application will affect the people who use it. Corporations are willing to deploy communication applications such as e-mail, Lotus Notes (Orlikowski, 1992) and, more recently, instant messaging (Lawton, 2003) as enterprise applications.

MessyBoard is, by design, a versatile and multi-purpose application. It is difficult to predict how MessyBoard will affect the dynamics and processes of any particular group based on the features and affordances of the medium. From the perspective of an executive in a large organization deciding whether or not to deploy MessyBoard, the most important question is this: Do groups find MessyBoard useful in general? More specifically, a decision-maker should know what attributes of groups are the best predictors of MessyBoard use so that she can make an informed decision based on knowledge of her organization. Is MessyBoard most useful for a certain kind of work, for groups of a certain size or for groups with a certain power structure? Answers to these questions can in-

form choices about whether to deploy MessyBoard and assist executives in targeting the deployment to the parts of the organization that are most likely to benefit.

I have discussed showing MessyBoard on a large public display (such as a projector or plasma screen) to lower the receiving cost of communication. These displays must be purchased and maintained by organizations and this is an additional cost above and beyond the cost of the MessyBoard server. Thus, it would also be useful to know the relative impact of a large public display given that a group can use MessyBoard.

I have also discussed the MessyBoard screen saver as a low-cost alternative to large public displays for reducing the receiving cost. Though the screen saver utilizes existing display hardware, it is far from clear whether it provides immense benefits for free. If users are not forced to install it they may not choose to install it on their own, since some users find screen savers annoying and others are loyal to their existing screen saver. If it is installed automatically they may choose to remove it.

5.2 Methodology Overview

I began with small number of users and then I gradually widened the MessyBoard user base through recruiting and advertising. I observed the first few groups very carefully and as the number of groups increased I relied more heavily on automated logging of MessyBoard activity and online surveys. Figure 5.1 summarizes the different methodologies that I used.

5.3 Prototype Deployment

I created and deployed a prototype MessyBoard system in order to explore the usefulness of the medium and observe similarities and/or differences between different groups of users. I described the early system and my observations of users' behavior in a paper in the proceedings of the 2002 Designing Interactive Systems conference (Fass et al., 2002).

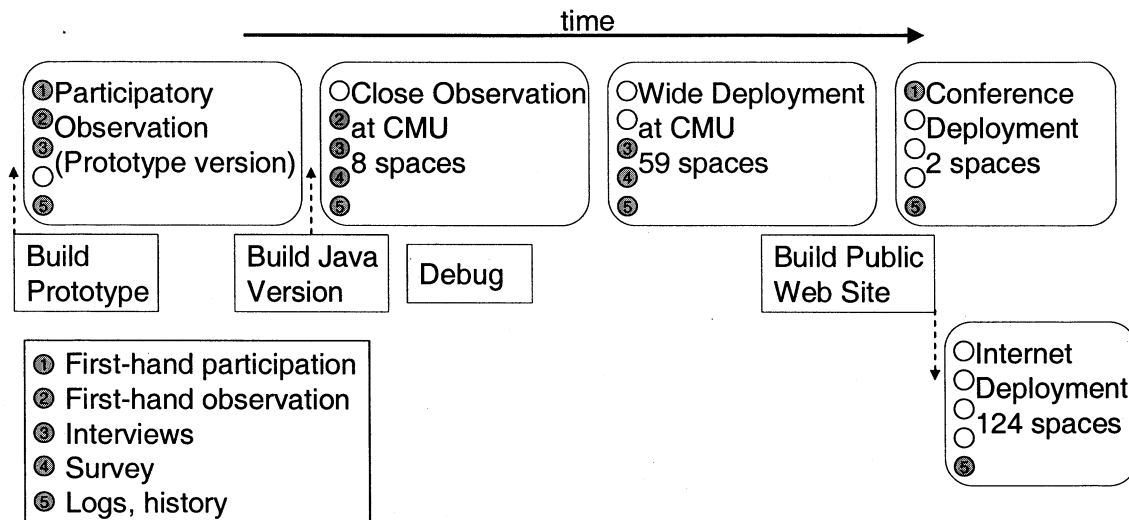


Figure 5.1. Summary of MessyBoard deployments and methodologies for observation.

5.3.1 Methods

The MessyBoard prototype system was an adaptation of the MessyDesk system (Fass et al., 2002). The most marked differences between the current and prototype systems have to do with startup cost: The prototype system required installation (unpacking a ZIP file) and it did not run in a web browser. Although I did implement a screen saver for this prototype, I did not release the screen saver until after I made the observations described below and few people used it.

The prototype did not have a history mechanism, and there was no way to view past states or recover old material. The prototype did not allow users to share files or draw freeform pen strokes and there was initially no identity feature to automatically indicate who authored a note using colors or fonts. (Again, this feature was added after I made the observations described below.) Aside from that, the prototype had most of the essential features of the current Java version: a finite 2D WYSIWIS space with freeform layout via direct manipulation and the ability to import content using drag-and-drop or copy-and-paste.

I set up two MessyBoard servers: one for the Info Cockpits project (Tan et al., 2001) (an effort to improve human memory by leveraging context effects) and one for the Alice

project (Conway, Audia, Burnette, Cosgrove, & Christiansen, 2000) (a drag-and-drop programming environment for creating interactive 3D worlds). Info Cockpits was a collaborative project involving Computer Scientists at Carnegie Mellon University and psychologists at the University of Virginia. The Alice group was an interdisciplinary team of programmers and artists at Carnegie Mellon. Both groups included members of the Stage3 research group at Carnegie Mellon. I was closely affiliated with both groups and I knew the members personally.

I projected the contents of both boards on different walls in our lab at Carnegie Mellon, and the researchers at The University of Virginia projected the Info Cockpits board in their lab. Snapshots of both boards are shown in Figure 5.2 and Figure 5.3.

To observe the system in use, I recorded graphical snapshots whenever a change was made. I observed the projection screens to see what people were doing, checked the snapshots when I missed something, and made notes. I met once with the users of each space to review the snapshots and discuss how they were using it and how I could make it better. The Alice MessyBoard space was used for approximately seven months and the Info Cockpits MessyBoard space was used for approximately one and a half years.

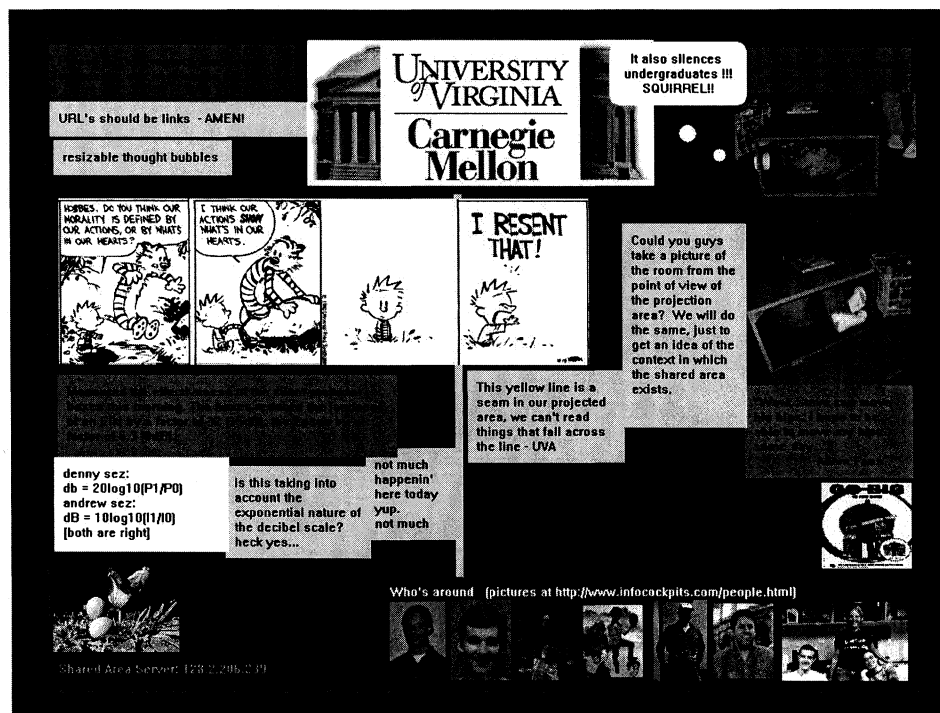


Figure 5.2. The Info Cockpits MessyBoard space.

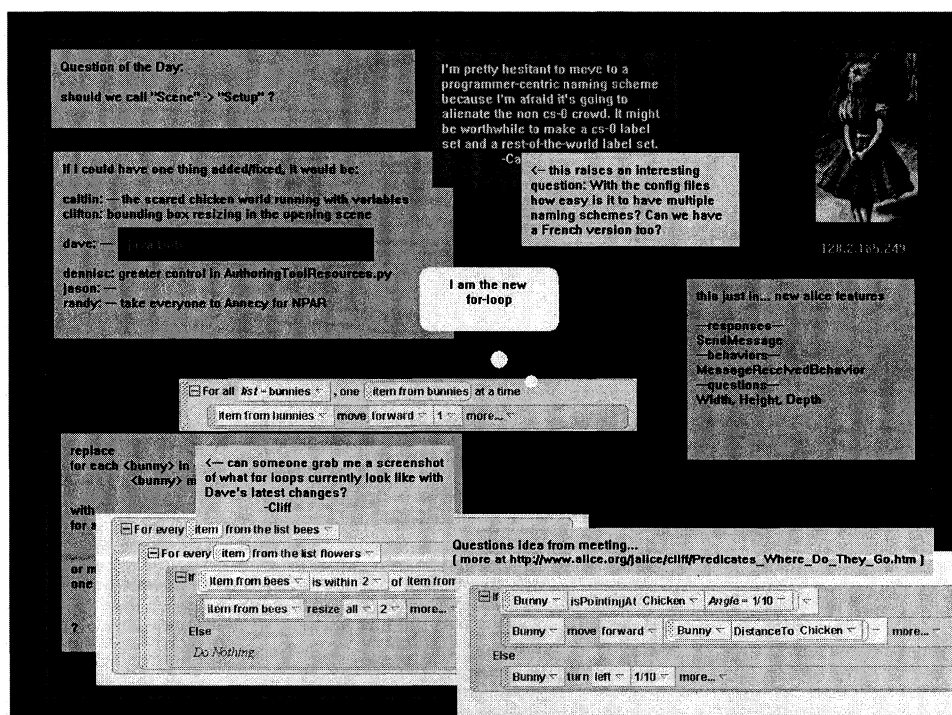


Figure 5.3. The Alice MessyBoard space.

5.3.2 Observations

The following list is a brief summary of my observations, and each is discussed in depth below.

- Different research teams use MessyBoard differently, and their usage reflects both the nature of the projects and the nature of their collaboration.
- Projecting the contents of the board on the wall affects the way people use it.
- People want to use the board to know who else is available, but they quickly learn to ignore this information if it is unreliable.
- A shared bulletin board is better than e-mail for scheduling meetings.
- Users sometimes have trouble figuring out who said what.
- Conversations on MessyBoard sometimes seem less polite than face-to-face conversations.
- A simple game or contest gets people more excited about MessyBoard and causes them to pay more attention.
- Users sometimes feel alienated when the board contains references that they don't understand.

5.3.2.1 Differences between research groups

I set up the Info Cockpits space first and encouraged researchers at Carnegie Mellon and The University of Virginia to use it. It immediately became a forum for discussing bugs in and suggested features for MessyBoard. People also used it to share humorous pictures and comments and play simple games. MessyBoard was not used very much for research collaboration. I suspect that MessyBoard was used this way because the users know each other well and they frequently contact each other by phone and e-mail, but nobody at Carnegie Mellon was collaborating with anyone at The University of Virginia on a specific project with concrete short-term goals.

I later set up the Alice space, which was used quite differently. The Alice team is mostly made up of researchers at Carnegie Mellon, with one member working remotely from Boston. The content of the Alice MessyBoard consists almost entirely of design discussions with snippets of code and snapshots of the interface, with an occasional inside joke or discussion of an unrelated topic. I believe that the Alice team uses MessyBoard this way because it is so well suited to their project: they need to collaborate with a remote member, they have long discussions about their interface design that span several meetings, and their discussions benefit from the ability to share 2D graphical information. MessyBoard is good for displaying a small piece of a visual interface and allowing many people to comment on it.

5.3.2.2 Projection

I observed that projecting MessyBoard in the lab makes a huge difference. Everyone looks at it as they walk by, and everyone in the lab can see that others are paying attention to it. This prompts people to participate in decorating and sharing pictures and text, since they know that others will notice it. In one instance I observed members of the Alice team gathered around the projection of the Alice MessyBoard discussing the snapshots of the interface that they had posted. Greenberg and Rounding observed similar behavior when they ran Notification Collage on a large public display (Greenberg & Rounding, 2001). Figure 5.4 shows a panoramic photo of part of our lab at Carnegie Mellon University with two MessyBoard spaces projected on the walls.

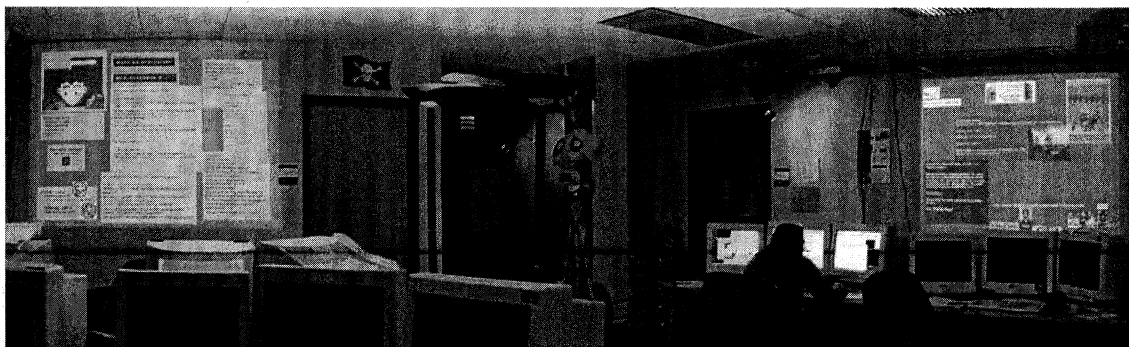


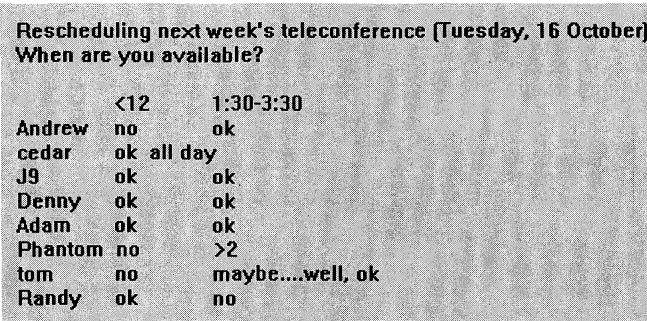
Figure 5.4. The Stage3 lab with the Alice (left) and Info Cockpits (right) MessyBoard spaces projected on the walls.

5.3.2.3 Awareness: Who's Around

One of the first things users did on the Info Cockpits board was to put pictures of themselves in the bottom right corner to indicate who was in the lab, as shown in Figure 5.2. This information is potentially very useful since the Info Cockpits team is spread over two universities. However, the information is unreliable because people often forget to remove their pictures when they go home, so users learned to ignore it. MessyBoard may benefit from a mechanism that automatically indicates who is paying attention to the medium such as the one used in AOL Instant Messenger (AOL, 2005), but this feature remains as future work.

5.3.2.4 MessyBoard is Good for Scheduling a Meeting

One of the best early uses for MessyBoard was scheduling a meeting. Email is very annoying for this purpose: each user ends up with a pile of responses in her inbox and it is difficult to figure out when everyone is available. With MessyBoard, each person put their availability into a single shared note so anyone could immediately see when people were available, as shown in Figure 5.5. Scheduling in MessyBoard became a routine practice and we refined our technique to use multiple notes and a calendar image pasted from a spreadsheet as shown in Figure 5.6.



Rescheduling next week's teleconference (Tuesday, 16 October)		
When are you available?		
	<12	1:30-3:30
Andrew	no	ok
cedar	ok	all day
J9	ok	ok
Denny	ok	ok
Adam	ok	ok
Phantom	no	>2
tom	no	maybe....well, ok
Randy	ok	no

Figure 5.5. Users schedule a teleconference on the Info Cockpits MessyBoard using a single note.

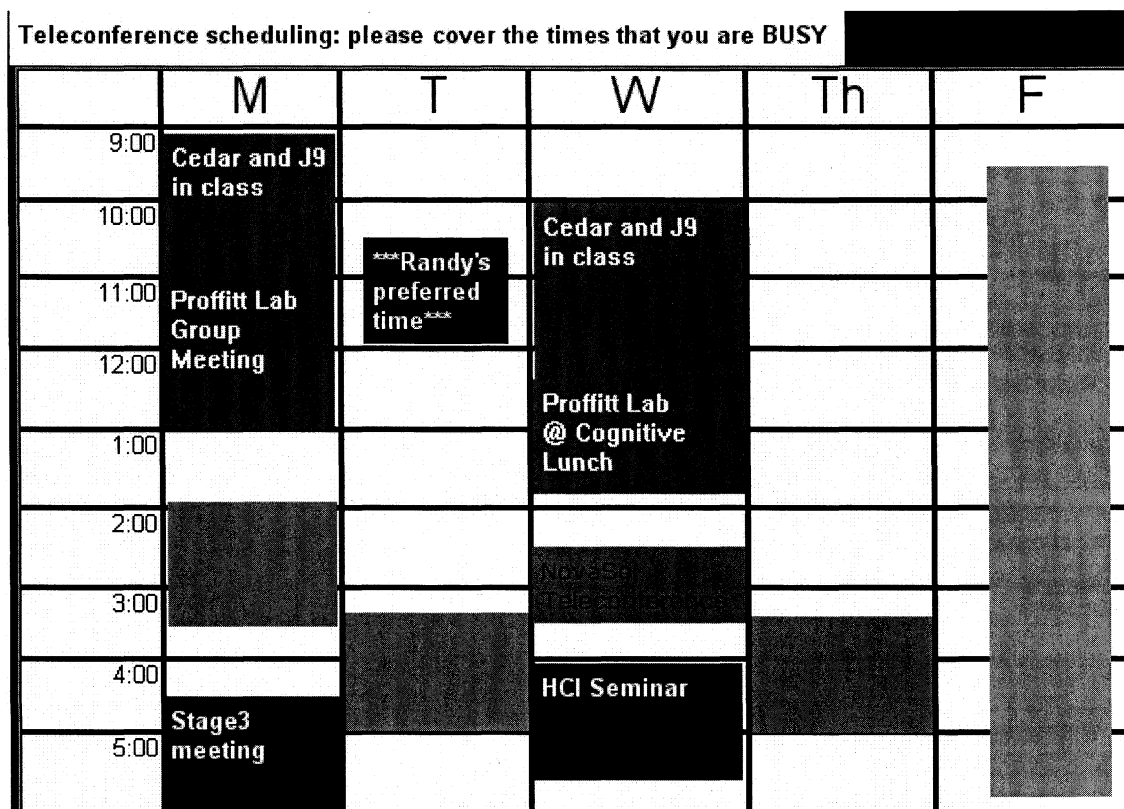


Figure 5.6. Users schedule a teleconference on the Info Cockpits MessyBoard by placing notes over an image pasted from a spreadsheet.

5.3.2.5 Authorship and Politeness

A technical discussion on sound measurement started when a researcher at The University of Virginia replied to an initial posting by a researcher at Carnegie Mellon. The discussion continued over several hours as shown in Figure 5.7. Though it was encouraging to see this kind of interaction, it revealed two shortcomings in MessyBoard. First, the authors had to explicitly write their names or manually choose a specific color for their notes so others could tell who had said what. Second, some users thought that the conversation became slightly rude. This was later confirmed at a face-to-face meeting with all of those involved, and it is known that electronic "conversations" often seem less polite (Brennan & Ohaeri, 1999) and lead people to evaluate each other less favorably (Kiesler, Zubrow, & Moses, 1985) than real-life conversations.

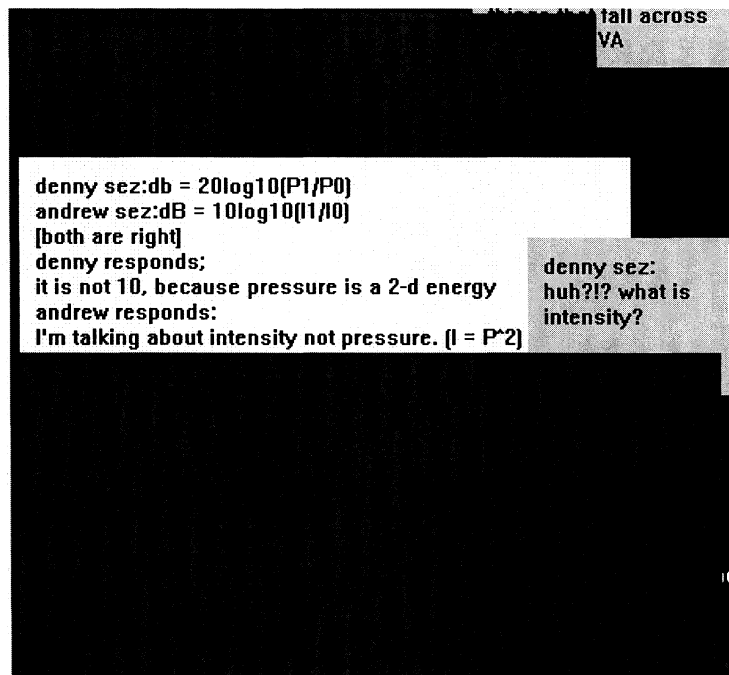


Figure 5.7. A technical conversation on the Info Cockpits MessyBoard space.

5.3.2.6 Simple Games Attract Attention

One of the first uses of MessyBoard was for simple games. Users played tic-tac-toe for a day or two with different people making moves every few hours. Someone then started a game of Pong, where Carnegie Mellon controlled one paddle, The University of Virginia controlled the other, and anyone could move the ball. This is shown in Figure 5.8. These games used only existing elements: the tic-tac-toe board was made up of 9 small notes, the pong paddles were long thin notes, and the ball was a picture of a circle. These games had no hard and fast rules - anyone could move any of the game elements, and there was a good deal of cheating. Nonetheless, users enjoyed these activities, and the fact that they did them at all is interesting given the effort required.

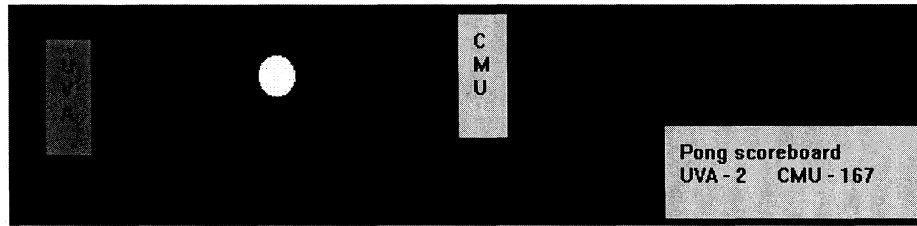


Figure 5.8. Users play a game of Pong in the Info Cockpits MessyBoard space.

5.3.2.7 Instant “War Room”

One of the most valuable uses of MessyBoard occurred during an anomalous period of high activity and communication. The Info Cockpits group was assigned to create a demonstration video on short notice. Since members at both The University of Virginia and Carnegie Mellon were working together so closely, there was a need for tight communication. The two labs held daily teleconferences and the MessyBoard space was co-opted as a dedicated workspace for this project. Users posted preliminary renderings, questions and comments on a daily basis until the video was complete. Figure 5.9 shows the MessyBoard space during this time.

This period is interesting because it represents a deviation from normal patterns of use for this MessyBoard space. Later observations support the notion the usefulness of a MessyBoard space cannot necessarily be judged based on the “normal” patterns of usage and interaction. MessyBoard may be worthwhile for the times when a group suddenly and unexpectedly needs a low-cost means of communication and they will use whatever is at hand.

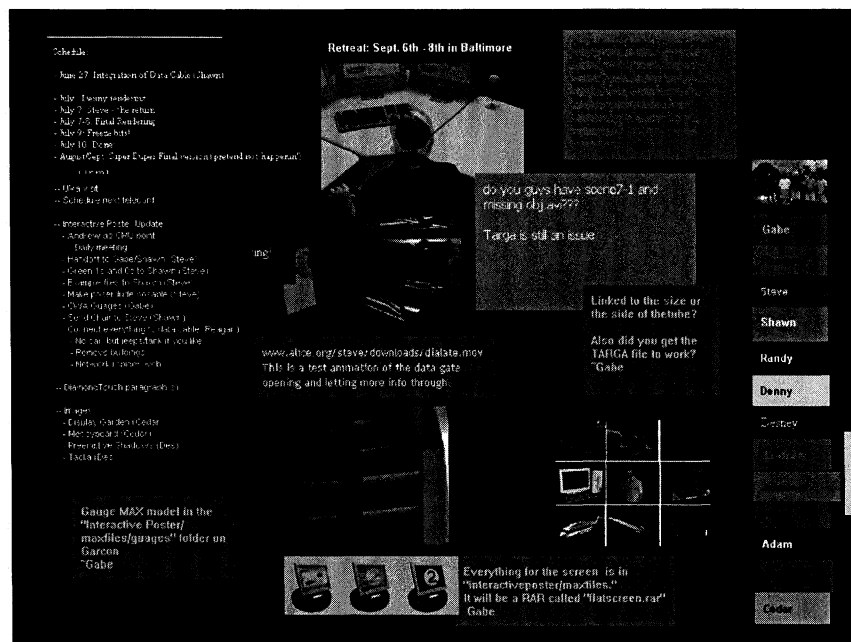


Figure 5.9. The Info Cockpits MessyBoard spaces becomes a dedicated project space for creating a video demonstration on short notice.

5.3.3 Lessons learned

The most important lesson learned from the MessyBoard prototype observations is that people did in fact find the medium useful and enjoyable and they used it to some extent. Achieving a critical mass of adoption for a new communication medium can be exceedingly difficult, even when the users are personal friends of the software author. Though adoption by friends is not strong positive evidence of the usefulness of MessyBoard, their failure to adopt it would have convinced me to abandon the project entirely.

Projecting the MessyBoard space on the wall of our lab was very successful. No one complained about the large public display being annoying or distracting, and I observed a number of spontaneous informal conversations around the projected image. The projection screen was a huge help in drawing attention to the medium and convincing people to install the software, though perhaps it would not have been so important if the startup cost were lower.

My observations of the MessyBoard prototype highlighted startup cost as a major barrier to adoption and long-term use. I informally observed that the mere cost of downloading and unpacking a zip file was a major impediment to adoption. Some users did not install it at all while others would ask someone to install it for them. Initially, users at The University of Virginia used MessyBoard only on one member's computer rather than installing it themselves. This ultimately led to the decision to re-implement MessyBoard from scratch so that it would run in a web browser.

I also observed that the limited space led to serious problems with clutter. Users seemed much more likely to post content on an empty board than they were to remove content posted by others in order to make room for new material. It often seemed that users had an accurate idea of which information was obsolete and which information should be preserved, but they were reluctant to delete it because they were not completely sure and there was no way to recover old material. This observation led to the addition of the history feature in the current version of MessyBoard as a way to instill confidence that deleted material could be recovered, thereby encouraging users to casually delete old content. I later discovered that the history mechanism has great value, even if it is almost, or never used by a group; its real value lies in giving social permission to delete items currently on the board, knowing they can be retrieved if necessary.

I observed that users needed to know who had authored which notes and they used different colors to identify content that they had posted. This led to the addition of a *name tag* feature in the prototype: Users created name tags for themselves which were represented as small rectangles in the MessyBoard space. By changing the font and color of the name tags they could change the default colors of their notes. A user could log in by clicking her name tag. Some users liked having the collection of name tags on the board as a sort of roster, but the name tags occupied precious space and other users felt that the space could be put to better use. In the current version of MessyBoard, users can log in and edit their default colors and font through the menu.

My observations of the MessyBoard prototype did not lead to a general conclusion about how different groups would use MessyBoard. It seemed that some of the use may have

been due to the novelty of the medium and no coherent pattern of use emerged. The two groups used the medium very differently, and the Info Cockpits group used the medium in different ways at different times.

5.4 Automation and Scripting

Users' control over the content on MessyBoard is a fundamental design principle. At the same time, users observed that perhaps some automated content would draw more attention to the medium on a regular basis or liven up the board during dull periods. Users were also interested in adding interactive games to the MessyBoard space, as they eventually lost interest in playing games like pong (Figure 5.8) without any real rules.

To facilitate rapid prototyping of automated behaviors, I integrated the Python scripting language (van Rossum & de Boer, 1991) into the MessyBoard client. I created an object-oriented Python API (the *Messy API*) for manipulating the contents of MessyBoard and a simple development environment called *MessyDev* for testing and debugging Python scripts. The API and implementation of MessyDev are described in a UIST 2002 poster abstract (Fass & Pausch, 2002).

I released MessyDev to the Stage3 research group at Carnegie Mellon University in order to see how they would use the Messy API and whether or not it was sufficient for the things they wanted to do. These scripter writers used MessyDev over a two-week period and they wrote several interesting behaviors:

- Sending the text of a note to Google's image search (Google, 2005b) and putting the first resulting picture on the board
- A Sliding Piece Puzzle that makes a puzzle out of any picture on MessyBoard
- Automatically moving objects so that they don't overlap
- Automatically posting images from a USB camera on MessyBoard at regular intervals
- A news ticker that automatically retrieves headlines and links from RSS feeds (RSS-DEV Working Group, 2005)

Though some of these automated behaviors were promising, none of them seemed to fundamentally alter the nature of the medium or lead to new and unexpected uses of MessyBoard. Thus, rather than turn these rough prototypes into robust code that I could deploy outside of the Stage3 lab, I chose to focus on improving the core properties of the medium.

5.5 Java MessyBoard Carnegie Mellon Deployment

As the new Java version of MessyBoard neared completion, I deployed it opportunistically to a total of 67 groups that I thought might benefit from using it. Different groups began using MessyBoard at different times and for different purposes. I changed my methodology over time in order to explore questions that were raised by my observations. In general, my main sources of data are logs of MessyBoard activity (for all groups), answers to open-ended interview questions (for earlier groups), and results of a survey (for some later groups).

In the following analysis, I separate groups of MessyBoard users into two classes: large groups and small groups. The four large groups range in size from 20 to 84 members and the 63 small groups range in size from one to 15 members. In general, the smaller groups are focused on a particular set of tasks or goals while members of the large groups tend to work individually or in smaller groups on unrelated tasks. One of the large groups comprises students in a large course. Since this group is qualitatively different from the other three large groups, I describe it separately from the other large groups in section 5.5.2.2.4.

The main finding from my observations is that the large groups are more likely to adopt MessyBoard than small groups. Of the four groups with 25 or more members, three of them used their MessyBoard spaces for more than 30 weeks and all three were still in use at the end of the analysis period. (The fourth group is the qualitatively different group mentioned above, and this space was not used much at all.) Many of the smaller groups did not use MessyBoard at all, and only 24 out of 63 of them used it for more than four weeks. (Some of the small groups were disbanded, but all of them existed for well over four weeks.)

The qualitative patterns of use were also quite different. For the large groups, I observed a constant commotion of activity with a mixture of playful behavior and more structured work-related behavior. For the small groups usage tapered off in most cases. Among the small groups that did use MessyBoard over the long term, their patterns of use were highly variable. However, one commonality was that small groups engaged in very little playful behavior on MessyBoard.

5.5.1 Large Groups

I deployed MessyBoard to three large groups which I refer to as A, B and C. In this section I discuss the details of my deployment to each group and my observations of each individual group's use. I conclude with a summary discussion of commonalities and differences in MessyBoard use for these large groups.

5.5.1.1 Methods

Though my methodology changed over time, I used two instruments to collect data for all three of the large groups. One is the logs of activity created by the MessyBoard server and the other is an online survey.

I selected large groups based on personal connections and familiarity, but I did not personally know most of the members of any of these groups prior to my study. These groups decided whether and how to use MessyBoard on their own. In the interest of full disclosure, I note the following relationships between myself and the participants:

- The leader of group A (an active member of the group) is a member of the thesis committee for this dissertation
- The chair of the department in which group C resides (not an active member of the group) is a member of the thesis committee for this dissertation
- Members of group B worked in an office adjacent to mine both before and during the study.

The MessyBoard server keeps log entries in a text file and it appends a new entry any time a communication is received from a client. (The only exception to this is the stream

of packets that are sent as the user interactively moves or resizes an object. The packets sent at the beginning and end of an interactive operation are logged.) The log entry indicates the kind of communication that was received (element creation request, property change, etc.) and any other relevant information such as the unique ID of the element and the identity of the user sending the request.

After the system was deployed, the logging system was improved in order to log additional information. In the improved system, all log entries indicate whether the communication comes from the MessyBoard applet, application or screen saver. In addition, the client was instrumented to notify the server of user activity that does not result in a change to the content on MessyBoard. For example, the client sends notifications to the server when the mouse is moved over the MessyBoard window, when the screen saver is dismissed with the Escape key, or when the stand-alone application is minimized or maximized. I refer to these notifications as *passive* events. Logging of history requests was added and then improved in order to log the time intervals that clients were requesting.

The server log captures records activity at a very fine grain: every packet sent from the client to the server results in a log entry. However, I expected that some users would carelessly create new objects on MessyBoard with little regard for existing content while others would meticulously place their new contribution and perhaps rearrange the surrounding content in order to avoid overlap or create an aesthetically pleasing arrangement. I wished to count both instances of use equally. I therefore defined a *transaction* as a period of continuous MessyBoard use where the user is idle for no more than 30 seconds. In this context, the term transaction refers only to the temporal spacing of events in time, not necessarily to a semantically meaningful set of actions. In the data that I present, both active and passive events are grouped into transactions.

To quantify the length of time that a group uses their MessyBoard space, I define the *life span* as the amount of time between the introduction of the MessyBoard space (the time that I make its presence known to the users) and the last time it was modified by a user. However, this measure can be misleading. My observations of the logs for individual

MessyBoard spaces revealed that spaces were often modified regularly for a time, left unmodified for a long time, and then modified once or twice after the long period of silence. These final modifications are minor and probably insignificant. To correct for this phenomenon, I define the *trimmed life span* as the time between the creation of the space and latest modification that is at least 48 hours before the final modification. Thus, spaces that are continually in use will lose only 48 hours by this definition, but spaces that lay dormant for long periods with one minor modification at the end will have their life spans shortened to the period where they were actually in use.

I announced an online survey to members of each of the large groups on December 6th, 2005, at which point all of the groups had been using MessyBoard for at least three months. The survey contained items intended to measure participants' attitudes about MessyBoard and their group in general. Survey items are listed in Appendix C. Participants were invited to fill out the survey via e-mail, and the invitation contained a URL that allowed them to fill out the survey using a web browser. The results were processed and stored using a server-side Python script (van Rossum & de Boer, 1991). The response rates for each group are listed below.

- Group A: 6 / 20 (30%)
- Group B: 15 / 25 (60%)
- Group C: 7 / 84 (8.3%)

In all cases, I stopped collecting data on January 20th, 2005.

5.5.1.1.1 Group A

When the new Java version of MessyBoard was nearing completion, I deployed it to a large research group at Carnegie Mellon in order to gather early feedback and observations and find bugs. This group, which I will refer to as group A, has been using MessyBoard since March 2004 and they continue to use it at the time of this writing.

I set up a MessyBoard space group A on a dedicated server such that they could access it at the URL www.messyboard.com/groupA. I later switched the domain to messy-

board.org and sent out an announcement to the group about the change. The group initially used an early version of the new Java MessyBoard implementation which was similar in most respects to the current version described in Chapter 2. Since group A started using MessyBoard, the following features were added: identities, background images, and a new look for notes (notes were originally rectangular with no drop shadows). Group A got each of these features in automatic upgrades as they were implemented.

Though the leader of group A is involved in the present research, most members of the group were strangers to me at the time they started using the software. Thus, observations of this group may be more credible as an indication of how groups will use MessyBoard in general as compared to my previous observations of friends and acquaintances using the prototype version.

I discussed MessyBoard with the group leader first and she decided that it should be used by members of group A to sign up for their weekly meetings before any other members knew about MessyBoard.

Several months later, when the software was ready, I set up the MessyBoard space for group A and gave them a brief demonstration at their weekly meeting. I then installed a projector in their shared workspace to display MessyBoard. I spent time in their shared work space and met with them individually in order to make sure that they knew how to use MessyBoard and to troubleshoot technical problems.

Group A meets once a week in the evening in order to socialize over dinner and discuss research papers in their field. I attended several of these meetings and participated in the discussions. Prior to the introduction of MessyBoard, the group's administrative assistant would send an e-mail message announcing the meeting and requesting RSVPs so as to order the right amount of food. Though attendance was always high for the meetings, the group leader was annoyed with the low number of RSVPs and we hypothesized that MessyBoard might improve the response rate. The group leader and administrative assistant agreed to give me copies of these announcements and all of the responses for several weeks before MessyBoard was introduced.

The MessyBoard server logged all MessyBoard activity for this group. The logging system was changed on June 9th, 2004 in order to log more detailed information such as passive activity (for example, moving the mouse over the MessyBoard window without actually modifying the content on the board), use of the history feature, and a hash of the client IP address in order to determine which transactions originated on the same computer without recording the actual IP address.

I conducted open-ended interviews with several members of the group shortly after introducing MessyBoard in order to understand what people were working on and the nature of their collaboration with one another. (Interview questions are listed in Appendix D)

5.5.1.1.2 Group B

Group B comprises all of the graduate students in a department. I had initially deployed the MessyBoard prototype to members of this group. They used it very little and eventually stopped using it completely. Some current members of the group remember that initial deployment, though many new members have arrived since then and all were willing to try it again.

I deployed the fully functional Java version of MessyBoard to group B and my deployment strategy differed from that for other groups. At the time that I was ready to deploy, this group did not have a regularly scheduled meeting that all of the members attended, thus I could not give a demonstration. Instead I roamed through their open-plan lab space, introduced myself to individuals one at a time, demonstrated the software to them and asked them if they would be willing to use it and participate in this study. I encouraged them to install the screen saver, though some of them did not because they were Macintosh users or because they simply preferred not to.

Initially this group had no large public display. They moved into a new office space on in August 2004 and I set up a projector in this space to display MessyBoard. In December 2004 the projector was removed.

I observed this group closely and I got to know some of them fairly well. Initially my office was close by to their lab space, and when they moved I took an office in a suite that they occupied. I chatted with them daily and ate lunch in their designated social area.

This group used MessyBoard in order to allocate offices in their new space when they moved, and this was an especially interesting event, as described below. I conducted open-ended interviews with several members of group B soon after this event had occurred. These interviews were largely unstructured. I asked them to tell me what had happened during the relevant time in their own words, then I asked each of them how they felt about using MessyBoard during this time.

5.5.1.1.3 Group C

Group C comprises all of the students in a 2-year professional Masters Degree program. As with group B, I had initially deployed the prototype version to this group. In contrast to group B, Group C posted a great deal of content on the beta version. Most of their use was playful in nature. Since their last use of the beta version, half of the students in the program had graduated and a new class of incoming students had replaced them.

I introduced the Java version of MessyBoard to group C by giving a demonstration at the end of a scheduled presentation where most of them were present. The first-year students in the program share a single large room and I put a projector in this space. I also had the system administrator configure the new students' workstations so that MessyBoard was the default screen saver. The projector bulb failed early on and it was never replaced.

5.5.1.2 Group A Observations

Figure 5.10 summarizes group A's MessyBoard activity since it was deployed. One data point is plotted for each day, and the Y value of each point represents the number of transactions occurring in a one-week window centered on noon of that day. Figure 5.11 is a screen shot depicting the typical appearance of their MessyBoard space.

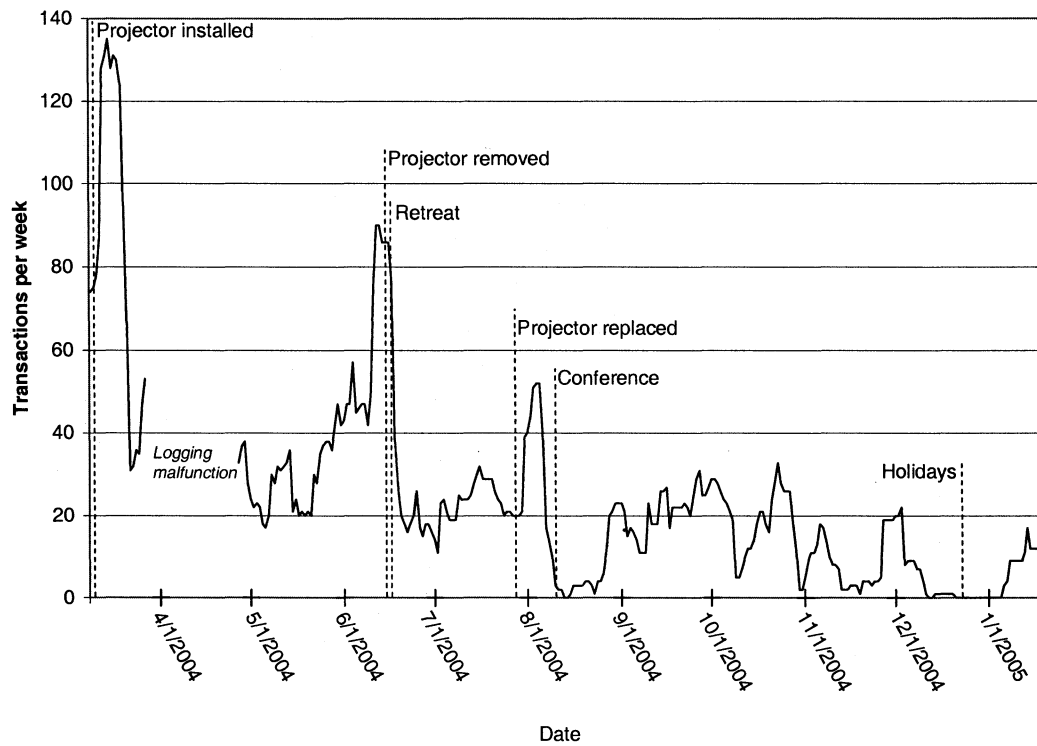


Figure 5.10. MessyBoard usage and events for group A.

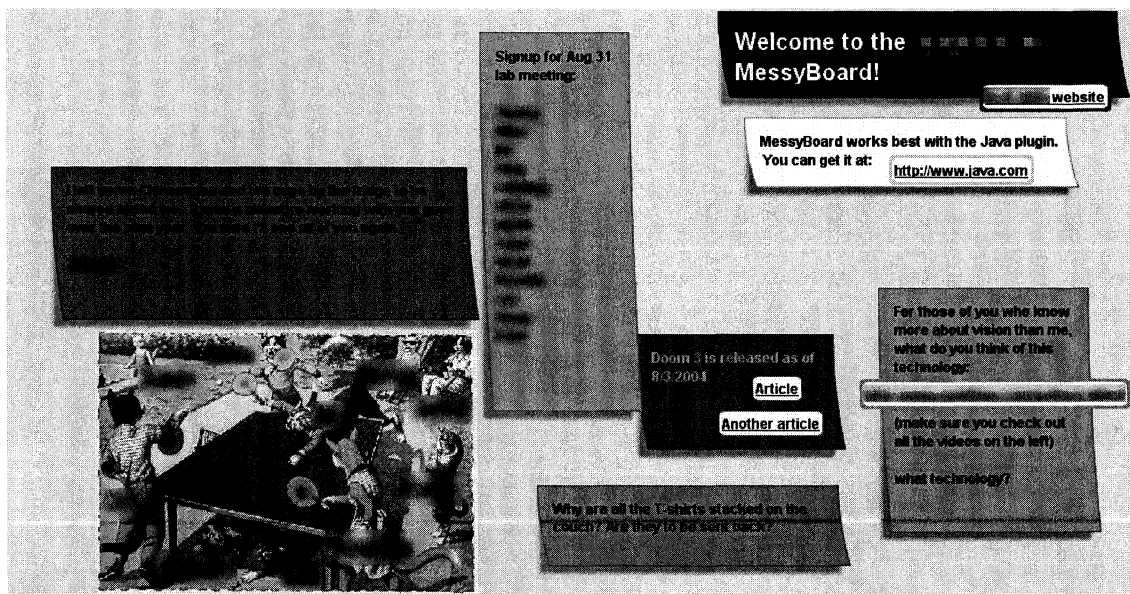


Figure 5.11. Group A's MessyBoard space.

The dominant use of MessyBoard for group A is signing up for their weekly meetings. The leader sends out an e-mail announcement to the group each week, reminding everyone to RSVP via MessyBoard. On the day of the meeting, the group's administrative assistant looks at the list and orders the appropriate amount of food. She then places a note over the attendance list to inform group members that food has been ordered and listing the name and phone number of the restaurant. In one instance, a group member reported that this information was useful when the food did not arrive on time and the assistant had already left for the day.

Though I had hypothesized that Group A might use MessyBoard for general discussion of research-related topics in their field, this only occurred once. A group member later told me that he had started the discussion in a deliberate attempt to get group members to use MessyBoard more often, but this pattern of discussion on MessyBoard did not continue.

Though back-and-forth discussion was rare, members of group A did occasionally post material for others to see. Some of this was related to their research and some of it was purely for enjoyment. Most of the material could be included in both categories. Figure 5.12 shows two examples.

Two particular instances of MessyBoard use stand out from this group's general pattern of weekly meeting sign-ups and occasional postings. The first is the table tennis tournament sign-up list, shown in Figure 5.13. Though many group members signed up, the tournament never actually occurred. The second instance is the group retreat sign-up list, shown in Figure 5.14. The group schedules a yearly retreat to discuss the research topics that they will pursue in the coming year. Each group member gives a presentation and receives feedback from other group members. Group members entered their names and the topics of their presentations into a note on MessyBoard before the retreat.

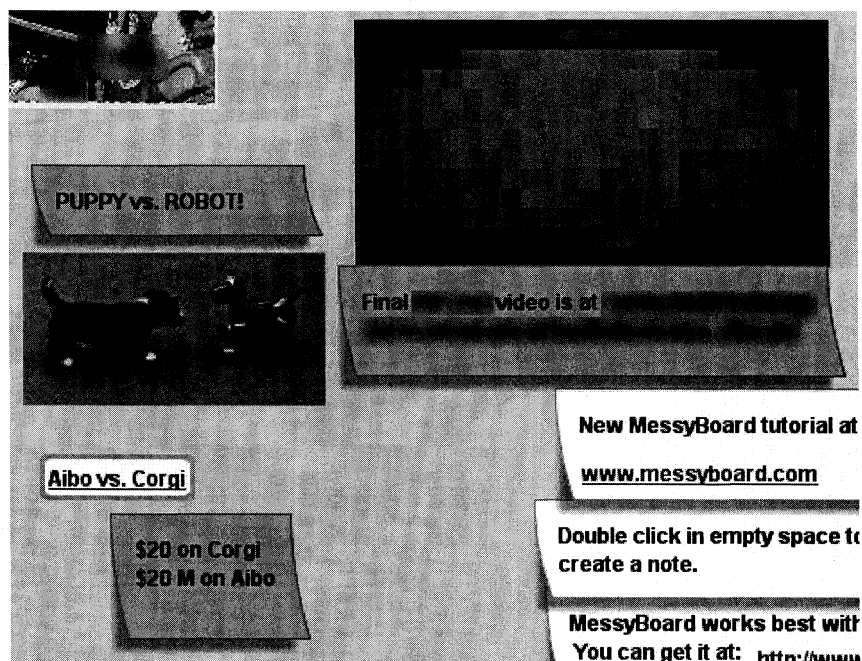


Figure 5.12. Two pictures on Group A's MessyBoard space. Both are related to the group's research, at least tangentially, and both are also posted for the enjoyment of the group members.

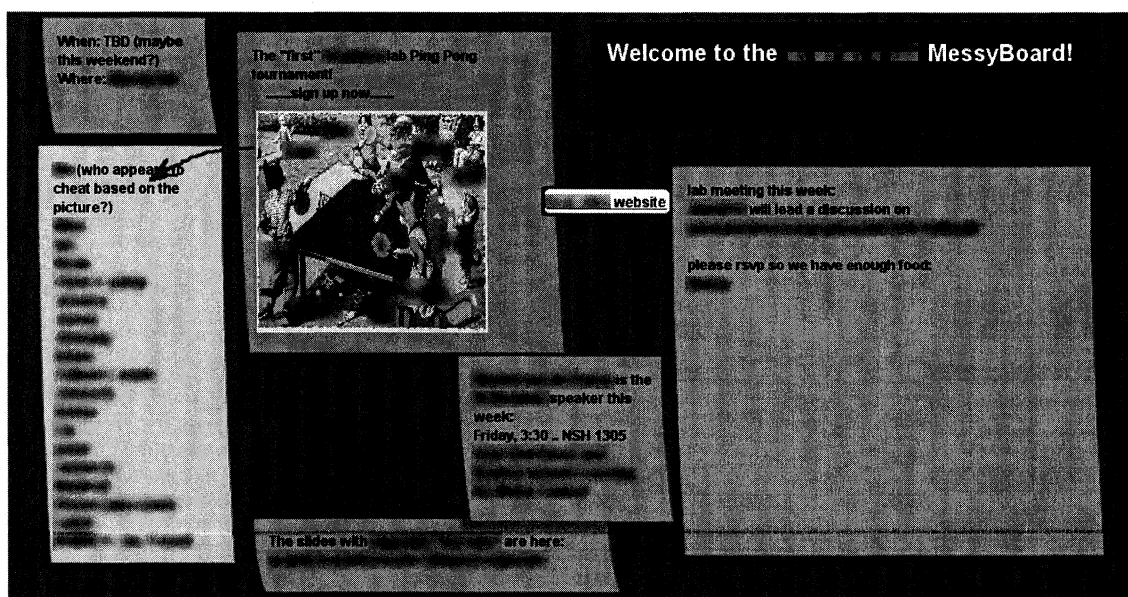


Figure 5.13. Group A uses MessyBoard to sign up for a table tennis tournament (at left).

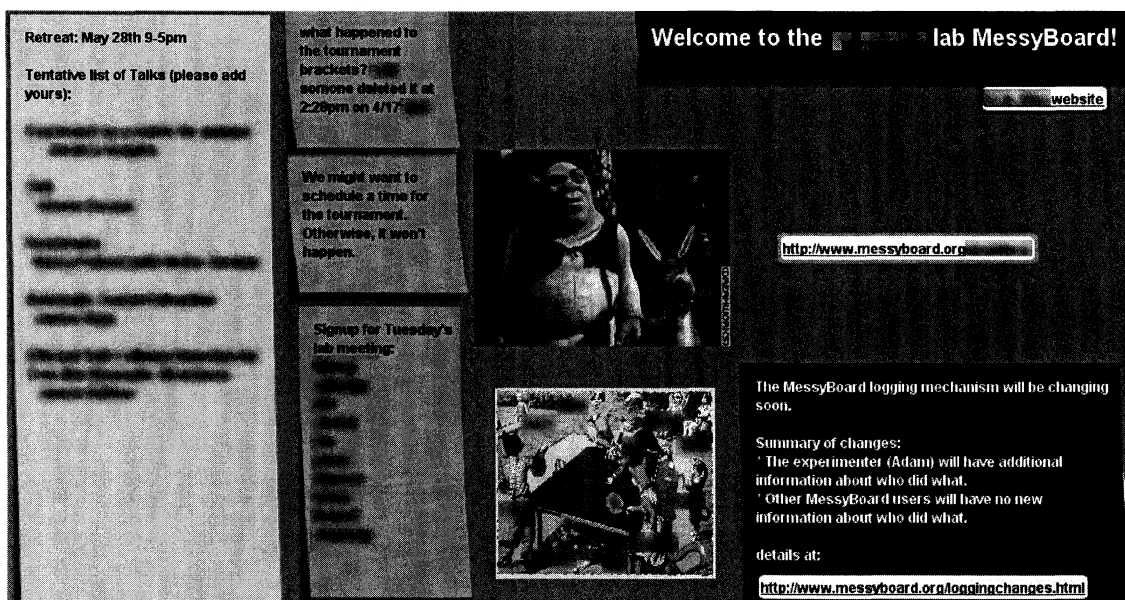


Figure 5.14. Group A uses MessyBoard to sign up to give talks at their upcoming retreat (at left).

On June 15th, 2004, the projector in group A's space malfunctioned. The projector was removed and shipped away for repairs. During this time, as shown in Figure 5.10, group A continued to use their MessyBoard space. Though it may appear that they used their MessyBoard space less often when the projector was not available, the removal of the projector is confounded with the occurrence of the group's retreat which happened at nearly the same time. Thus, it is likely that members signing up for the retreat accounted for much of the activity before the projector was removed and the retreat is responsible for the drop in activity rather than the loss of the projector.

Though they still used MessyBoard without the projector, they commented that they thought MessyBoard was far less useful without it.

5.5.1.3 Group B Observations

We observed that many of the students in Group B removed the MessyBoard screen saver and most of them reported viewing it in a web browser or using a stand-alone application that we provided. Some of the students did continue running the screen saver, and their screens are visible to others. Figure 5.15 shows that the screen saver generates a low level

of activity, but the applet is mostly responsible for the large spikes in usage. The short dotted line plots the number of unique computers that connected to the MessyBoard server using the screen saver within a one-week window centered on noon of each day. Several users installed the screen saver initially, but most of them later removed it.

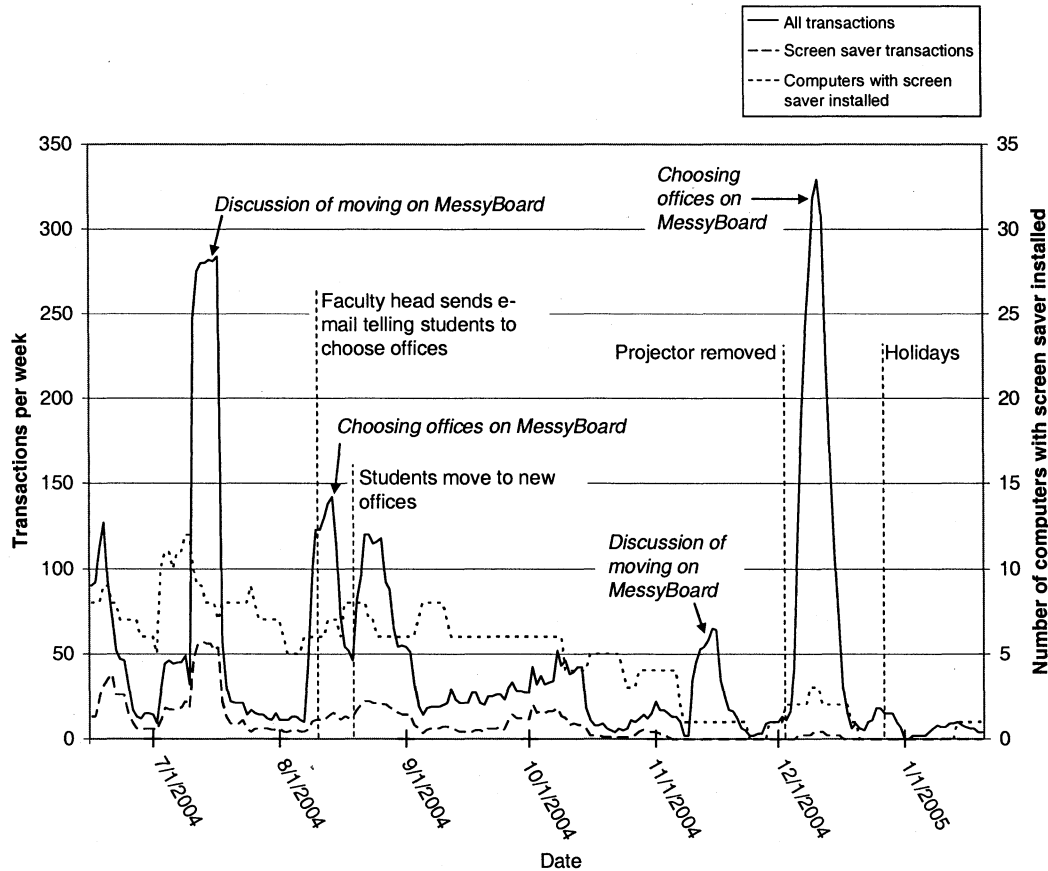


Figure 5.15. MessyBoard usage and events for group B.

Initially, the students used MessyBoard in a playful manner. They posted humorous pictures, announcements for social events and other notes on miscellaneous topics as shown in Figure 5.16.

Five weeks after they began using MessyBoard, the faculty head of the program informed the student ombudsman that new office space was becoming available and some of the students would move into that space. The ombudsman sent e-mail to all of the students to

invite discussion about how to fairly allocate people to the new offices, and he suggested that they discuss it on MessyBoard in order to avoid excessive e-mail traffic.

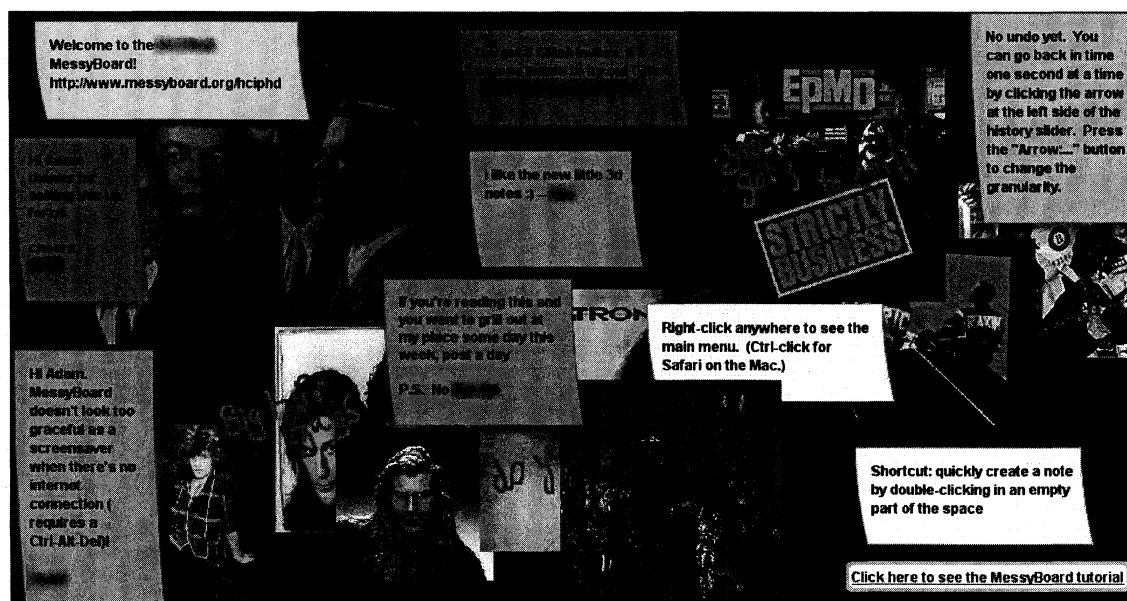


Figure 5.16. Group B uses their MessyBoard space to post humorous pictures and announce social events.

A lively discussion ensued over the next two days and MessyBoard was covered with small notes expressing various opinions about how to use the new space and how to fairly allocate the most desirable locations. This is shown in Figure 5.17. Someone eventually cleaned up all of these notes, and the ombudsman sent a summary of the discussion to all of the students via e-mail. The students also left two lists of names on the board: a list of people who wanted a new office and a list of people who wanted to stay where they were.

In general, interviews revealed that the students were ambivalent about using MessyBoard for this discussion. They like it because it kept the conversation from “clogging” their e-mail inboxes. They also liked being able to see and respond to opinions in real time. However, most students were frustrated by the chaotic nature of conversation on MessyBoard. Many of the notes were responses to other notes, but the original notes were sometimes deleted or buried under new content. Not all members of this group used the identity feature, so authorship was not always clear.

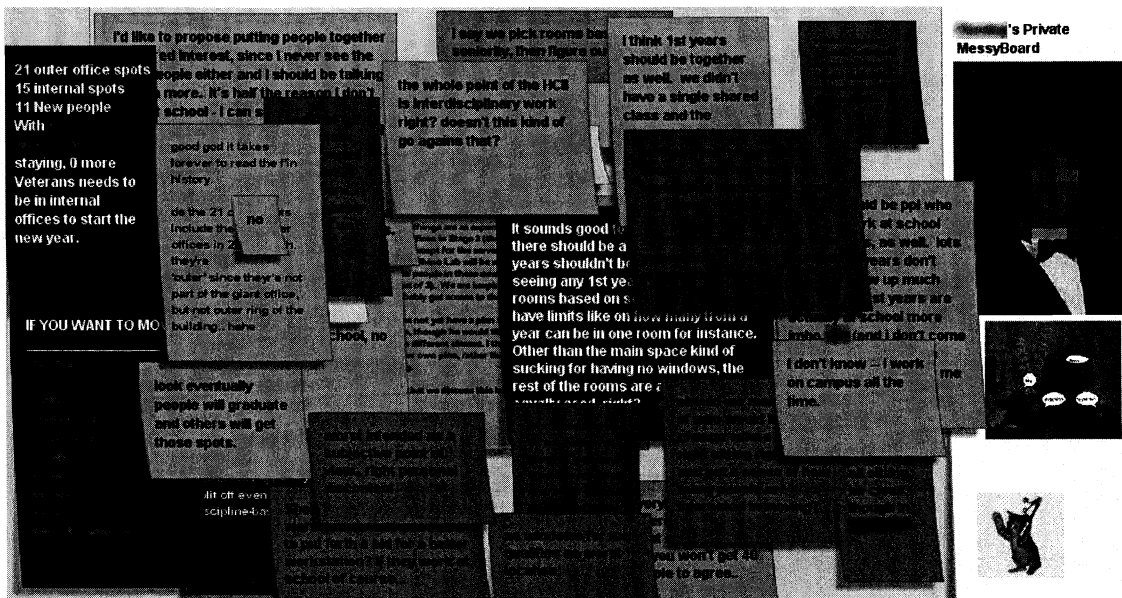


Figure 5.17. Group B has a discussion about how to fairly assign people to offices in their new office space.

Students who attempted to use MessyBoard's history feature to view past comments were disappointed. The history scroll bar allows users to specify an absolute time by clicking, but it does not provide an easy way to find times when important changes occurred. Performance was slow due to the amount of activity and a naïve implementation.

Four weeks later, the faculty head informed all of the students via e-mail that they would be getting a different set of offices than what had been originally announced and that all of the students would choose a new location one at a time based on seniority and a randomly generated order within classes. He included the random order and a floor plan image depicting all of the available space.

Several students (not including the ombudsman) collaborated to create a choosing process on MessyBoard, as shown in Figure 5.18. A note with all of the students' names in the designated choosing order was placed along the left-hand side. The floor plan image was placed in the center and a note was created for each room. Students were instructed on MessyBoard and via e-mail that when their turn came, they should remove their names from the choosing order, place it on the note corresponding to their chosen room, and then e-mail all of the students and the faculty head. By this point the faculty head had

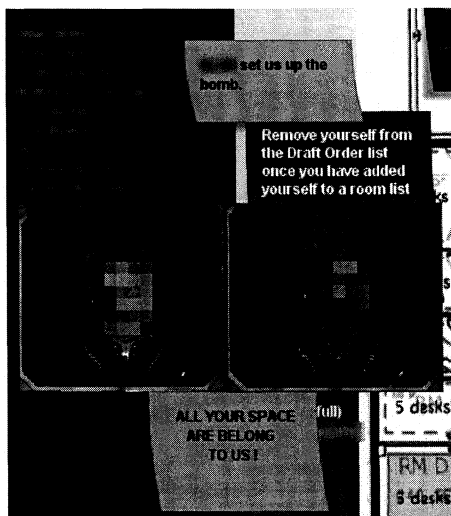


Figure 5.19. Members of Group B make jokes about the office selection process on MessyBoard. The pixelated images are pictures of two faculty members who were involved with the process, edited into a screen shot from a video game. The pictures and the phrase "all your space are belong to us" refer to a poorly translated video game that became a cult phenomenon on the Internet (Sega, 2005).

The students were overwhelmingly positive about their use of MessyBoard in selecting offices. The shared visual representation on MessyBoard made it easy for people to keep track of the selection process as it occurred. Students reported using MessyBoard to select an office at work and from home, and at least one student used it while traveling. In six out of seven interviews, students expressed that they were very happy with MessyBoard for this task. (In the first interview the student simply related the facts and I neglected to ask for an opinion on how well it had worked.) Some of their comments appear below.

"I found it comforting that we could see the progress of people making these decisions and we knew when our turn was coming up and we knew what other people had decided..."

"This gives you an excellent snapshot ... of what the current status is and ... where am I and what's going on now."

"It was nice to have the visual layout on the map and I think the notes were good because it would help us see who was where already."

"This was a big success largely because of the value of being free form ... it's not structured but there's enough capability there to create semi-structured things like these lists where people's names were moved off of the ordered picking list into the room list ... you get to make the assignment in the context."

"This way people could really see where they were going to be living and for something that has a very strong visual element and, uh, not just a conceptual element this is concrete and visual this is ideal and it also allowed a lot of us to do this process in the evenings and the early mornings..."

When group B moved, I set up a projector in their new space to display MessyBoard. However, not all members moved to the new space, so many group members did not see the projected display on a daily basis. As with group A, the introduction of the projector is confounded with another event, in this case the move to new office space. Changes in the amount and nature of MessyBoard activity are likely due to this event, not the introduction of the projector. The lively discussion and the orderly moving process established precedents for future MessyBoard use and these same patterns have been re-instantiated as group members are once again moved to new office spaces, as shown in Figure 5.20.

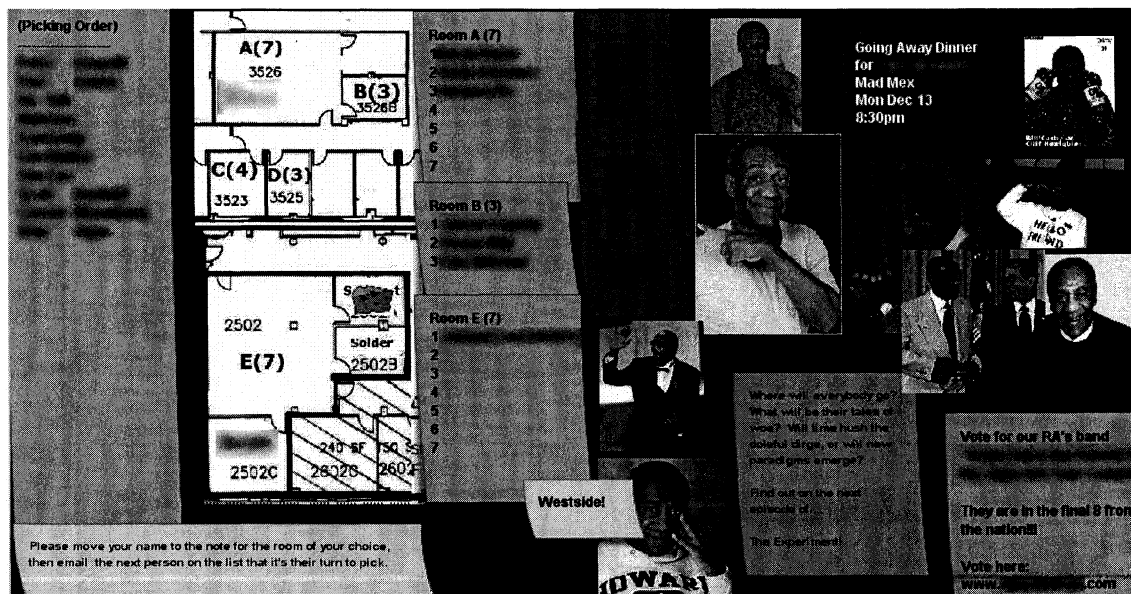


Figure 5.20. Group B uses a similar process to choose offices a second time.

5.5.1.4 Group C Observations

Since group C's MessyBoard space was first announced to the group members, it has been a steady commotion of jokes, announcements, news flashes and other types of playful behavior. Figure 5.21 depicts group C's overall usage level over time. Figure 5.22 and Figure 5.23 are typical of the appearance of this MessyBoard space.

Group C uses MessyBoard quite often to organize social events and other extracurricular activities. This includes looking for teammates for organized sports leagues, and a trip to a local amusement park, among other events, as shown in Figure 5.24.

Another common use of MessyBoard by group C is for humorous pictures, as shown in Figure 5.25. Some of these are digital photos taken by group members, though many of them come from the web. The humor is highly dependent on the group's context.

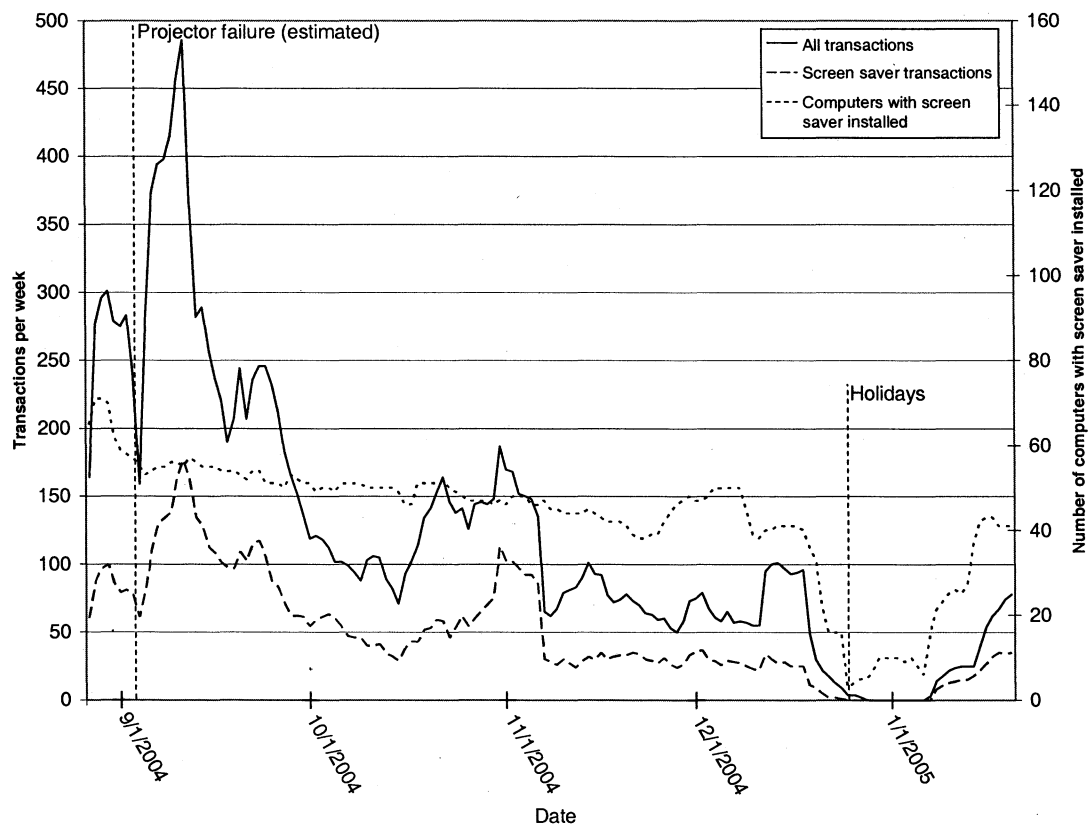


Figure 5.21. MessyBoard usage and events for group C.

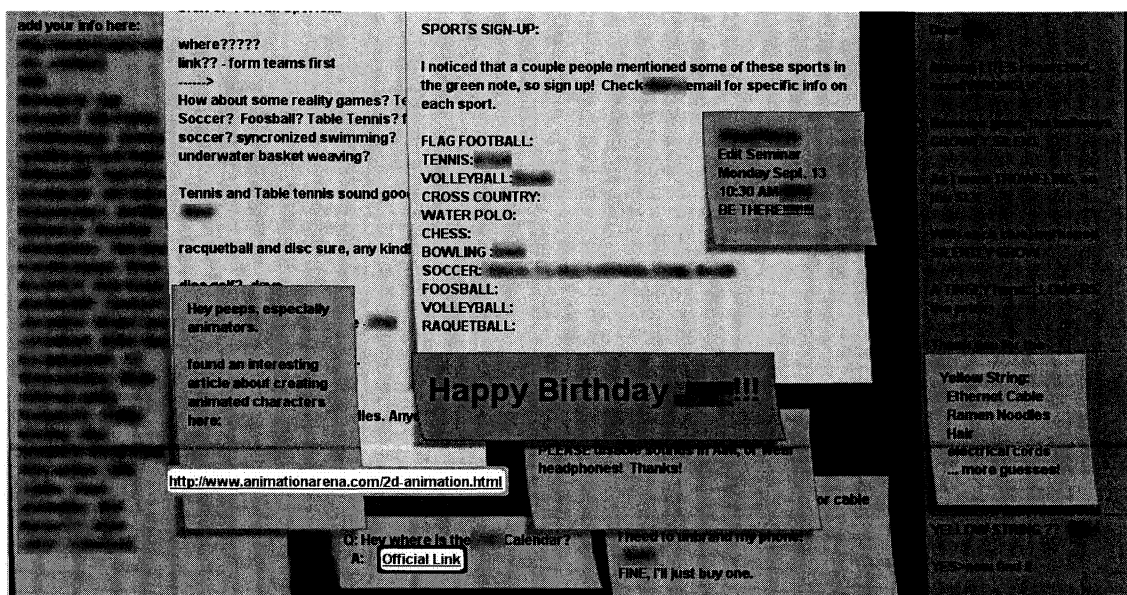


Figure 5.22. Group C's MessyBoard space.

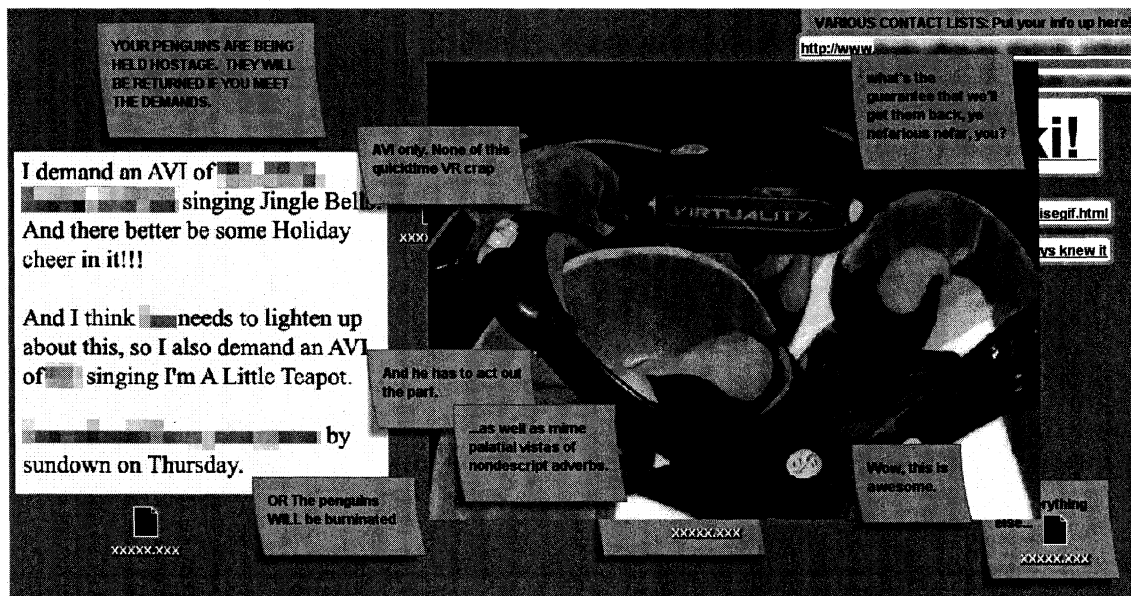
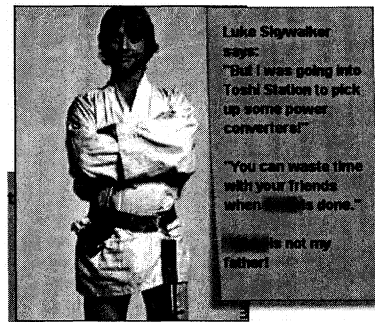
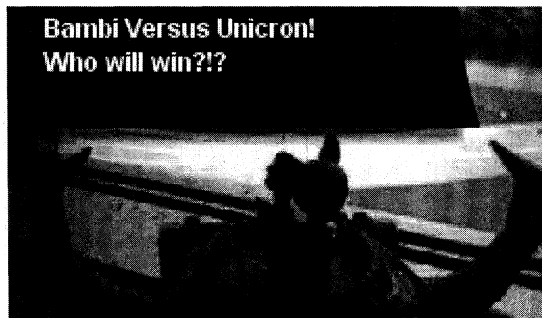


Figure 5.25. Group C posts humorous pictures in their MessyBoard space.

There were several instances of long-term organized behavior on group C's MessyBoard space. In the most striking example, they used MessyBoard to collaborate on the creation of a historical timeline on a corridor wall in their building. Figure 5.26 demonstrates how they used MessyBoard to assign members to different time periods. Other examples of organized behavior include the organization of sports teams mentioned above, a collaborative list of screen names for AOL Instant Messenger, and a collaborative list of birthdays for group members.

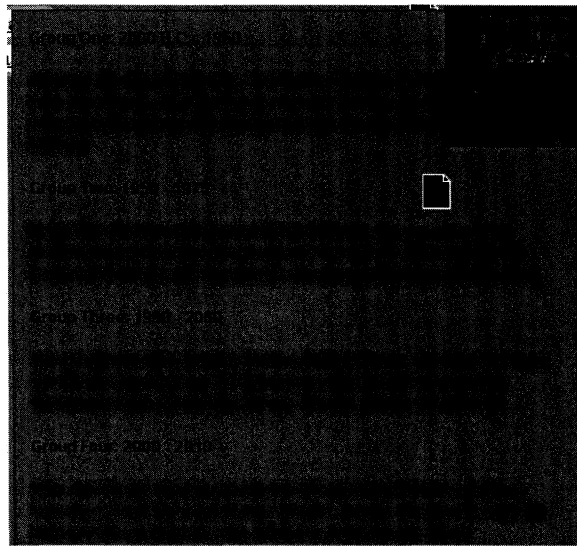


Figure 5.26. Group C coordinates the creation of a historical timeline.

MessyBoard was installed as the default screen saver for incoming students, who compose roughly half the members of this group. As shown in Figure 5.21, the screen saver is responsible for about 41% of this group's total transactions and it is clear that the screen saver is responsible for much of the increased usage during spikes in activity. The MessyBoard screen saver was installed on over 60 computers including personal computers used by individual students and shared computers in public labs, and it was still running on over 40 of them at the end of the semester.

The bulb for the MessyBoard projector in the first-year students' room malfunctioned at some point early in the semester and it was never replaced. The exact time is unknown,

but one student estimates that it stopped working in early September. MessyBoard use remained high throughout the semester despite this as shown in Figure 5.21.

5.5.1.5 Discussion of Large Group Observations

Groups A, B and C are similar in that they all use MessyBoard for a mixture of work-related and playful activity. In all three groups, work and play were mixed over time and at any given time there was likely to be both work-related and playful content on the board.

There is a major difference between the groups in their amount of MessyBoard use. Figure 5.27 compares the amounts of MessyBoard activity for groups A, B and C. The amount of activity is positively correlated with the size of the group, but other factors are also very likely involved in this difference as well. For example, groups B and C were known to interact with each other more often than group A prior to the introduction of MessyBoard. All three groups had un-moderated e-mail distribution lists before they began using MessyBoard and based on comments from group members it appears that groups B and C used their lists far more often than group A.

Group A was also different in that structured work-related use (signing up for meetings) made up a larger fraction of their overall use than groups B and C. This was clearly due to the fact that group A had a leader who instructed members to use MessyBoard this way. On the survey, most members of groups B and C indicated that their group had no leader, while members of group A indicated that their group had a few leaders and that those leaders influenced how they used MessyBoard (though they believed that other group members also had influence).

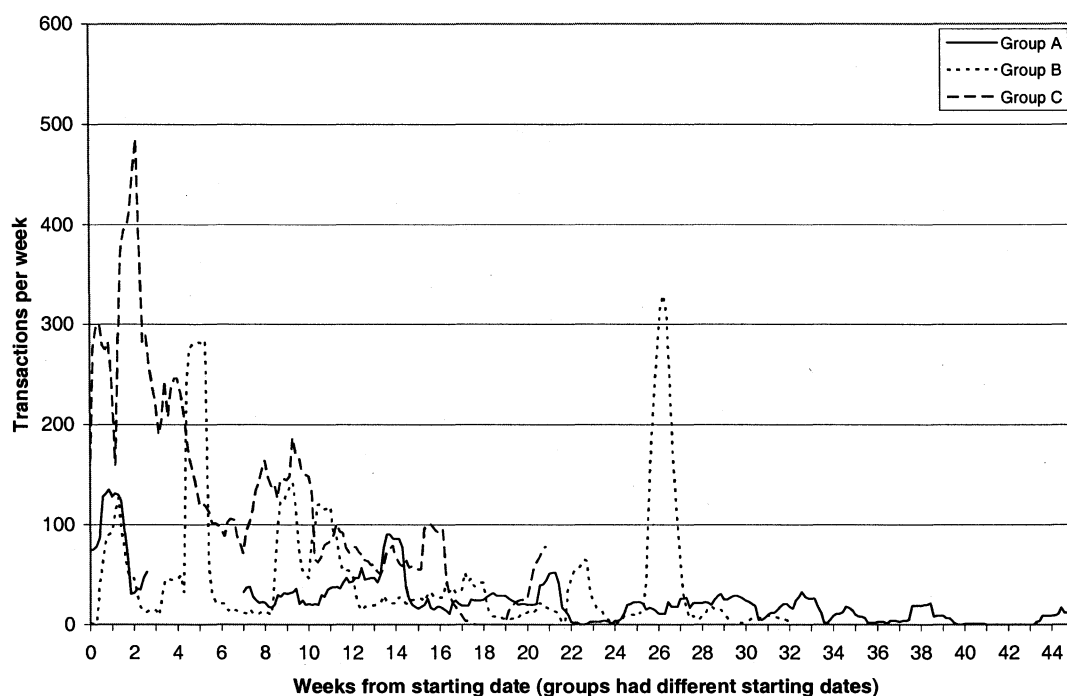


Figure 5.27. MessyBoard usage for groups A, B and C.

With regard to the value of a large public display, the implications of these observations are unclear. Members of group A missed the projector while it was gone, but they still used MessyBoard, likely because of their structured pattern of weekly use. Members of Group B started using MessyBoard and found it most useful before they had the projector, since it helped them accomplish a critical task (moving to a new space.) Group C's projector malfunctioned early on and they still used MessyBoard a great deal, though it is unclear if this is because so many of them used the screen saver or because they are a more interactive group in general.

With regard to the value of the screen saver, these observations lead toward a tentative conclusion that the screen saver can drive day-to-day usage of and attention to MessyBoard, but only if a sufficient number of people install it and refrain from removing it. This was accomplished with group C by having a system administrator install it on all of their computers, as well as on computers in public lab spaces.

For all three large groups, it appears that a small number of users are responsible for most of the activity on MessyBoard. Figure 5.28 is a histogram showing the percentage of MessyBoard transactions that originated from each IP address, and it is clear that a few addresses contribute disproportionately to the total. This argument assumes at least a rough correspondence between IP addresses and users. This is known to hold true in group B. Group A has a few shared computers and group C has many shared computers, which may account for the thicker tails in their distributions. Only static addresses in the 128.2 block are used in this analysis, and these addresses account for 87%, 60% and 81% of the activity for groups A, B and C respectively.

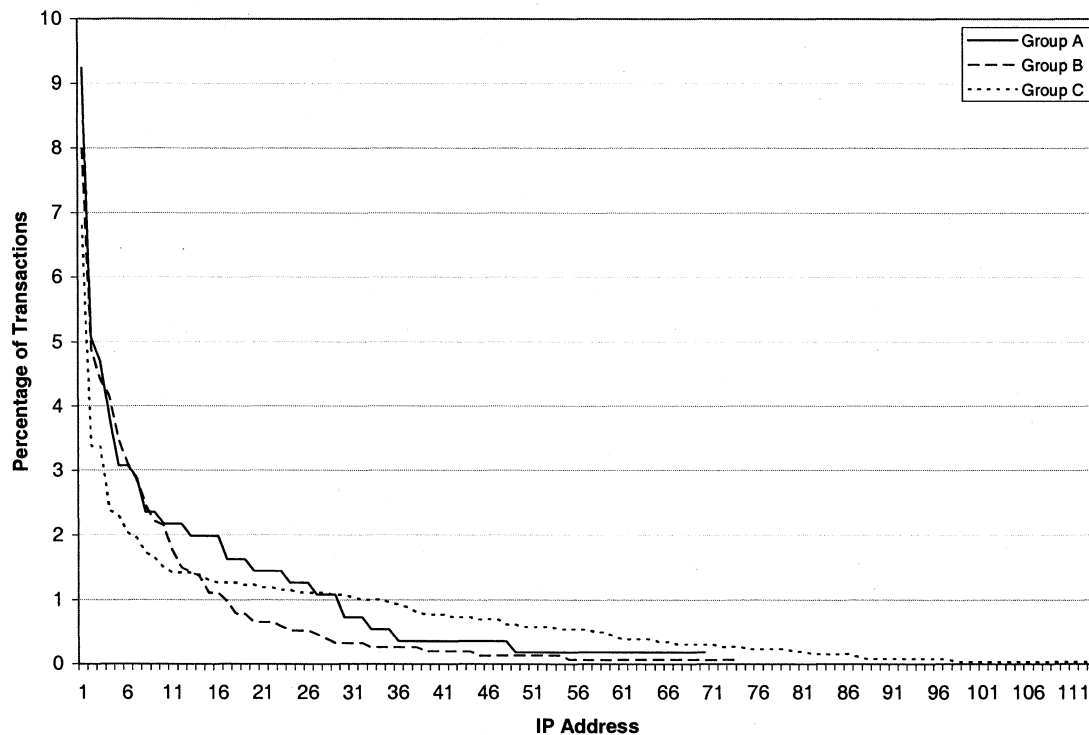


Figure 5.28. Percentage of transactions from individual static IP addresses.

5.5.2 Small Groups

MessyBoard was initially designed for small groups of people with between 5 and 15 members. I had hypothesized that for small groups focused on a limited set of goals

MessyBoard would be an extremely easy way to keep track of their progress and share files.

5.5.2.1 Methods

I contacted friends and colleagues in order to learn about groups that might find MessyBoard useful and I deployed MessyBoard to a total of 37 of these groups. Though I learned of these groups through personal contacts, I knew at most one or two group members in any of the groups. 31 of these groups were made up of students working on course projects and 6 were research groups comprising students, faculty and staff members. The student project groups were drawn mainly from three different courses at Carnegie Mellon University in which I actively recruited participants. I refer to the courses as Alpha, Beta and Gamma. A few groups in other courses contained members who were already familiar with MessyBoard and they contacted me to request a MessyBoard spaces for their groups.

I first demonstrated MessyBoard to the students in course Alpha during one of their scheduled lectures and asked them to contact me if they were interested in using it. Three out of six groups contacted me, and I subsequently set up a MessyBoard space for each group and attended one of their group meetings. At the meetings I helped them install the latest version of Java and troubleshoot technical problems and I gave another demonstration in order to ensure that they knew how to use MessyBoard. These project groups had dedicated offices and I was able to install projectors in these offices to display MessyBoard.

For the student project groups in courses Beta and Gamma, I demonstrated MessyBoard during one of their course lectures and then I created a MessyBoard space for each group based on names and descriptions that I received from the course instructors. I then sent an e-mail announcement to each group that their MessyBoard space was ready for them to use. In all cases, I encouraged users to contact me with problems or feedback at any time.

The MessyBoard server logged all MessyBoard activity for each of these groups as described previously. Again, some of these groups started using MessyBoard before the logging system was updated as described above.

I conducted open-ended interviews with the group members in course Alpha and the interview questions were focused on their collaboration and their use of MessyBoard. (Interview questions are listed in Appendix E).

As with the large groups, I conducted an online survey. Since the survey was developed after the groups in course Alpha had finished their projects, those groups answered the survey questions several months after their groups disbanded. All other groups responded to the survey as they were finishing their group projects.

In all cases, members of the small groups used the full-featured Java implementation of MessyBoard as described in Chapter 2.

5.5.2.2 Student Project Group Observations

5.5.2.2.1 Course Alpha

Course Alpha was a project course for students receiving a Masters' Degree in Human-Computer Interaction. The students were assigned to six interdisciplinary project groups of five students each and they worked together over a semester and a summer. Members of these groups had backgrounds in computer science, design and psychology.

All of the project groups had already been working together for a semester and MessyBoard was introduced to them at the beginning of the summer. The groups were to spend the entire summer working only on their course projects. Three of the six groups chose to use MessyBoard and I refer to these groups as D, E and F. Groups D and E requested a MessyBoard space immediately at the beginning of the summer and group F requested a MessyBoard space one week into the summer. Shortly after I demonstrated MessyBoard, each group was assigned a dedicated project room. For the groups that decided to participate in the study, I set up projectors in their project rooms to display MessyBoard. Figure 5.29 compares the MessyBoard usage for all three groups. Figure 5.30, Figure 5.31 and

Figure 5.32 indicate when these groups had projectors to display MessyBoard in their group project rooms.

Group D designed a scientific graphing calculator. The group had no leader during the summer, though one of the members had served as the leader during the previous semester. The former leader was already familiar with MessyBoard but had never used it to collaborate with colleagues. This group member was enthusiastic about MessyBoard and may have convinced the others to try it. I did not give this group a projector until they had already been using MessyBoard for five weeks. Group D's MessyBoard usage is depicted in Figure 5.30.

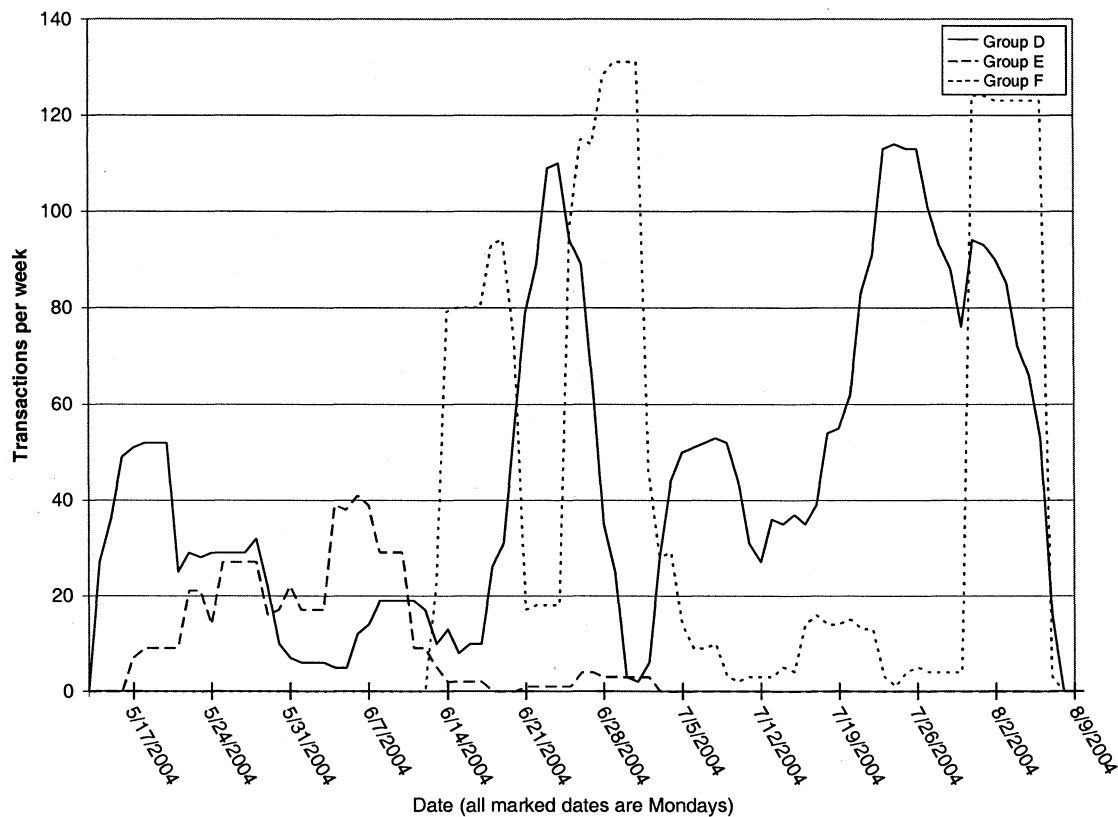


Figure 5.29. MessyBoard usage for groups D, E and F.

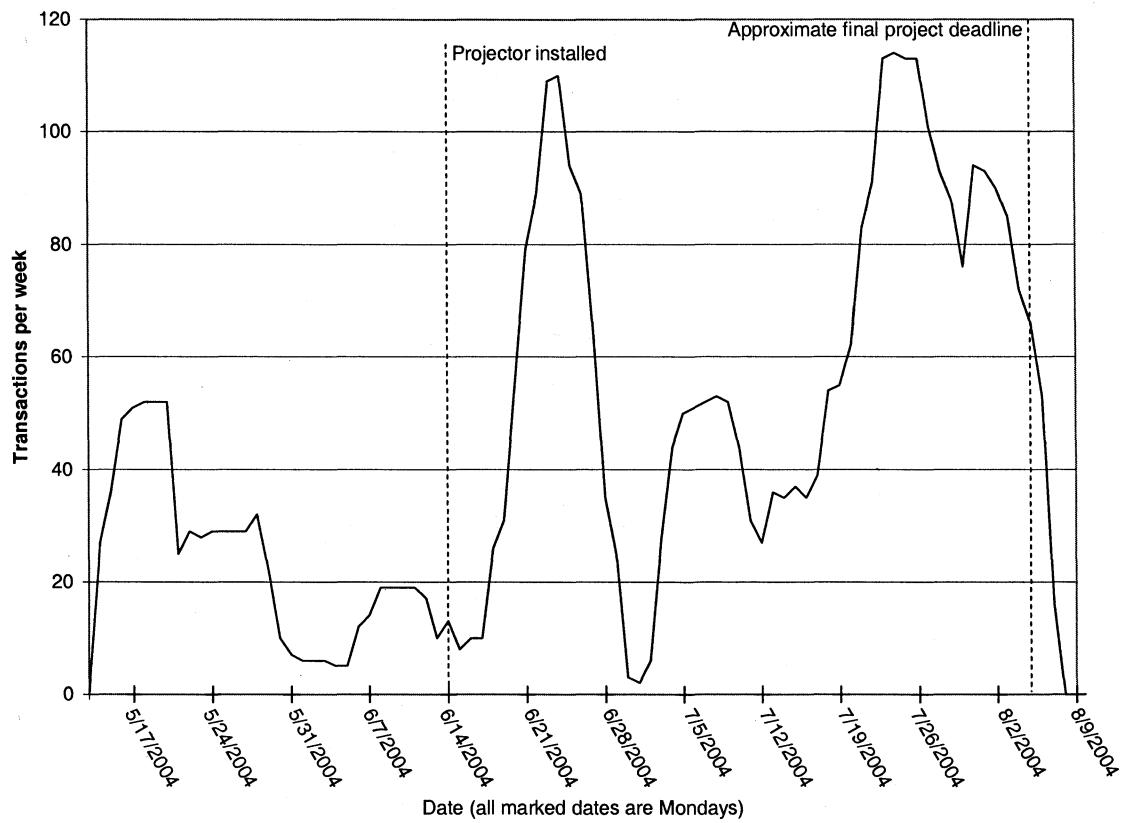


Figure 5.30. MessyBoard usage and events for group D.

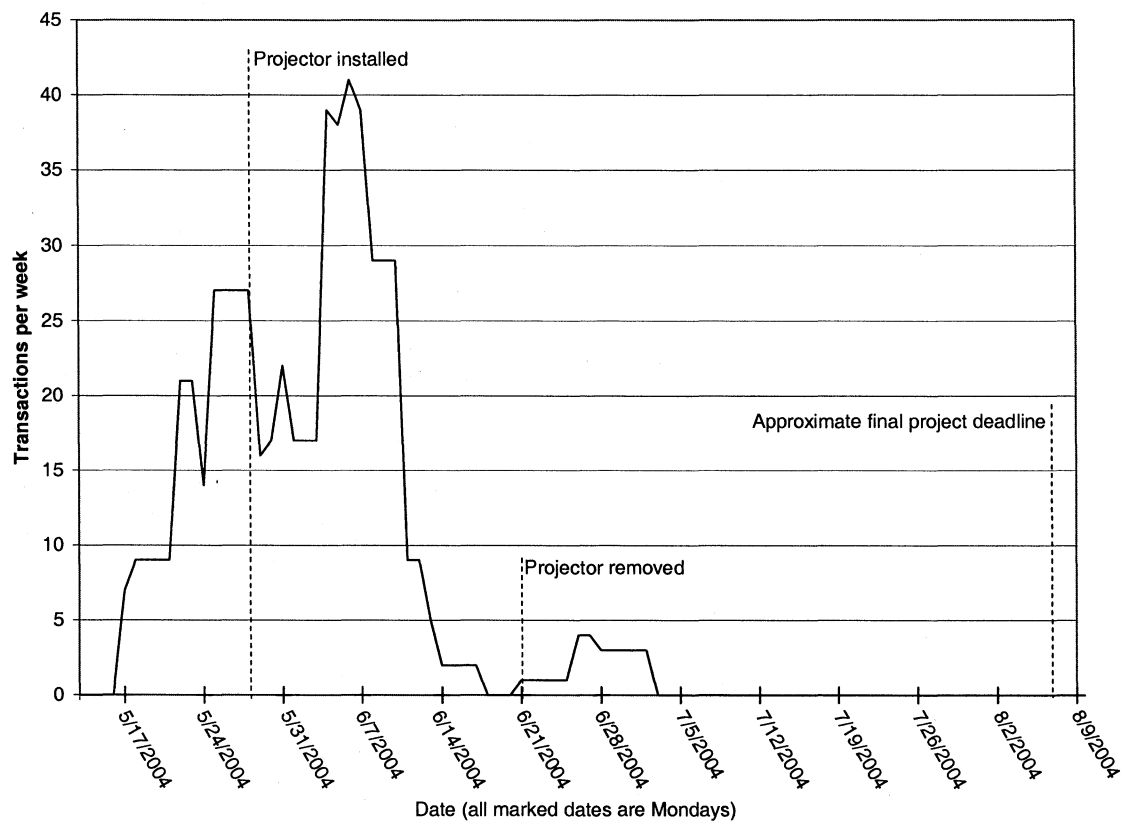


Figure 5.31. MessyBoard usage and events for group E.

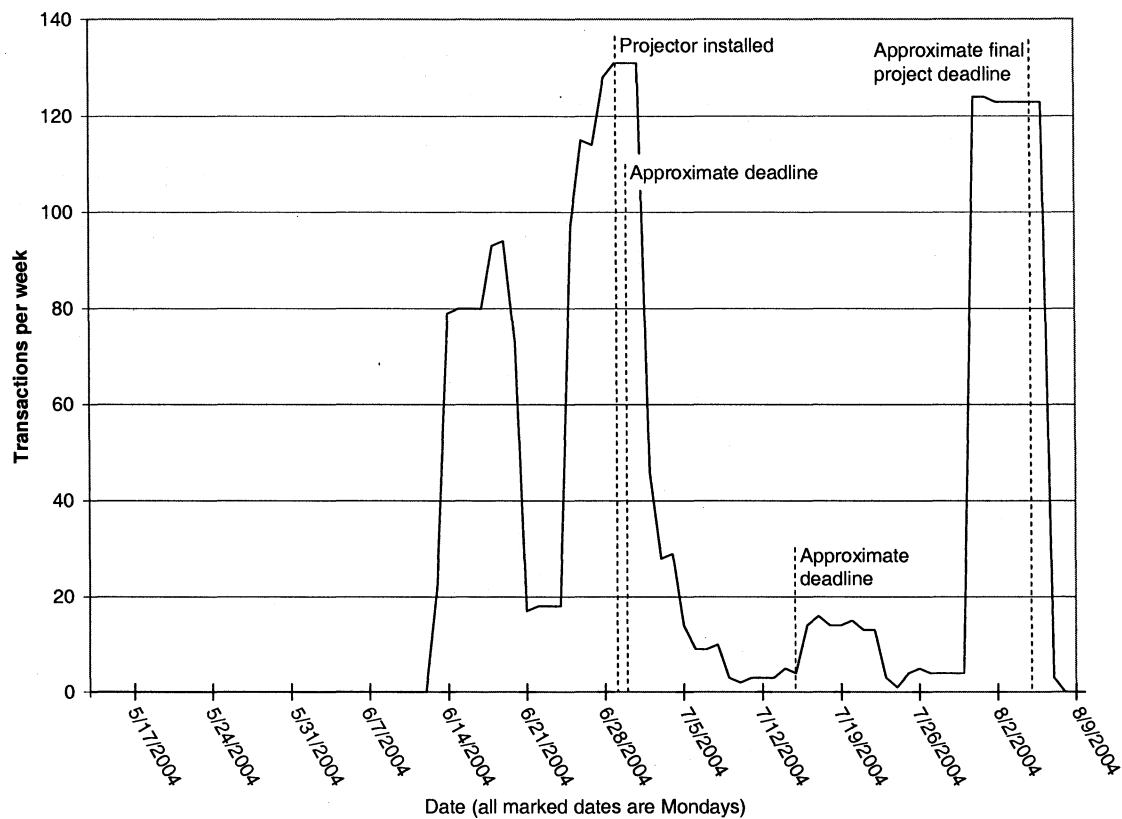


Figure 5.32. MessyBoard usage and events for group F.

All members of group D reported substantial use of MessyBoard. They used MessyBoard frequently to share files such as text documents, presentation slides and source code for their prototypes. They often grouped the file icons spatially and annotated them with notes as shown in Figure 5.33. In interviews, all of the group members reported that they liked using MessyBoard this way, though some emphasized the ease of uploading files using drag-and-drop and others emphasized the fact that it was easy to find current files as compared to FTP.

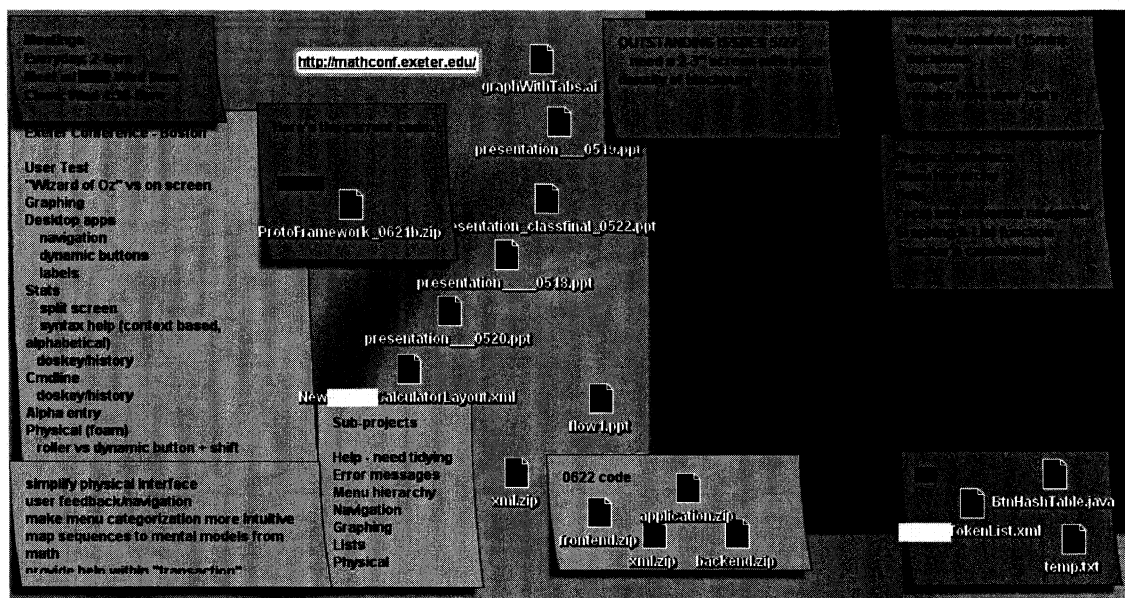


Figure 5.33. Group D's MessyBoard space.

Members of Group D also used MessyBoard during their meetings in order to record lists of tasks, design changes and outstanding issues. Most of these notes are in list form with a title and many short items, one per line as shown in Figure 5.33. Group members reported that MessyBoard worked well as a place to store this information. MessyBoard did not work well as a communication medium for exchanging messages when they were not together since they were not confident that others would receive them.

Group E also designed a scientific graphing calculator. None of the members of this group were familiar with MessyBoard. I set up a projector in this group's space at week 3 and took it away at week 6. Figure 5.31 depicts the MessyBoard usage and events for group E.

Only two members of group E reported significant MessyBoard use, though all reported seeing it often when it was projected on the wall. The leader added some content, but the vast majority was added by another member. Overall, this group used MessyBoard far less than other groups as shown in Figure 5.29.

Early on, group E's leader posted lists of tasks similar to group D. As time went on, a second group member started posting screen shots with accompanying notes as shown in

Figure 5.34. This member reported in an interview that some of the content was related to other members' parts of the project and her intent was to stimulate discussion, but other members did not participate. He reported that he had decided to stop using MessyBoard even before we removed the projector from the room.

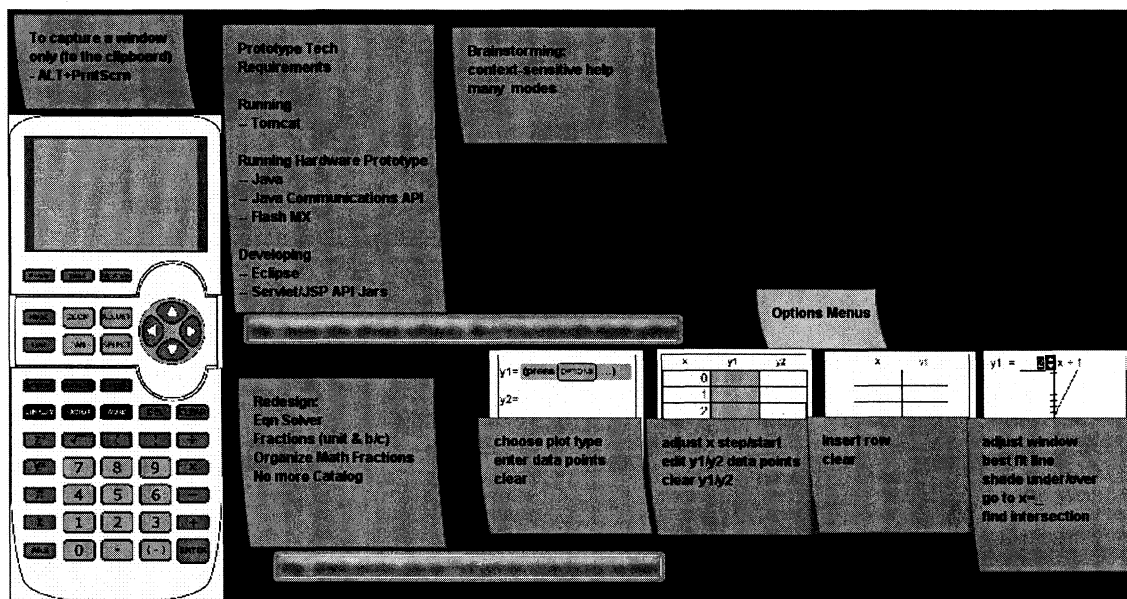


Figure 5.34. Group E's MessyBoard space.

Most group E members expressed a desire to post visual materials such as design mock-ups and storyboards on MessyBoard, but they complained that MessyBoard did not provide enough space for several large images and it did not support interactive Flash prototypes. One of them commented that it was easier to have a team member walk over and look at her computer screen.

We observed that group E had already established methods for communicating and coordinating. They used a Wiki (Cunningham, 2005) to share files and they kept track of their goals and deadlines using sticky notes on a large piece of paper taped to the wall. They posted design prototypes and other project-related materials on the wall as well.

Group F designed an enhanced corporate e-mail system. None of the group members were familiar with MessyBoard. I set up a projector in this group's space after they had been using MessyBoard for 3 weeks. This group had a project manager, but other mem-

bers did not describe this person as a leader. The project manager seemed neutral toward MessyBoard, and another member of the group was quite enthusiastic.

Group F used MessyBoard primarily to coordinate collocated synchronous activity. Specifically, they used it to create lists of bugs in their prototype and they manipulated the lists as they worked to keep track of who was fixing each bug and which problems had so far been fixed, as shown in Figure 5.35. They used it in a similar manner to fix problems on their web site describing the project. In all, there were three instances of this kind of use, each spanning less than a day, with very little use in between as shown in Figure 5.32. Each instance occurred immediately before a deadline. All members reported participating in these sessions, though the project manager recalls using MessyBoard more than other members.

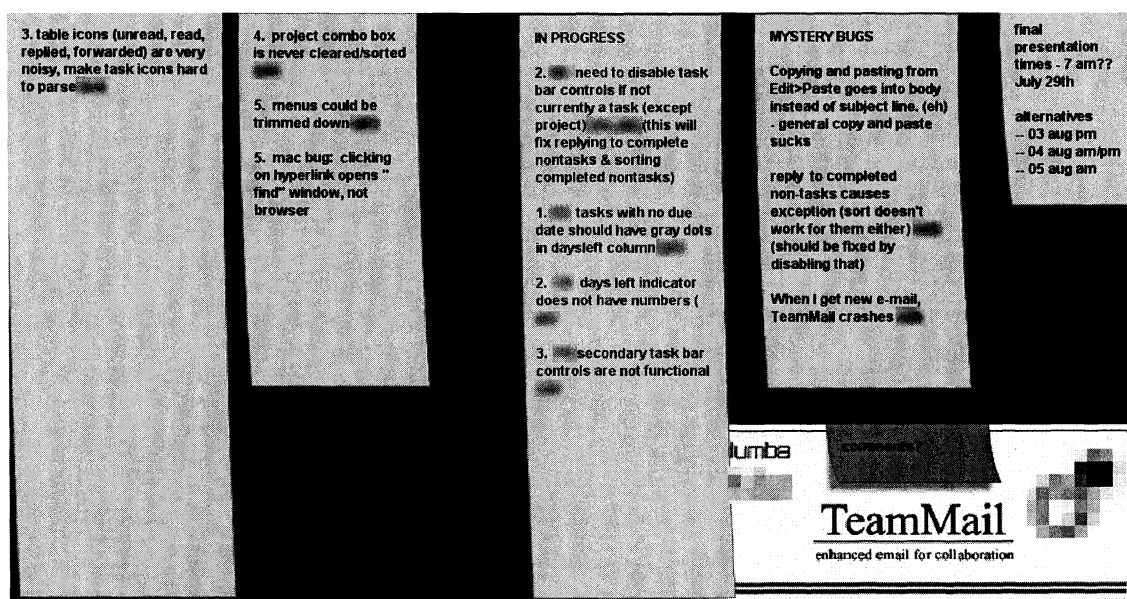


Figure 5.35. Group F's MessyBoard space.

Members of group F reported that they did not have the projector on very often since it was noisy and dim. They decided on their own to set up a monitor as a dedicated MessyBoard display.

Members of group F reported that they liked using MessyBoard to keep track of bugs because they could all view it from their own workstations. They had previously used the

whiteboard for real-time bug tracking, but they preferred MessyBoard because they could all modify it without getting out of their chairs and walking across the room.

5.5.2.2.2 Course Beta

Course Beta comprised undergraduate students at Carnegie Mellon working on semester-long projects in 5 groups of four Carnegie Mellon students each. In addition, each group included a foreign student working from a university in The Netherlands for a total of five members per group. For this reason, the course instructor was quite enthusiastic about having the students use MessyBoard to collaborate and I was granted half of an entire lecture to demonstrate MessyBoard and answer questions. This course was taught by the design department and all of the students had a design background.

I set up a MessyBoard space for each individual group and an additional space for the entire course to use. Though different groups used their MessyBoard spaces in varying amounts, there is a clear overall trend: students in course Beta used MessyBoard for a brief period early on in the semester and then their use quickly tapered off. Figure 5.36 illustrates this trend in all of the group MessyBoard spaces and in the space for the entire course. For two of the group MessyBoard spaces and the space for the entire course, there was also a small spurt of activity in November.

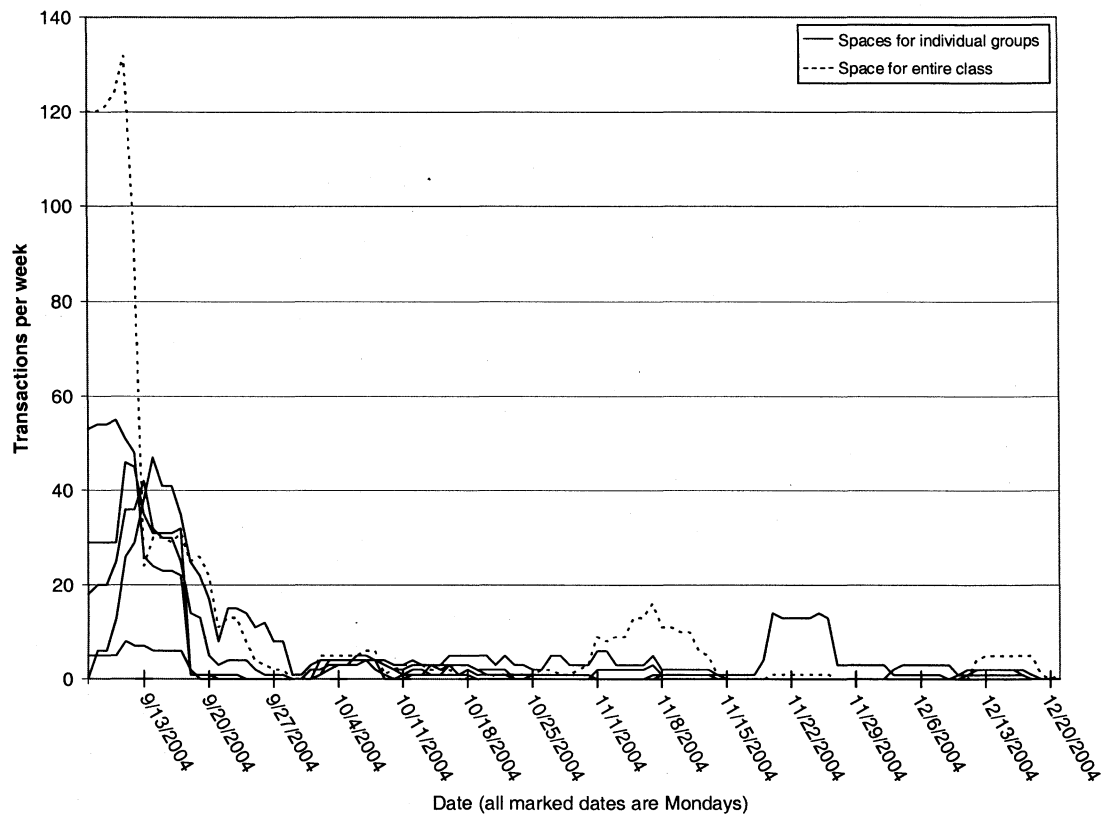


Figure 5.36. MessyBoard usage for students in course Beta.

The early use for these groups, and on the board for the entire course, is dominated by introductions and exchanges of contact information. Students used MessyBoard to exchange pictures of themselves, e-mail addresses and user names for chat programs, as shown in Figure 5.37. One group used MessyBoard early on to create an affinity diagram, as shown in Figure 5.38.

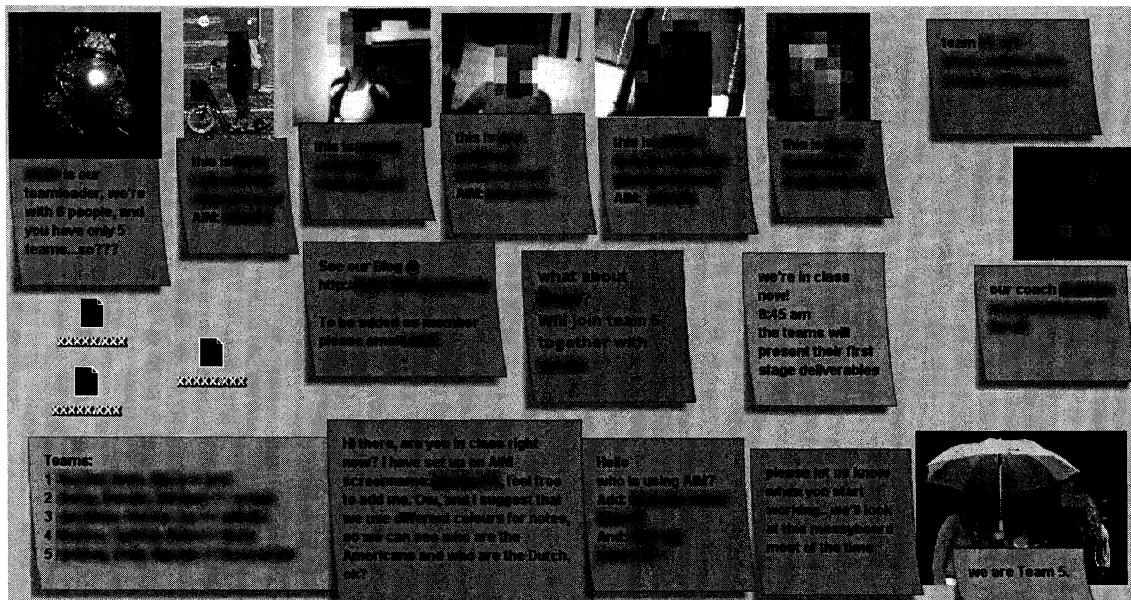


Figure 5.37. Students in Course Beta use MessyBoard to introduce themselves and to share contact information.

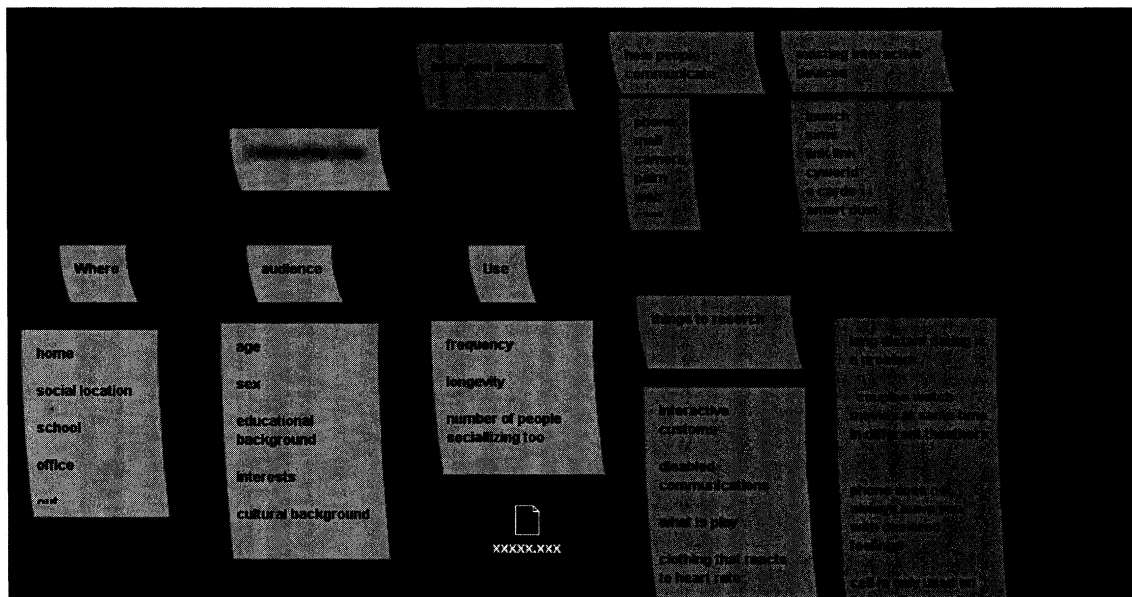


Figure 5.38. One of the groups in Course Beta uses MessyBoard to create an affinity diagram.

Ten of the students in course Beta completed the survey and responses to an open-ended question confirmed that they used MessyBoard primarily for sharing files and for ex-

changing contact information, and that the files were presentations and other deliverables for the course project. Out of all the collaboration activities listed in the survey, they rated file-sharing as the most important with an average rating of 6.2 on a seven point scale. The only other activity rated higher than 4.0 on average was brainstorming, and responses indicated that MessyBoard was not used often for this activity.

5.5.2.2.3 Courses Gamma, Delta and Epsilon

Course Gamma was a project course for a professional Masters Degree program and course Delta was a graduate course intended for Masters and Ph.D. students. Both courses were highly technical and mostly drew students with Computer Science and Engineering backgrounds.

The instructors for course Gamma were generally unenthusiastic about MessyBoard, allowing me to demonstrate it to a small audience of students who showed up early in the morning before the lecture was scheduled to begin. The instructor for course Delta was more accommodating and granted me a brief demonstration during the scheduled course period, similar to course Alpha.

After giving the students a demonstration, I created a MessyBoard space for each project group based on names and group descriptions that I received from the instructors. I sent out an e-mail announcement to each group with the URL for their MessyBoard space.

None of the groups in either of these courses used MessyBoard. Few students even looked at their MessyBoard spaces, and those that took the time did little more than to briefly experiment with the features. Only one of the students responded to the survey, so it is difficult to know why these groups did not use MessyBoard.

Course Epsilon taught members of Group C about the fundamentals of film-making. Students were divided into eleven groups of four students each and the groups worked together over the course of the entire semester. I created MessyBoard spaces for each of the groups and one for the entire class. Though the space for the entire class was used little and abandoned early on, most of the project groups used their MessyBoard spaces on at

least one occasion. The dominant use was sharing files, with files spatially grouped and labeled with notes, similar to group D in course Beta.

Since students in course Epsilon did not use their MessyBoard spaces extensively and most of them received surveys regarding their membership in other MessyBoard groups, I did not administer a survey asking about use of these MessyBoard spaces.

5.5.2.2.4 Other Student Project Groups

The two groups discussed in this section were made up of students working on collaborative projects. One group had two members and the other had three. In contrast to the small groups mentioned above, I made no explicit effort to recruit these groups. The groups approached me about using MessyBoard because they were already familiar with it. Despite their apparent motivation and interest, neither of these groups used MessyBoard much at all. As shown in Figure 5.39, both groups had brief spurts of activity lasting about a month and then usage stopped entirely. (By coincidence, both groups started using MessyBoard within a few days of each other.)

Both students in the two-member group responded to the survey and indicated that they used MessyBoard primarily to share files.

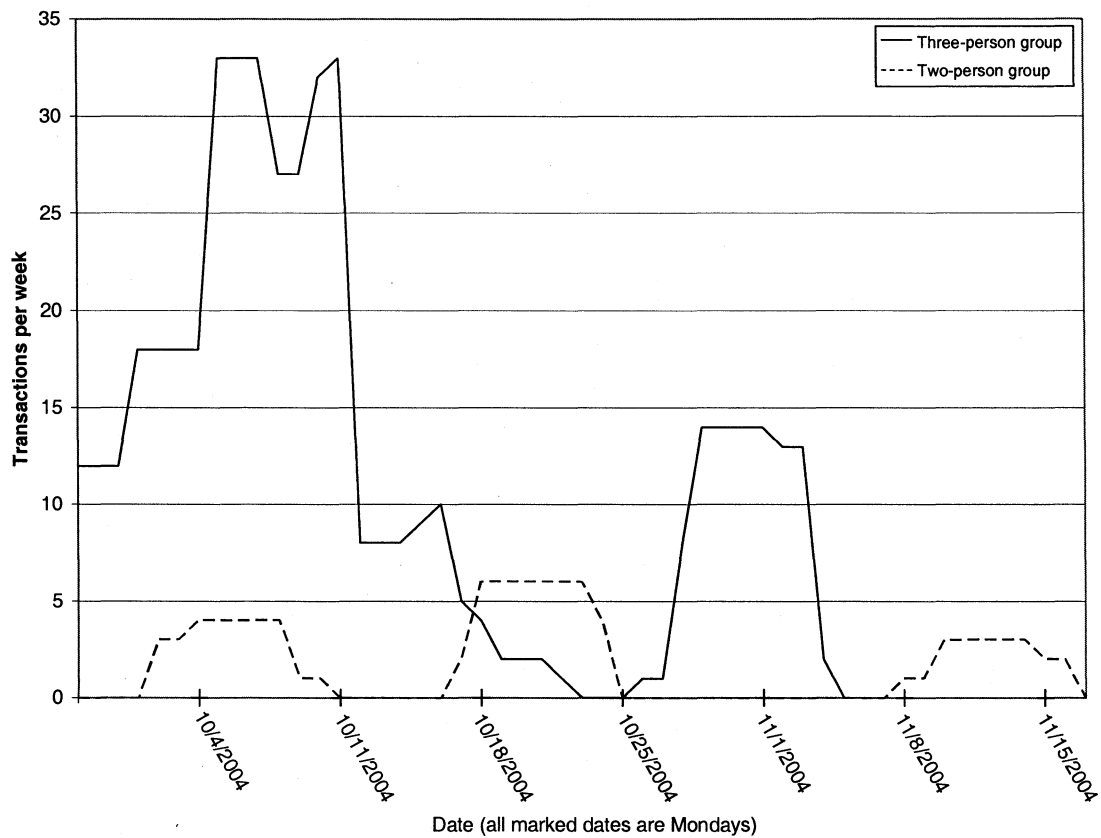


Figure 5.39. MessyBoard usage for two project groups that approached me and requested MessyBoard spaces.

5.5.2.3 Research Group Observations

I recruited 3 small research groups to use MessyBoard, each comprising students, staff and faculty members at Carnegie Mellon. I refer to these groups as G, H and I.

Group G is made up of graduate students, undergraduates and staff members (10 people in total) who work for a single professor. The group has a shared lab where some of the undergraduates and staff members work, but I did not install a projector there. I demonstrated MessyBoard to the group at one of their meetings and then met with the graduate students and the professor individually in order to teach them how to use MessyBoard and to install the MessyBoard screen saver. A graduate student reportedly installed the MessyBoard screen saver on one of the computers in the shared lab. I was unable to in-

interview all of the members of this group, but I did talk briefly to the professor and we interviewed two of the graduate students.

Group G's overall use of MessyBoard was low as shown in Figure 5.40, despite the fact that several of them installed the screen saver, including the group leader. (Due to a logging error, the exact number of users who installed the screen saver is unknown.) Early on, the group leader posted a note expressing doubt about MessyBoard's usefulness. Later, he posted a calendar and asked group members to mark the days that they would be on vacation using the freehand drawing tool, as shown in Figure 5.41. Most group members were slow to respond, and MessyBoard was used very little for the rest of the summer.

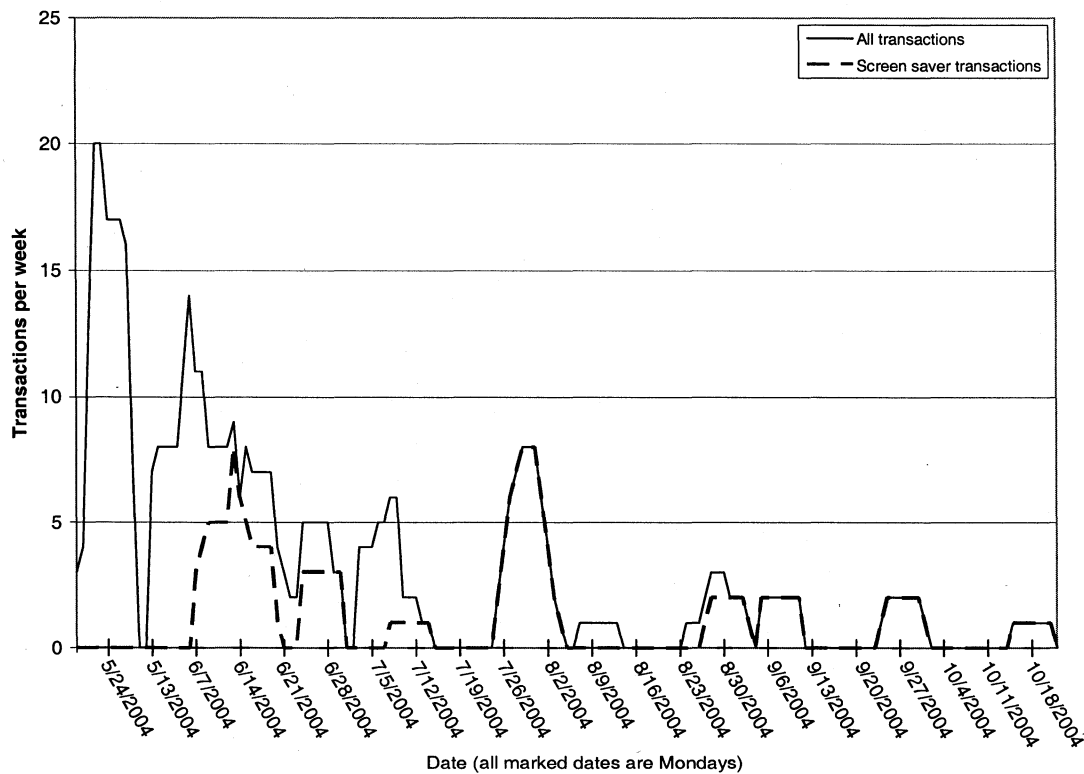


Figure 5.40. MessyBoard usage for group G.

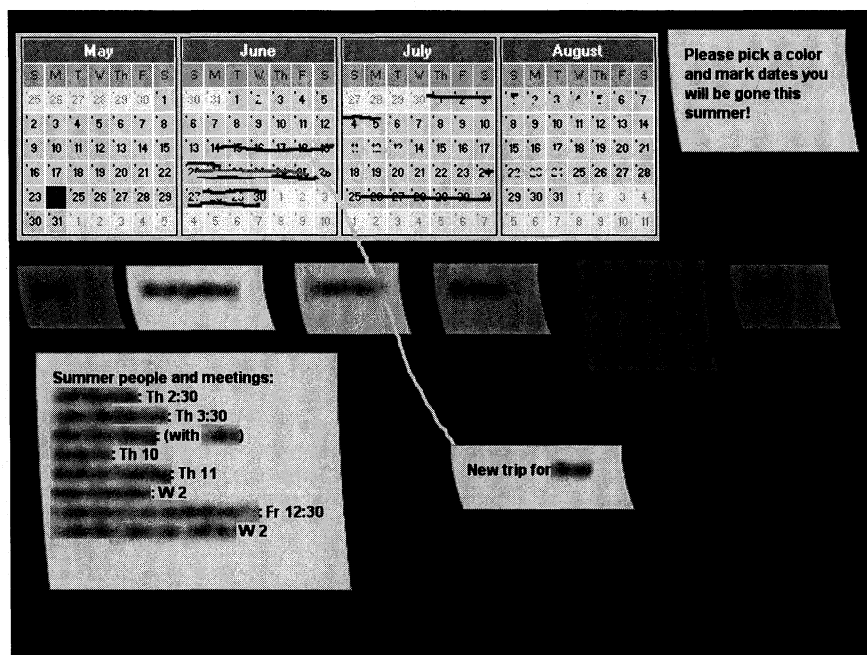


Figure 5.41. Group G uses MessyBoard to build a shared representation of the dates that they will be out of town.

When asked why group G had not used MessyBoard, the leader's opinion was that they could not think of anything to do with it. He added that the vacation calendar was not useful and he did not refer to it. Group members discussed using MessyBoard to schedule use of their shared lab for experiments, but nobody took any action. It is not clear whether or not some of the undergraduates ever looked at MessyBoard at all.

Group H is made up of faculty and undergraduates (seven people in total). One of the professors is the leader, and they collaborate on the design aspects of a larger interdisciplinary research project. Two of the faculty members share an office and the rest of them are dispersed over two buildings on the same campus. I demonstrated MessyBoard to the entire group at a meeting and sent out an e-mail announcement. I met with the leader individually to show him how to run MessyBoard, but I did not do this for other group members. I did not set up a projector in any of their offices. I interviewed the leader of this group individually and I spoke to two of the other members in a joint interview.

Group H's overall use of MessyBoard was low, as shown in Figure 5.42, but it was apparently of some value to them. They used it to share files and a few lists, and the two faculty members said that it was nice to be able to quickly put a file on MessyBoard and then verbally tell others to go get it. The group leader said that he only used it a little because he was not a daily participant in the project. Qualitatively, this group's behavior is similar to that of student group D, though they did not use it nearly as much.

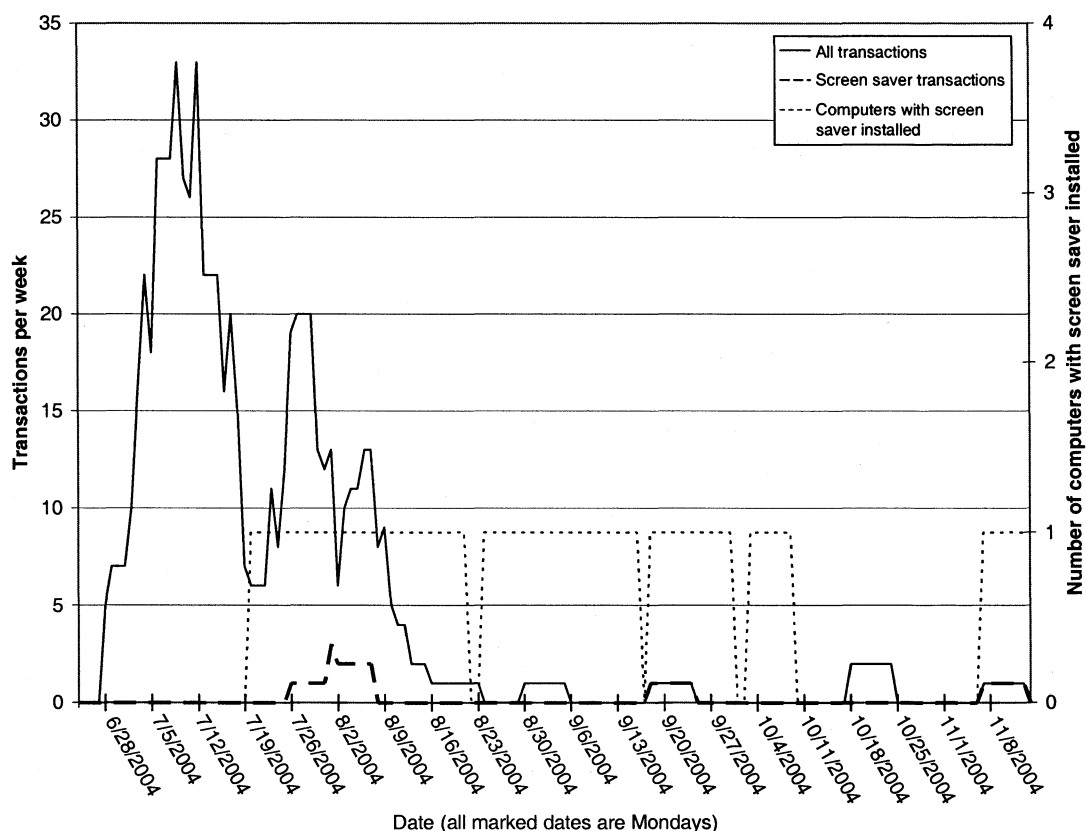


Figure 5.42. MessyBoard usage for group H.

Group I is made up of graduate students and staff lead by a professor (six people in total). As shown in Figure 5.43, they used MessyBoard very little. Besides some initial discussions about features in MessyBoard, members of group I use their MessyBoard space mainly for brief announcements. I did not interview members of this group, but responses to the open-ended survey item confirmed that they used their MessyBoard space as a

general-purpose bulletin board for announcements and meeting reminders and little of the content was closely related to their work.

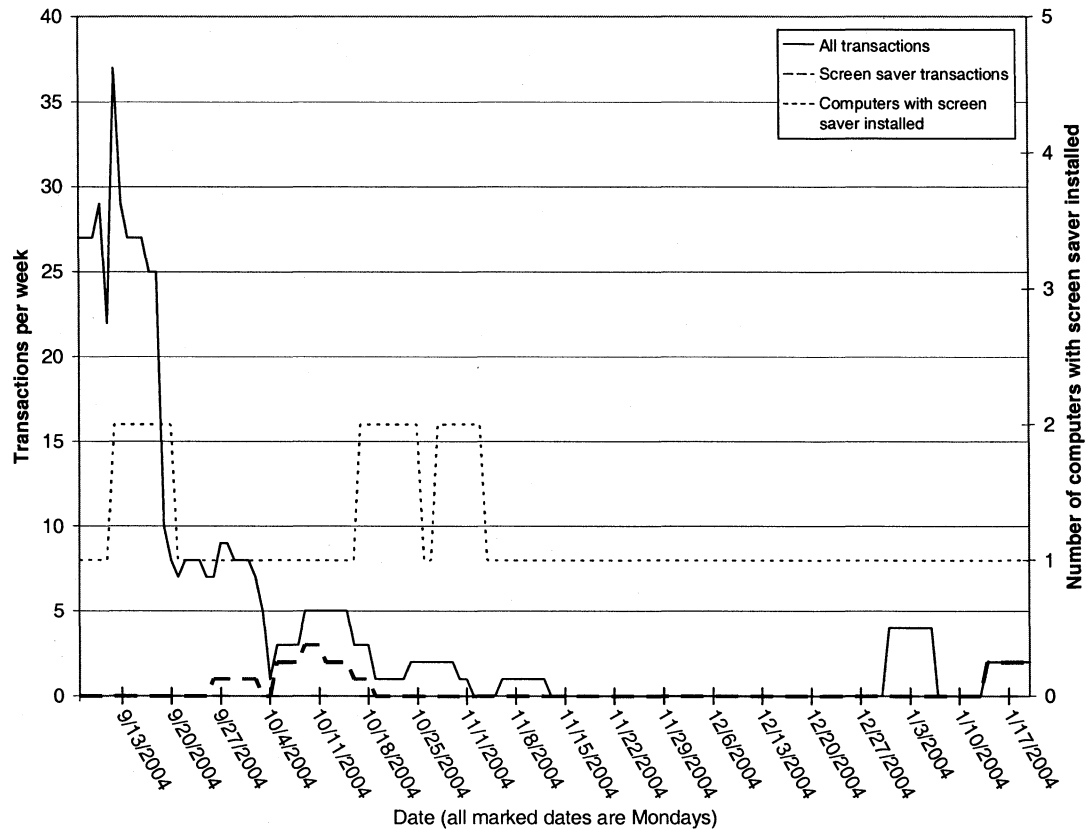


Figure 5.43. MessyBoard usage for group I.

5.5.2.4 Discussion of Small Group Observations

Figure 5.44 shows the amounts of activity for all of the small groups described in this section. Clearly, only group D used MessyBoard regularly over an extended period of time. Group F used MessyBoard a great deal in several short spurts of activity.

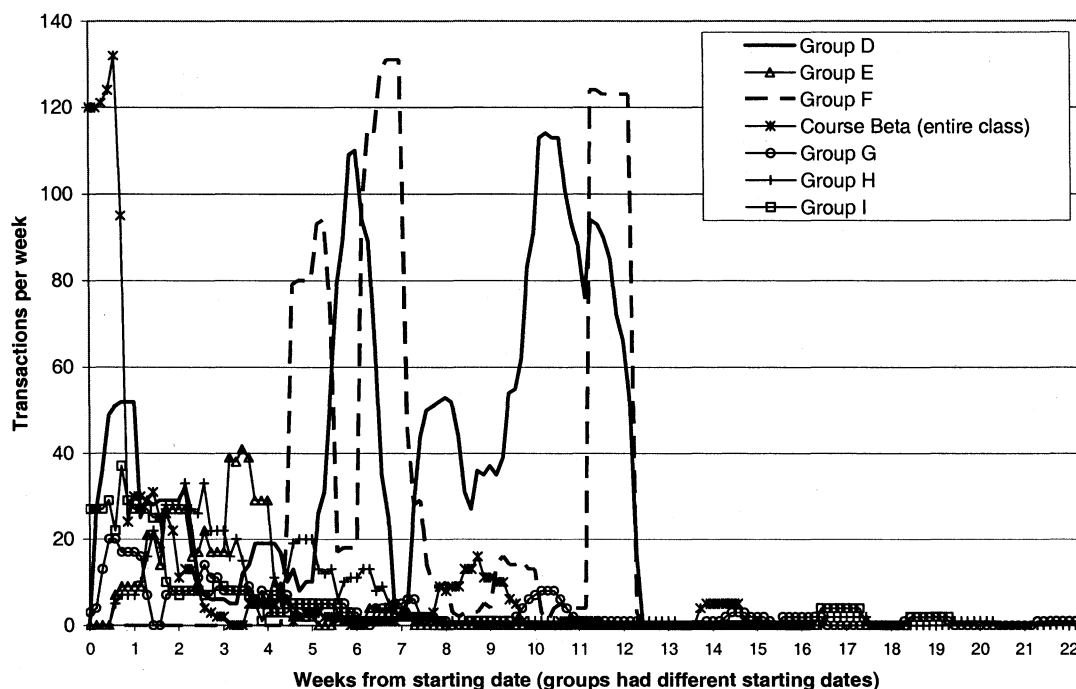


Figure 5.44. MessyBoard usage for the small groups that used it the most.

The high amounts of usage for groups D and F may be explained by a combination of two factors: *group focus* on shared goals and the existence of a *critical task* that MessyBoard performs better than existing tools. Groups in course Alpha were more focused than any of the other small groups here, since they all worked on the same project to the exclusion of all else. Though some of the faculty and staff members in the research groups may have also worked full time, survey results indicate that even in these groups the group members are working on different and sometimes unrelated tasks.

The existence of a critical task to which MessyBoard is well-suited may explain why Groups D and F used MessyBoard more than group E, which was also highly focused, and why the nature of use differs so much between groups D and F. Members of group D indicated in interviews that sharing files was extremely important and that they were dissatisfied with their FTP server because they had difficulty finding the most current files. MessyBoard gave them an easy way to share files, group them spatially and annotate them so that it was always clear which files were in the current working set.

Prior to the introduction of MessyBoard, group F had already established a pattern of working together intensely to fix bugs in the hours before a deadline. They reported in interviews that they had previously used the whiteboard in their shared office to keep track of outstanding bugs and to keep track of who was working on what. They found MessyBoard to be a superior solution because they could all view it on their own screens and they could coordinate without getting up from their seats and walking to the whiteboard.

Group E made heavy use of paper artifacts and their shared office was decorated with story boards, lists and a large timeline covered with sticky notes. In interviews, members reported that they had a need to share storyboards, design mock-ups and interactive prototypes written in Flash. MessyBoard does not display interactive Flash content and the size and resolution were too limited for the kinds of artifacts that this team wished to share.

5.6 Other Observations of MessyBoard Use

Though MessyBoard was designed to support communication and coordination in stable groups, the versatility of the medium opens a wider set of possibilities. In this section, I discuss different kinds of uses for MessyBoard that have been tried and my observations.

5.6.1 Broadcast from Few to Many

A particular department at Carnegie Mellon has a strong need for coordination and sharing of information since they share a facility and a set of equipment that they all use for different purposes throughout the semester. I refer to this department as group G. A staff member in the department approached me after hearing about MessyBoard through a member of another group that was using it.

Members of group G had traditionally used a physical bulletin board to coordinate their activities and they were struggling with the integration of new digital media. They wanted something that was remotely accessible but still physically present in the environment.

With my help, they set up a digital projector in their facility and projected MessyBoard in a prominent location. A small group of students, staff and faculty responsible for coordinating the activities of the department were given access to the MessyBoard space, which was protected by a password so that the entire department could not access it. The students posted timely announcements on the MessyBoard space so that passers-by would see it on the projection screen.

I briefly interviewed the small group of students who had access to the MessyBoard space near the end of the semester. They reported that they liked the idea, but they were still in the midst of figuring out what information should go on it and how many people were actually paying attention to it. They were still struggling to find the best way to integrate MessyBoard, their web site and their physical bulletin board in order to disseminate information.

5.6.2 Place-based Use

Group C contained a sub-community of students who were interested in recording and editing sound. They had set up a recording studio as a shared resource for all of the students in the department. After speaking with one of the students, I set up a MessyBoard space specifically for the studio and announced it using an e-mail distribution list specifically for students who were interested in sound recording. I installed the MessyBoard screen saver on two shared computers in the sound studio.

This MessyBoard space was used very little and it is was of questionable value to the students. A few students used the board to post questions or requests for help related to the sound equipment, but only one answer was ever posted. A request for a particular sound effect did not prompt anyone to share a recorded sound file or link.

5.6.3 Administrative Assistants

Administrative assistants (referred to as *admins*) at Carnegie Mellon assist the faculty and students by performing tasks such as filing receipts for reimbursement, making travel arrangements, scheduling the use of rooms and planning events. Though admins often do not work in stable groups, they communicate frequently with a small set of people includ-

ing their assigned faculty members and other admins in their department. I hypothesized that they might find MessyBoard useful for ad-hoc collaborations on transient tasks and for sharing documents and other files.

In most cases, I contacted administrative assistants in two departments by visiting their offices un-announced. I demonstrated MessyBoard and asked if they would be interested in using it. One additional admin contacted me after attending one of my talks. In all cases, if the admin was interested, I set up a MessyBoard space and then made a follow-up visit in order to make sure that they could use it on their own computer.

I chose to set up a MessyBoard space for each individual admin because it was not clear with whom they would use it to communicate. I hypothesized that each admin might start by using MessyBoard as a personal organizational tool and when an admin with her own MessyBoard was assigned a collaborative task, she might spontaneously decide to use it and invite others to post material on it. There were two exceptions to this pattern. Two pairs of admins worked in a shared office and needed to communicate with each other, and in those cases I created a single MessyBoard space for the pair to share.

I encouraged the admins to try the MessyBoard screen saver and seven out of eight agreed to have it installed on their workstations.

After several weeks, I conducted interviews with four of the admins. Interviews were brief and informal because: 1) the admins did not use MessyBoard very much, and the main point of the interviews was to find out why, and 2) admins are busy and scheduling large blocks of time was too difficult. Only one of the admins continued to use MessyBoard for any length of time, and she used it on only a few occasions to transfer files between computers.

The admins offered a variety of reasons for why they did not use MessyBoard, and it is difficult to draw any general conclusion. One salient hypothesis is that they might have been less technologically savvy than other users, but I took great caution to mitigate this factor when I introduced MessyBoard to these users. I met with them each individually and I personally installed the latest version of Java on their computers. I created shortcuts

on their desktops that would launch the stand-alone application and I made sure that they knew how to launch MessyBoard and use the basic features.

5.6.4 Academic Conferences

The design of MessyBoard is based on the assumption that each space will be used by members of a small group who know and trust one another. The finite space allows a limited amount of information, making the use of a single space by a large organization impractical. Free access and the lack of per-user permissions make it unlikely that a group of strangers would use it to store important information.

However, the features of MessyBoard could also be well-suited to a different kind of group. The group of attendees at a small or mid-sized academic conference is a comparatively large group (usually at least 100 people). However, for the time that they are at the conference, they are unusually focused on a small number of common activities and events. They are together for a short amount of time, so they may not generate an overwhelming amount of important information. Though they do not work together as a single group, they do have a dense network of collaborations and personal friendships that engenders norms of politeness and trust.

Similar gatherings are common outside of academia, such as management seminars, trade shows and conventions. These events serve as important ways to build social networks, exchange information and forge new collaborations. The cost of these events is extremely high, since participants must take time off from their usual routines and they must often travel and stay in hotels. Thus, if an electronic communication tool such as MessyBoard helped attendees make the most of their short time at a conference, it would provide enormous value to workers in many domains and professions.

An electronic communication medium might also serve to bridge the separation between conference attendees and interested people in other places, allowing those who cannot attend the conference to share in some of the benefits. Streaming video, webcasting and blogging are becoming increasingly common at live events, along with chat rooms that are open to both attendees and the general public. While video and chat allow remote par-

ticipants to participate in the events of the moment, perhaps MessyBoard could be an efficient way for a casual remote participant to "drop in" on a conference for a few seconds and get a quick summary of what is going on.

I hypothesized that MessyBoard users at a conference might use MessyBoard for announcements such as the locations or times of conference events, for posting material relevant to the conference presentations, or for posting humorous material. However, it was not clear that users would use MessyBoard at all in such a setting due to intimidation or time constraints. Alternatively, perhaps the short duration of the conference would mean that usage patterns would be dominated by the novelty of the medium and useful conventions would not have enough time to evolve:

5.6.4.1 Methods

I deployed MessyBoard at two academic conferences. User Interface Software and Technology (UIST) 2004 took place in Santa Fe, New Mexico with 186 registered attendees. Computer Supported Cooperative Work (CSCW) 2004 was held in Chicago, Illinois and 471 people attended.

At both conferences I set up a single MessyBoard space with the name of the conference. Both conferences were held in hotels. UIST is a single track conference and most of the presentations took place in a single room. CSCW had three simultaneous tracks throughout the conference and presentations took place in three adjacent rooms with a common lobby area. In both cases, I set up a projector and a screen to display MessyBoard all day in a highly visible area close to the presentation rooms as shown in Figure 5.45.



Figure 5.45. MessyBoard projected opposite the registration table at UIST 2004.

Both conferences provided wireless Internet access to attendees who brought their own laptop computers with wireless adapters. I allowed and encouraged attendees at both conferences to access MessyBoard using their own computers. However, the wireless network was problematic at both conferences. At UIST the signal strength was weak in the presentation room, resulting in extremely long download times for the MessyBoard applet. At CSCW network access was intermittent and the bandwidth was highly dependant on the actions of other attendees.

For both conferences, I displayed the conference MessyBoard space at Carnegie Mellon in the shared workspace for group B (discussed in Sections 5.5.1.1.2 and 5.5.1.1.3). Since this group already had their own MessyBoard space, I configured the client software to switch back and forth between the conference space and the group's own space with a smooth fading transition. I announced this change to group B by e-mail before the UIST conference and arranged for a group member to make sure that the projector was on during the day. I also gave them the URL for the UIST MessyBoard space and invited them to contribute. Due to an oversight on my part, I did not announce the change before CSCW and it is unclear how often the projector at Carnegie Mellon was on during this time. (I was normally responsible for turning the projector on and off.)

At both conferences, I presented MessyBoard as a public demonstration at a refereed demonstration reception. In both cases, the demonstration took place on the evening of the first day of the conference. I demonstrated MessyBoard to interested attendees and told them how to access the conference MessyBoard space. I also handed out business cards with the MessyBoard main page URL. For the duration of each conference, the first item on the main MessyBoard web page was a link to the conference's MessyBoard space.

Due to differences between the two conferences and lessons learned from my experience at UIST, I did some things differently at CSCW. At UIST I placed the projector in the registration area. This was a high traffic area since attendants had to go there at least to register and food and coffee was placed nearby during breaks. However, food and coffee were also available in other locations and the registration area was not immediately visible to attendees exiting the presentation room. In short, MessyBoard was visible but it was not unavoidable.

The laptop computer that ran the MessyBoard projector was accessible to the UIST student volunteers seated at the registration desk, but it was not available for use by attendees. Attendees needed to use their own laptop computers in order to modify MessyBoard.

I placed some content on the UIST MessyBoard throughout the conference. Initially I placed a welcome message and some instructions for accessing MessyBoard along with the URL for the conference MessyBoard space. On the first day I posted pictures from research papers that were presented at the conference, and each day I posted a short daily schedule. I deleted only my own postings but I did not delete anyone else's.

The CSCW organizers were more enthusiastic about MessyBoard and one of them was already posting relevant information on it before the conference began. I personally posted less material on the MessyBoard space. I initially posted a conference logo with a welcome message and instructions, but I did not post any material related to the conference presentations. I made bitmap schedules in advance for each day and posted them on

MessyBoard daily. (The multi-track schedule for CSCW demanded a table format that would have taken up too much space if I posted it in a note as I did for UIST.)

The layout of the hotel conference space at CSCW was quite favorable compared to UIST. Each presentation room had a single exit into a shared lobby area. I was able to place the MessyBoard projection screen such that it was visible to all participants immediately upon exiting the presentation rooms. I placed a second laptop on a table near the projection screen to serve as a public MessyBoard terminal, since several UIST attendees expressed disappointment with the lack of a public MessyBoard terminal.

The CSCW organizers set up three IRC channels to serve as "back-channel" communication for the three paper sessions. I created three separate MessyBoard spaces for this same purpose named `cscwA`, `cscwB` and `cscwC`. These were separate from the main conference space, which was simply named `cscw`. Thus, there were 4 separate MessyBoard spaces for this conference. Only the main conference space was shown on the projector. In contrast to UIST, the CSCW organizers published the URLs for all of the conference MessyBoard spaces in the printed conference materials that were handed out to all attendees.

5.6.4.2 Observations

Figure 5.46 and Figure 5.47 show the MessyBoard spaces as they were being used at the UIST and CSCW conferences. In general, a few of the conference attendees used MessyBoard to post announcements, share humorous material, or simply to experiment with the medium. MessyBoard was not used at either conference to coordinate outings or conference events. Student volunteers may have used MessyBoard most often, especially at UIST where I set up the projector and a laptop computer at the student volunteer desk. Participation from the students in group B at Carnegie Mellon was minimal. At the conference, many attendees told me that they did not understand that they could access MessyBoard from their own laptop computers or that they had tried to access it and the wireless network was not working properly.

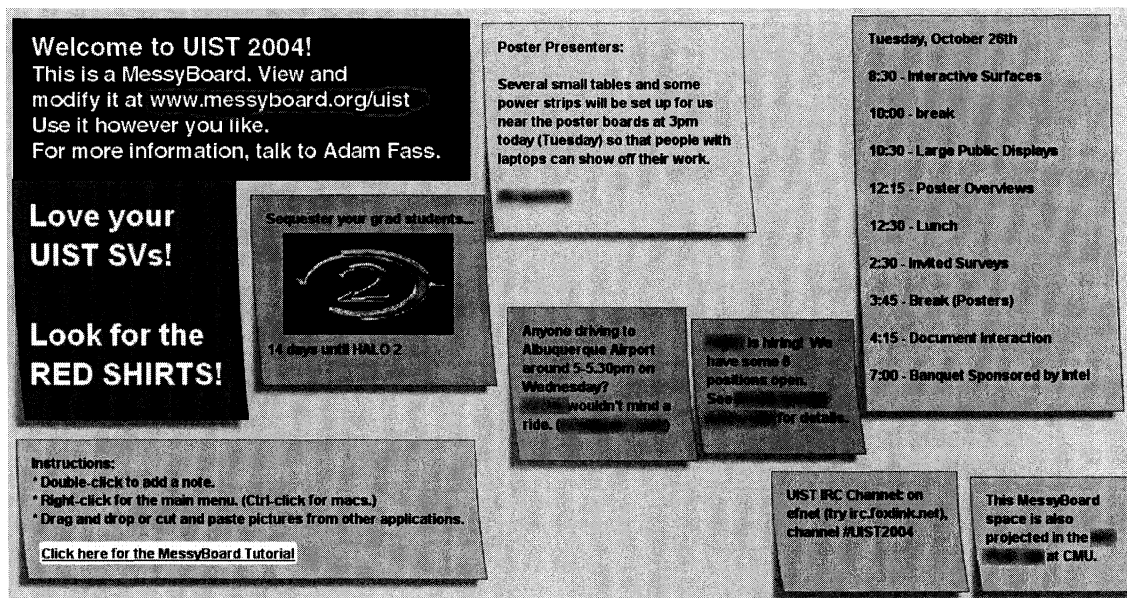


Figure 5.46. The UIST 2004 MessyBoard space.

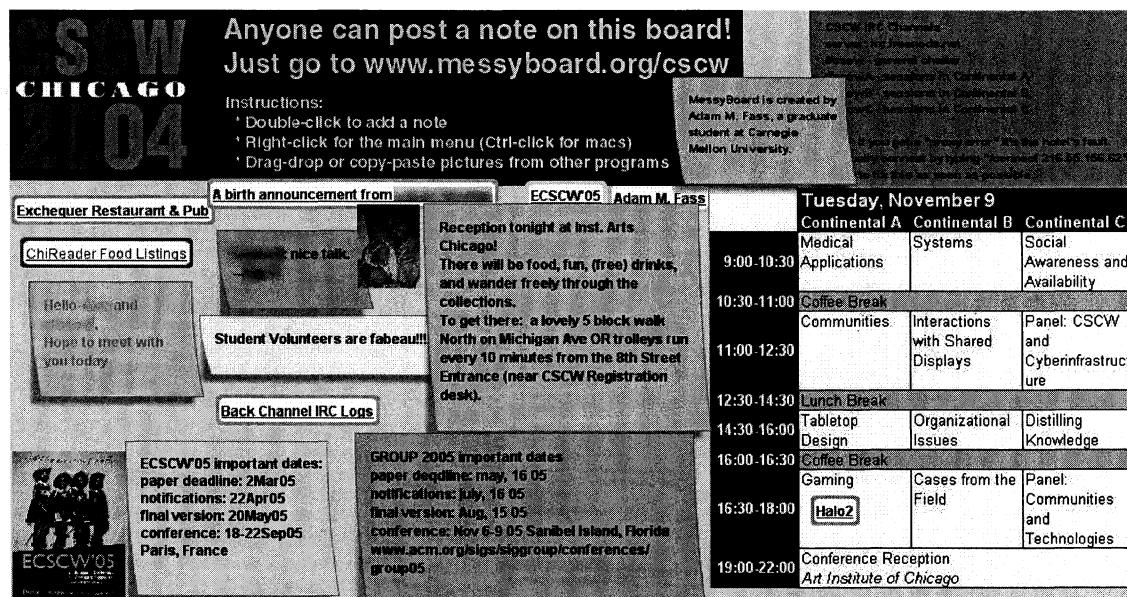


Figure 5.47. The CSCW 2004 MessyBoard space.

As shown in Figure 5.48, the CSCW MessyBoard was used more than the UIST MessyBoard space. A number of factors may have caused this. CSCW is a larger conference and the MessyBoard projection screen at CSCW was positioned such that people would see it more often than the projection screen at UIST. One of the student organizers for

CSCW was enthusiastic about MessyBoard and had already posted some announcements before the conference started. Another difference is that I deleted content from the CSCW MessyBoard every night, making room for new content the following day. In both cases, usage started high on the first day and declined throughout the conference.

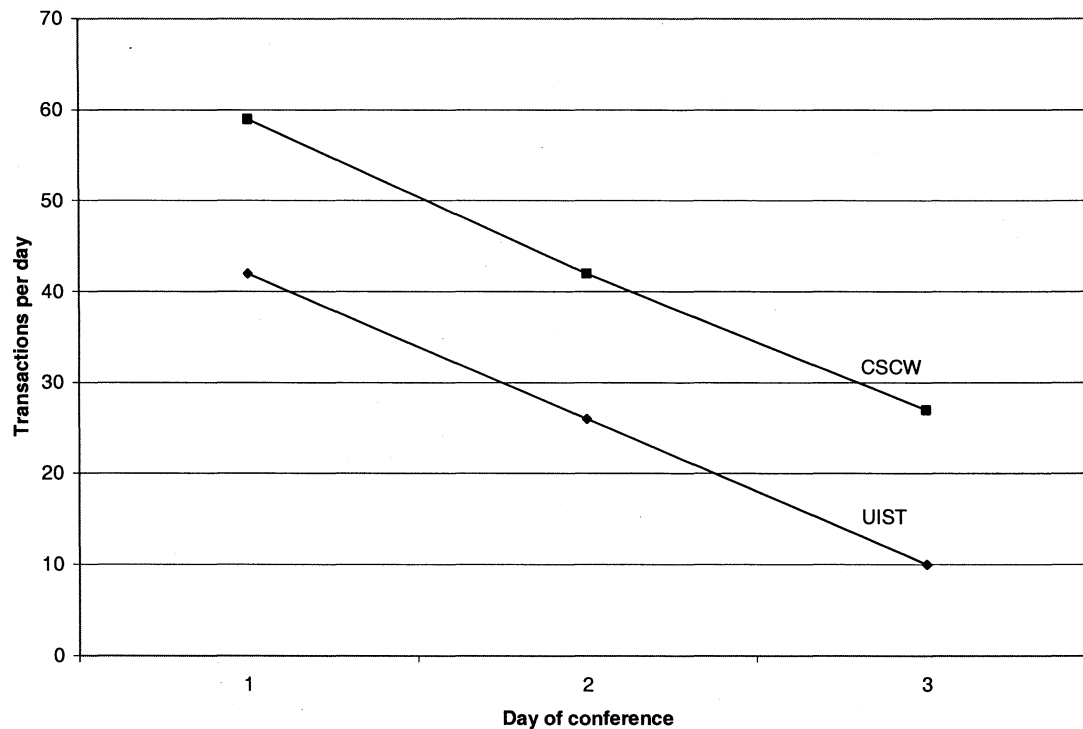


Figure 5.48. MessyBoard usage for the UIST and CSCW conferences.

5.6.5 Short-term Groups

First year students in group C (discussed in Sections 5.5.1.1.3 and 5.5.1.4) are required to take a project course in which they work in four-person teams for two weeks at a time creating interactive Virtual Reality worlds. After each two-week cycle, the groups are shuffled and they begin a new two-week project with a new team. There were thirteen groups for each cycle and five cycles, for a total of 65 groups, though the groups were always chosen from the same pool of 55 people. Since messyboard.org did not yet allow users to create their own MessyBoard spaces, I set up 15 MessyBoard spaces named for the course with the numbers 1 through 15 appended at the end. I announced the creation

of these spaces to the entire course and invited the students to use them however they pleased. As members of group C, the students were already familiar with the medium and they did not need to spend any of their (very) limited time learning how to use it.

Figure 5.49 shows how some of these groups used their MessyBoard spaces. 17 out of the 65 groups chose to use MessyBoard. Most of them used it to share files, artwork, code, to-do lists and humorous pictures of themselves. There were also a few instances of conversations and brainstorming.

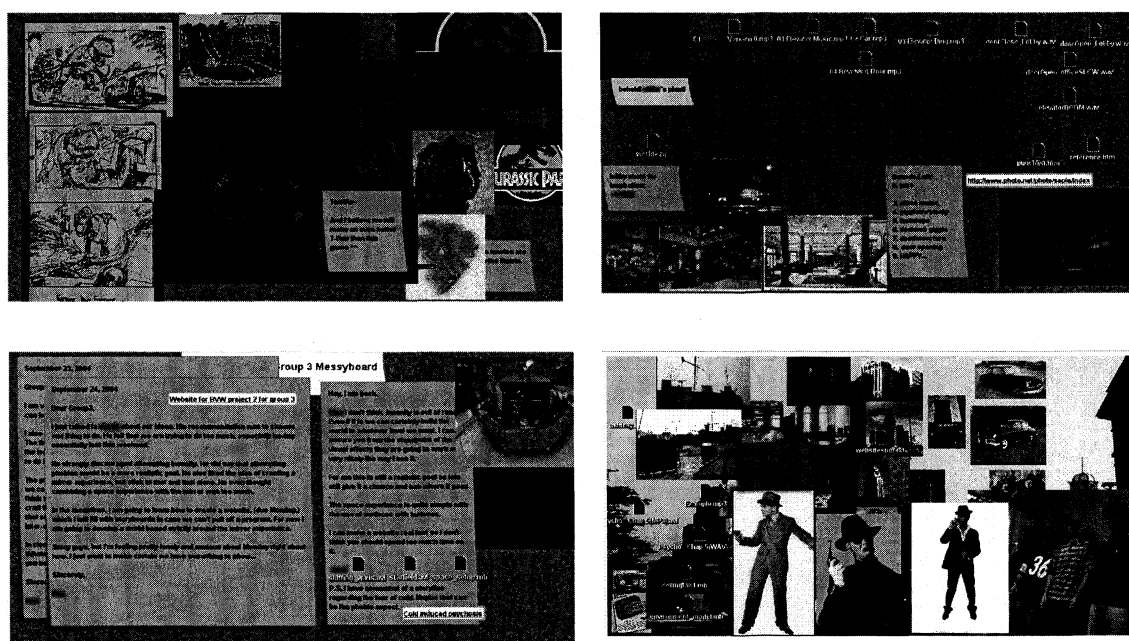


Figure 5.49. MessyBoard spaces used by class project groups working on short-term projects.

5.7 Internet Deployment

I have discussed MessyBoard assuming that it is used by groups of knowledge workers (Kidd, 1994). MessyBoard might be useful in other domains, such as families coordinating their schedules or groups of friends keeping in touch. However, it is not clear if these users will find value in MessyBoard, how they will use it, and how well the characteristics of the medium suit their needs.

Another motivation for a public MessyBoard deployment is as a proof-of-concept for a truly scalable monolithic MessyBoard server cluster that could serve the entire Internet. Such a server may be necessary for families, clubs and other groups of users that exist outside of an organization with a dedicated IT staff. It is not clear how the public would treat such a server given the unfettered access that is so important to reducing startup cost. Will they create a large number of spaces that they never use? Will they create many short-lived spaces for ad-hoc groups and projects, or will they create long-lived spaces for stable groups and projects? Answers to questions such as these are important for computing the cost of a monolithic MessyBoard server cluster and balancing it against potential revenues from advertising or subscription fees.

While it is impossible to prove that such a business venture could not succeed under any circumstances, the observations presented here suggest that it would be risky at best. MessyBoard was announced to many student organizations and clubs and it was advertised on a popular web search engine for users who searched for relevant terms. Despite the use of this targeted advertising, most of the MessyBoard spaces that were created were used very little. Several spaces were used in interesting ways, but their use is too diverse to suggest a core demographic of likely MessyBoard users.

5.7.1 Methods

In order to explore the use of MessyBoard by a diverse set of users, I created a publicly available MessyBoard server cluster as described in Sections 2.2 and 3.6, allowing any user to create their own MessyBoard space. The server went online on October 21st, 2004. In this analysis, I include events between that date and March 23rd, 2005.

I publicized the server at the UIST and CSCW conferences and I sent announcements to personal friends and students at Carnegie Mellon. Later, I sent an announcement to a student representative from every registered student organization at Carnegie Mellon University and the University of Pittsburgh. Finally, I purchased advertising space on the Web using Google AdWords (Google, 2005a). The AdWords program displays the MessyBoard advertisement, shown in Figure 5.50, when a user searches for certain hand-picked terms. Google also pays web publishers to display targeted advertisements, so the

MessyBoard advertisement is sometimes displayed on popular web sites like download.com. The total cost of the MessyBoard advertising campaign was \$51.08 with an average cost per click of around seven cents. I logged the number of page views of the main page at www.messyboard.org using a free counter called Nedstat basic (Nedstat, 2005) beginning January 2nd, 2005.

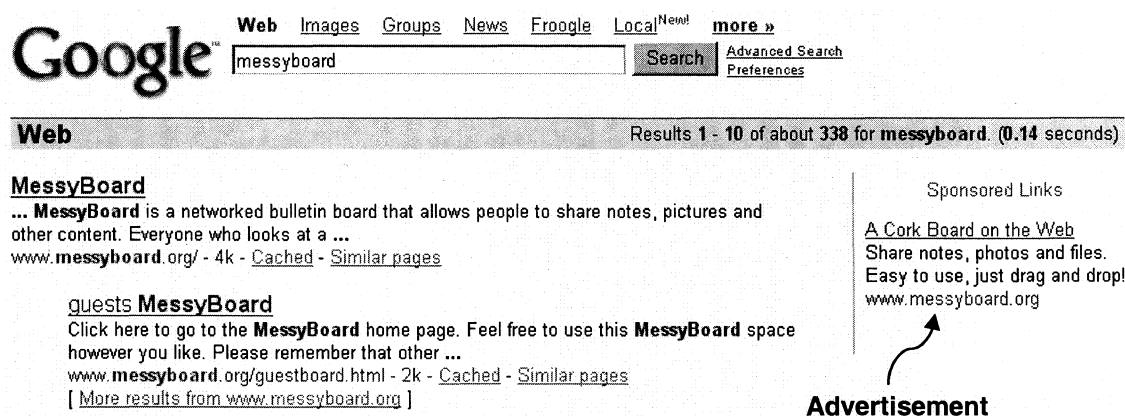


Figure 5.50. The Google AdWords program displays a simple MessyBoard advertisement on the Google search results page.

Figure 5.51 depicts the history of the MessyBoard public Internet deployment. Though it is not possible to discern how the different users found out about MessyBoard, the Google advertisements are also clearly responsible for driving a substantial amount of traffic to the web site. Google reports a total of 1,142 clicks on the advertisements during the period of analysis (135 From Google's search results and 1,007 from other web sites that display Google's advertisements) and Nedstat reports a total of 1,948 page views. The sharpest rise in MessyBoard space creation and the largest peak in page views for the front page of the main MessyBoard web site occurred when I sent out an announcement to all of the student organizations at Carnegie Mellon University and the University of

Pittsburgh.

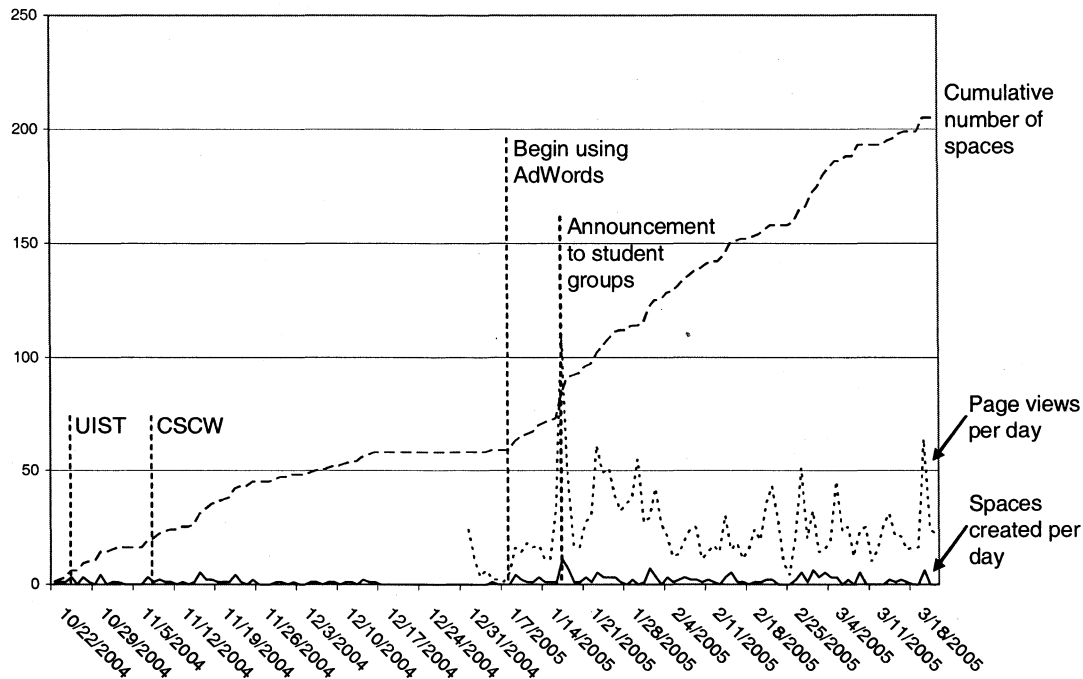


Figure 5.51. Number of MessyBoard spaces created each day, page views for the main MessyBoard web page per day, and cumulative number of MessyBoard spaces created for each day of the public Internet deployment.

During the period of analysis, a total of 205 MessyBoard spaces were created. Of these, I exclude the 71 spaces that were created in the last six weeks of the analysis period, since users of these spaces did not have time to settle into a long-term usage pattern. Of the remaining 134 spaces, I exclude another ten spaces for which the server never logged a connection from a client. The web browser is automatically directed to the MessyBoard space after a board is created, and the most likely reason for a space never being viewed is that the user did not have Java installed. Thus, a total of 124 spaces are considered in the following analysis.

Some of the MessyBoard spaces were created and used by members of the groups described in previous sections. These users may behave differently than other users since they are already familiar with MessyBoard and they may have received training and instructions from me. I estimate an upper bound on the number of spaces that I might have

influenced in this way by counting the number of spaces that logged at least one connection from an IP address at Carnegie Mellon University (in the 128.2 or 128.237 blocks). Of the 124 spaces, 48 logged at least one such connection. Thus, I am reasonably certain that at least 76 of the MessyBoard spaces considered here were created by people who were previously were unfamiliar with MessyBoard.

5.7.2 Observations

5.7.2.1 Why People Used MessyBoard

The Google AdWords program tracks the number of clicks on the MessyBoard advertisement as well as the terms that the user was searching for when that advertisement appeared, as shown in Figure 5.52. Over a third of the 131 clicks on the MessyBoard advertisements in Google's search results came from users who were searching for the terms "upload files". The next highest term was "my corkboard", the name of a related software product (MyCorkboard, 2005). The third highest term was "family pictures". It is unknown how many of these clicks actually resulted in people creating MessyBoard spaces, but these numbers suggest that a substantial portion of MessyBoard users were looking to use it as a way to store and/or share files.

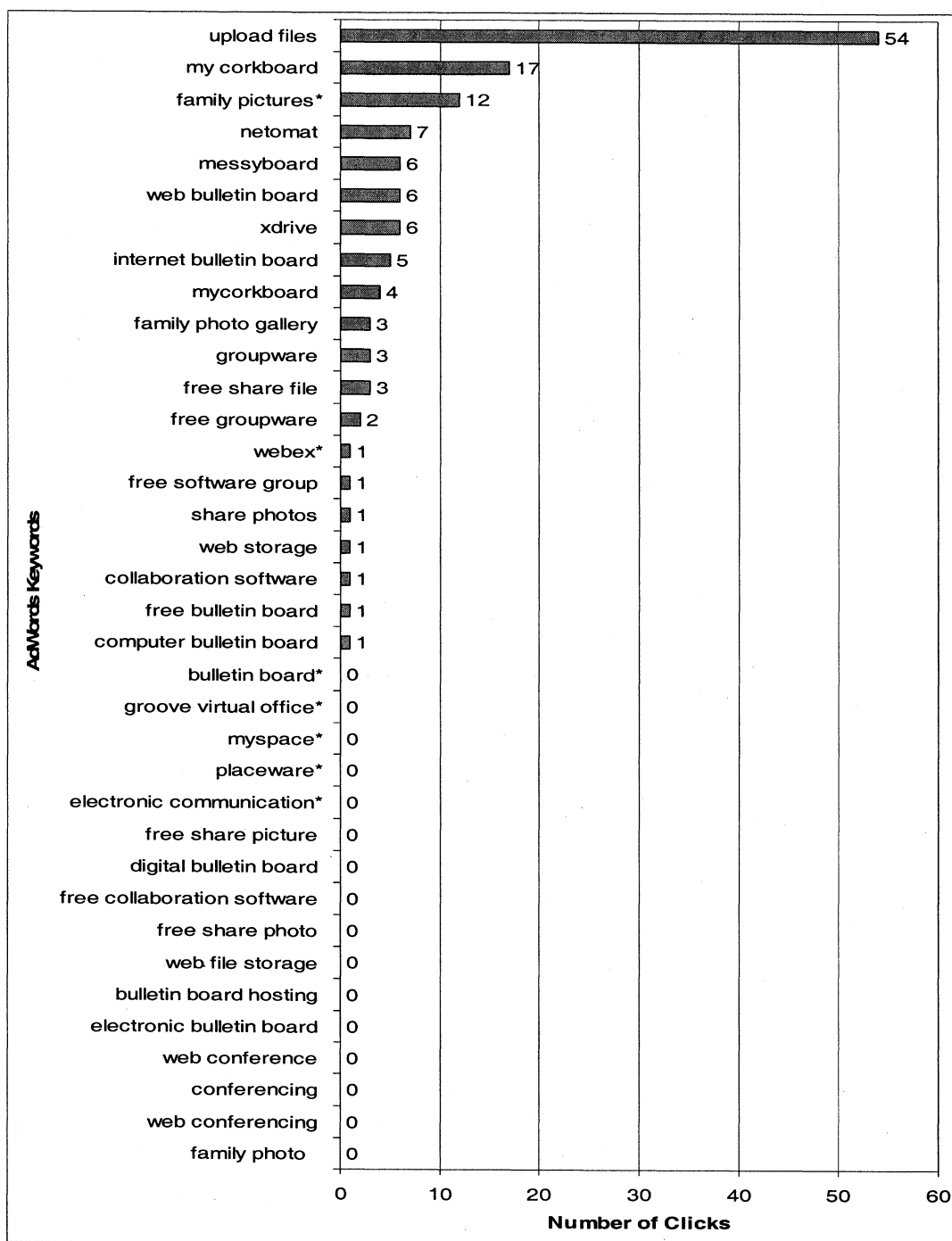


Figure 5.52. The number of clicks on the AdWords advertisement shown in Google's search results for each keyword. The total number of clicks is 135. Keywords marked with a * were disabled early on because too few users clicked on them relative to the number of times that the advertisement was shown.

I looked at each space individually and categorized them by their intended purposes, regardless of how much they were actually used. Of the 124 spaces, 54 had enough content to suggest an intended purpose. I selected a set of categories and then iteratively classified each board and refined the categories. The resulting categories are:

- Group communication: communication and coordination among a restricted set of people
- Group file sharing: sharing files among a restricted set of people
- Couple communication: communication between a pair of people who appear to be romantically involved
- Synchronous photo sharing: one person showing another person photos synchronously
- Personal organization: a single user keeping track of things to do and/or scheduled appointments
- Personal storage: a single user storing files and/or pictures

The counts for each category are plotted in Figure 5.53. If most users did in fact discover MessyBoard while searching for a way to store or share files, they must have quickly realized that MessyBoard was not the tool they were looking for. The vast majority of users who did anything with their MessyBoard spaces intended to use them for some kind of group communication. Only four spaces were intended primarily for file storage, and it does not appear that any spaces were used for distributing files to the general public.

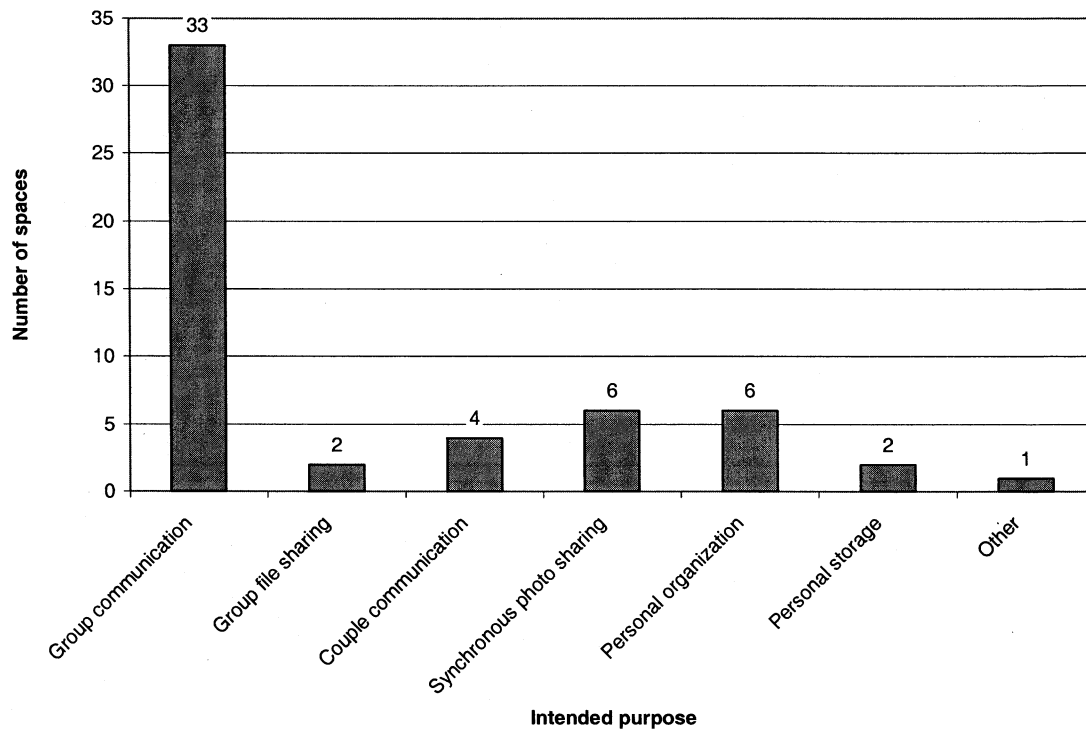


Figure 5.53. The number of spaces created for each intended purpose (54 in total). All of the spaces created for synchronous photo sharing appear to be created by the same person.

Figure 5.54 shows only the boards that were intended for group communication broken down by the kind of group. There is no clear trend toward any particular kind of group.

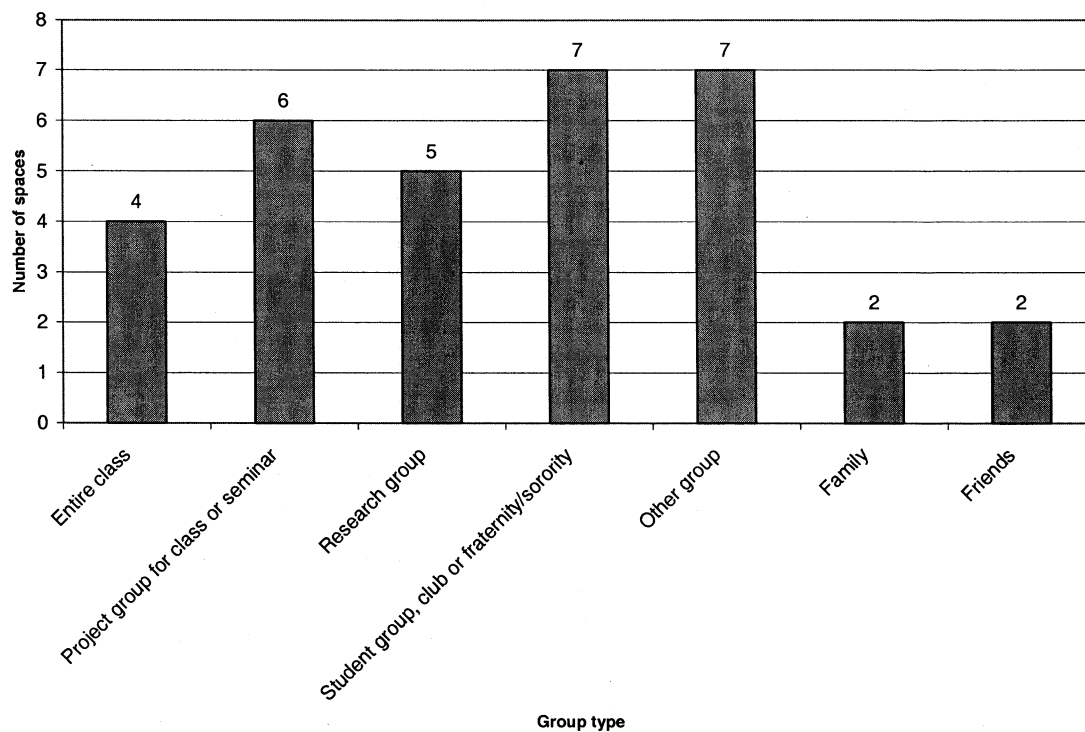


Figure 5.54. The number of spaces intended for group communication for each kind of group (33 in total). Three of the four spaces for entire classes appear to be for different kinds of discussion in the same class.

5.7.2.2 How Long People Used MessyBoard

Most users did not use their MessyBoard spaces for very long. Figure 5.55 plots the cumulative number of MessyBoard spaces vs. the minimum life span of the spaces in weeks. The life span is the amount of time between the creation of the space and the last time it was modified by a user. The maximum possible life span in the period of analysis is 21 weeks. The fact that none of the servers have the maximum possible life span does not imply that none of them are still in use. This is because a space that was created in the middle of the analysis period cannot have a life span exceeding the remainder of the analysis period. Four of the MessyBoard spaces were modified in the last two weeks of the analysis period.

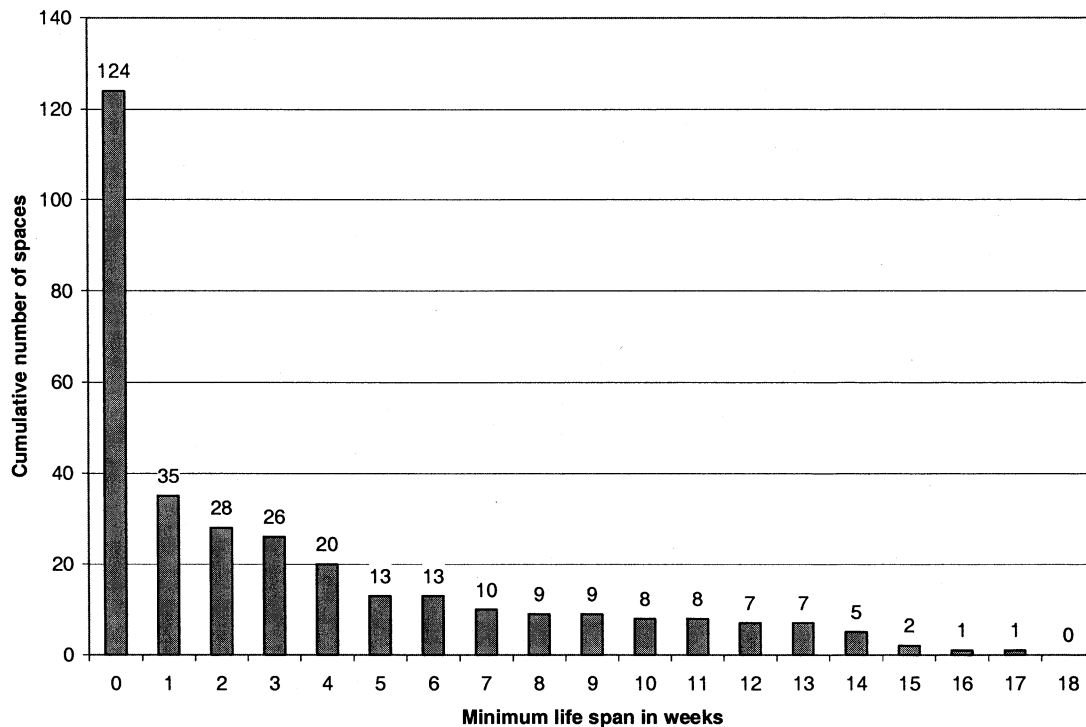


Figure 5.55. Cumulative number of MessyBoard spaces vs. minimum life span. For each life span on the X axis, the bar indicates how many spaces were used for at least that many weeks. All 124 spaces were used for zero or more weeks, 35 were used for one or more weeks, etc.

As with the MessyBoard spaces used by groups at Carnegie Mellon University, my observations of the logs for individual MessyBoard spaces revealed that spaces were often modified regularly for a time, left unmodified for a long time, and then modified once or twice after the long period of silence. These final modifications were minor: Typically a single object was moved or deleted. It is not clear why users often make minor modifications to their MessyBoard spaces after long periods of inactivity but never resume modifying it regularly. However, this phenomenon makes the life span, as defined above, somewhat misleading.

To correct for this phenomenon, I define the **trimmed life span** as the time between the creation of the space and latest modification that is at least 48 hours before the final modification. Thus, spaces that are continually in use will lose only 48 hours by this defi-

dition, but spaces that lay dormant for long periods with one minor modification at the end will have their life spans shortened to the period where they were actually in use. Figure 5.56 plots the cumulative number of servers vs. the trimmed life span in weeks. Here, the drop in the number of spaces after one week is even more dramatic.

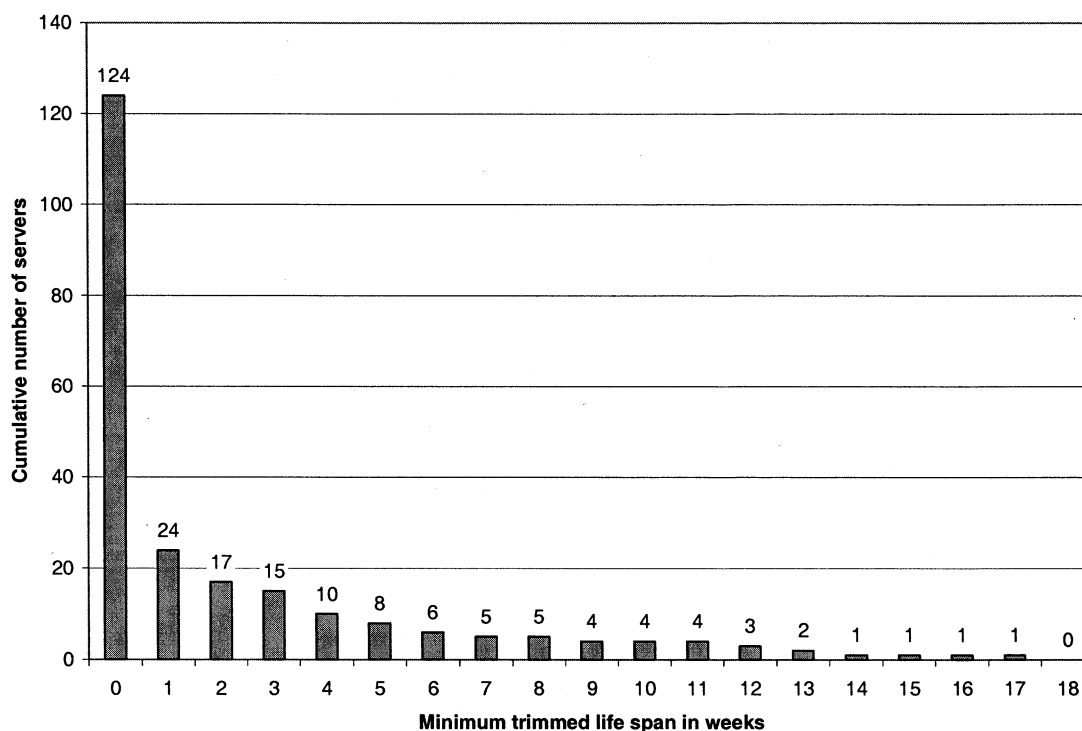


Figure 5.56. Cumulative number of MessyBoard spaces vs. minimum trimmed life span.

5.7.2.3 How Many People Used Each MessyBoard Space

The number of users for a particular MessyBoard space is difficult to estimate since some users are likely to have dynamically assigned IP addresses and many users did not use the identity feature. For each space, I estimate a lower bound on the number of users for that space by taking the maximum of the number of identities and the largest number of simultaneous connections to that space. Since MessyBoard is often used asynchronously and many users do not create identities, this is a conservative estimate.

Figure 5.57 plots the cumulative number of servers for each minimum number of users. About half of the spaces have at least two users and about a quarter of the have three or more users.

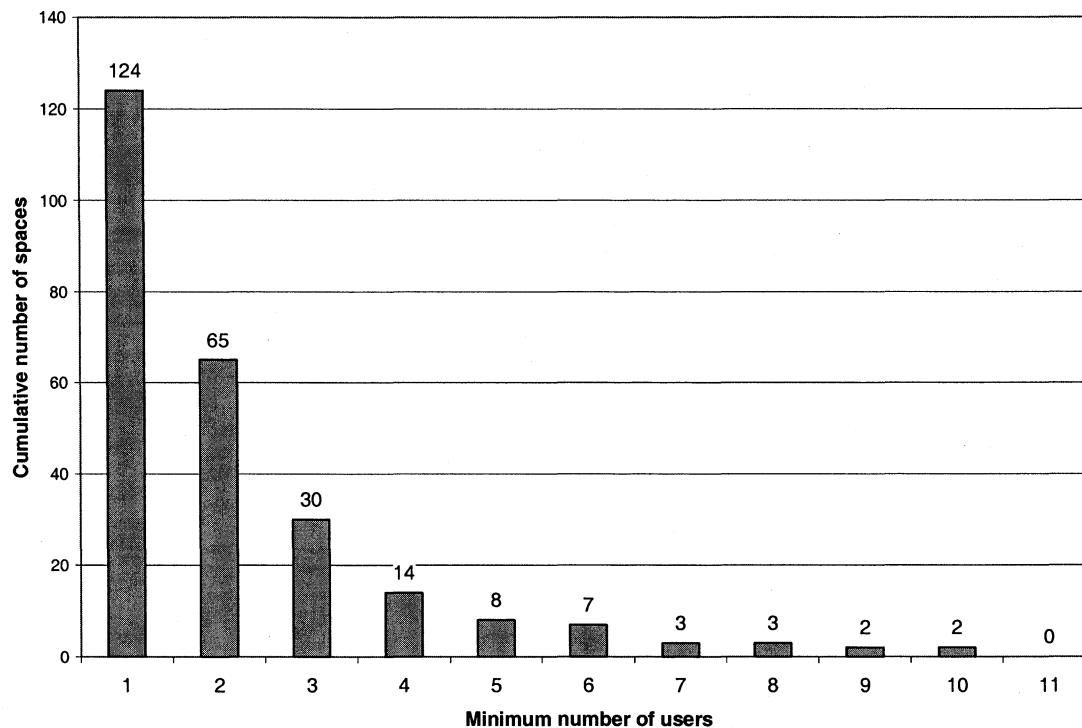


Figure 5.57. The cumulative number of spaces for each minimum number of users. All 124 spaces had at least one user, 65 spaces had two or more users, etc.

5.7.2.4 Real-time Collaboration

Though MessyBoard is designed primarily for asynchronous collaboration, some of the spaces were used predominantly for real-time communication such as chatting and sharing pictures. I define a space to be dominated by synchronous activity if at least half of the transactions take place within ten minutes of another transaction by another user.

Based on this definition, 21 of the 124 spaces are dominated by synchronous activity. Of these spaces, six were apparently created by the same person and used only once for sharing photographs of a trip to Mexico. Ten of the spaces were used only as a novelty and abandoned after an initial spurt of activity. The remaining five were used for a variety of

purposes including sharing multimedia files for a collaborative project, planning a wedding, goofing off and socializing. The average life span for these five remaining boards is approximately 7.3 weeks.

5.7.2.5 Which Features Were Used

Figure 5.58 depicts the average number of times that each kind of object was created, the number of times users changed the background image and color and the number of times they clicked on the history slider over all 20 of the spaces with a life span of greater than four weeks. Notes were created far more often than any other kind of object. Links were the least popular kind of object and users made few changes to the background image or background color. Though files were used a good deal more than pictures and drawings on average, further analysis reveals that this difference is largely due to the users of one MessyBoard space who used it intensively for file sharing. If this outlier is excluded, file creation drops to an average of 8.4.

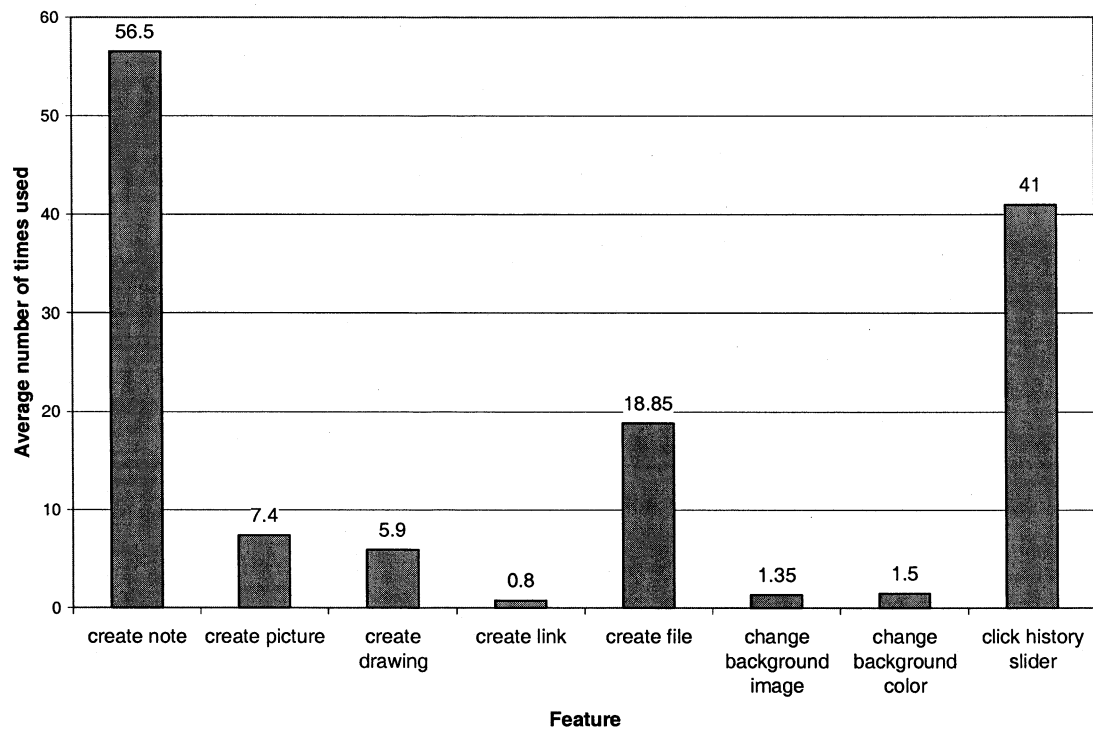


Figure 5.58. The average number of times that various features were used for the 20 MessyBoard spaces with a lifespan greater than four weeks.

Figure 5.59 plots the median counts for the same set of spaces and features. Here we see at least half of the MessyBoard spaces never contained a file or link, but many users uploaded a picture or created a drawing at least once.

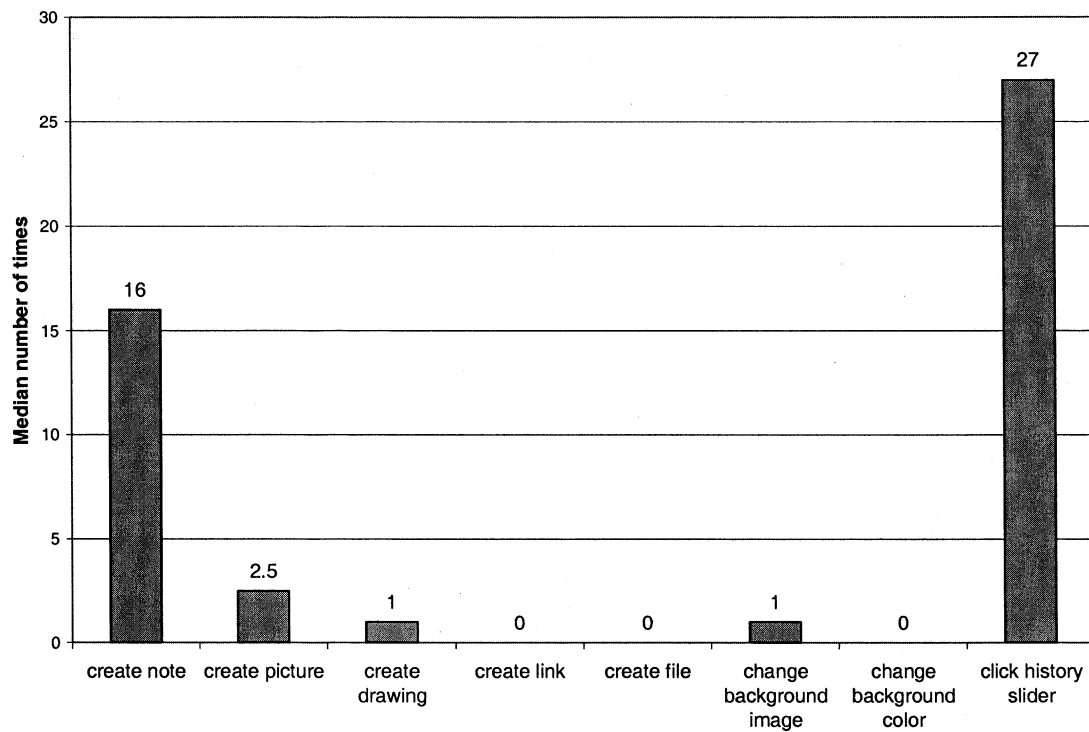


Figure 5.59. The median number of times that various features were used for the 20 MessyBoard spaces with a lifespan greater than four weeks. (The median for odd-numbered data sets is the average of the two values in the center.)

5.7.2.6 Screen Saver Use

Twelve of the 124 MessyBoard spaces logged at least one connection from the MessyBoard screen saver. For eleven of these spaces the connection was logged in the first week after the space was created. Figure 5.60 depicts the trimmed life span distribution for spaces that logged at least one connection from the screen saver. Even for users who tried the screen saver, it is still the case that most MessyBoard spaces are not used for very long.

5.7.2.7 Interesting Spaces

Despite the fact that most MessyBoard spaces were not used for very long, six of the spaces were used in interesting ways and two of them are still being used at the time of

this dissertation's completion. Here I offer brief descriptions and screen shots of these spaces.

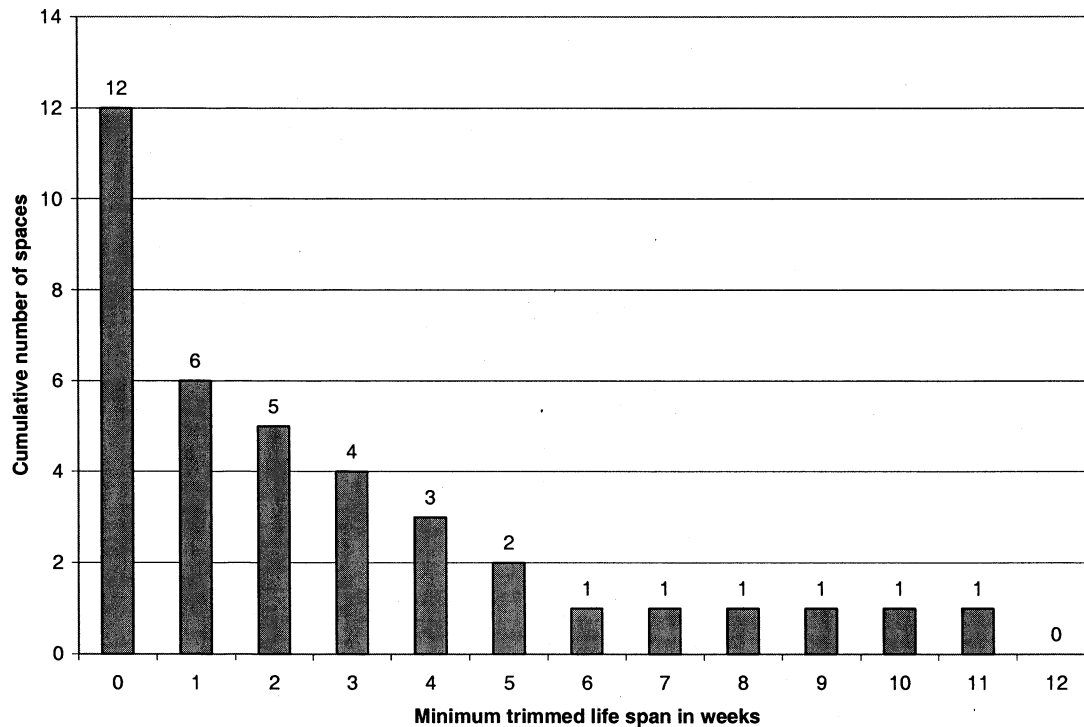


Figure 5.60. Cumulative number of MessyBoard spaces vs. minimum life span for spaces that logged at least one connection from the screen saver.

Figure 5.61 shows a space used by an engaged couple. Though they do not use the MessyBoard screen saver, they view and modify their MessyBoard space frequently. The space is filled with to-do lists and short messages that they send to each other, and much of the content seems related to the planning of their upcoming wedding. Most of the activity in this space is synchronous, though much of the content is long-lived and used asynchronously. This MessyBoard has been in use for over three months and is still in use as of the completion of this dissertation.

Figure 5.62 depicts a MessyBoard space used by a group of female college students. They used MessyBoard to exchange humorous pictures and comments and most of their interaction was synchronous.

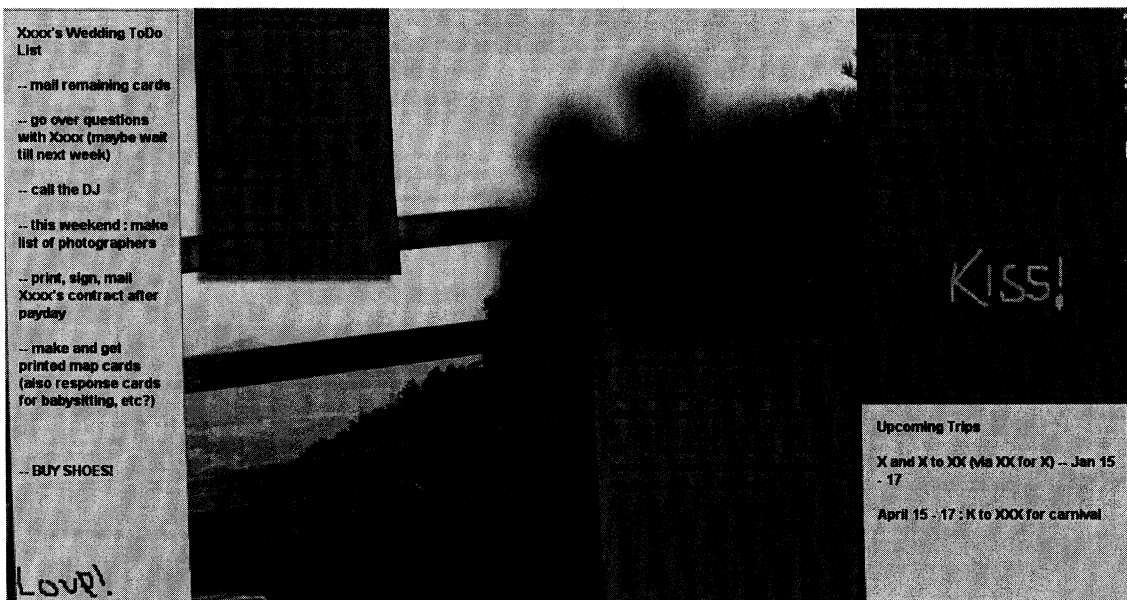


Figure 5.61. A couple uses MessyBoard to manage to-do lists and to send messages to each other. The image has been blurred to maintain their anonymity.

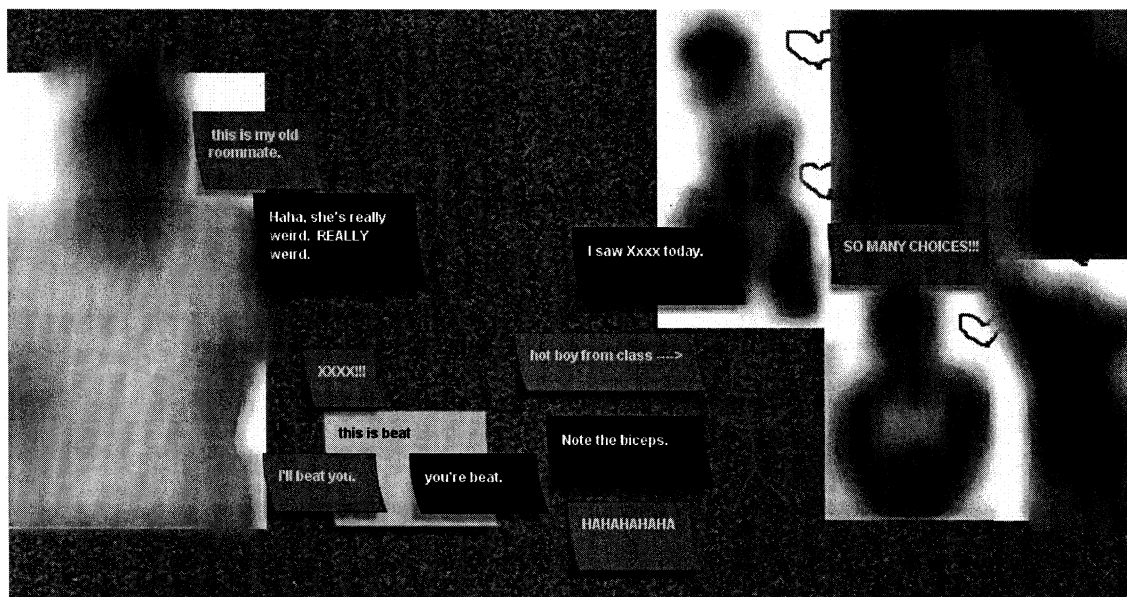


Figure 5.62. A group of female college students use MessyBoard to share comments and pictures. The images have been blurred to maintain their anonymity.

Figure 5.63 depicts a MessyBoard space used by the members of a band. They used it continuously for about two and a half weeks to share humorous comments, pictures and drawings and to find a time when all of them would be available to rehearse. This space appears to have at least one user in common with the space described above.

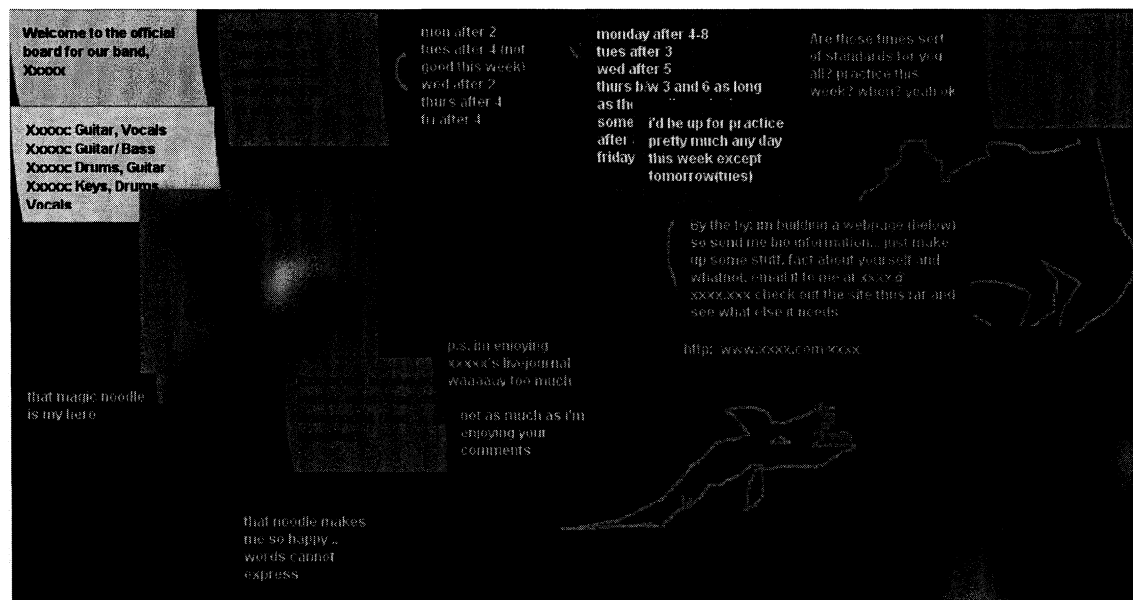


Figure 5.63. Members of a band use this space to chat and find a time to rehearse. The images have been blurred to maintain their anonymity.

Figure 5.64 shows a MessyBoard space used by members of a short-term project team. An inside source revealed that these users were visitors to Group C, described in Section 5.5.1.1.3. The collaborated over several weeks to create a digital interactive entertainment experience as part of a seminar. The used MessyBoard extensively to share files and they organized their files by placing related groups on top of a single note with a label. This use is similar to that of Group D as described in Section 5.5.2.2.1.

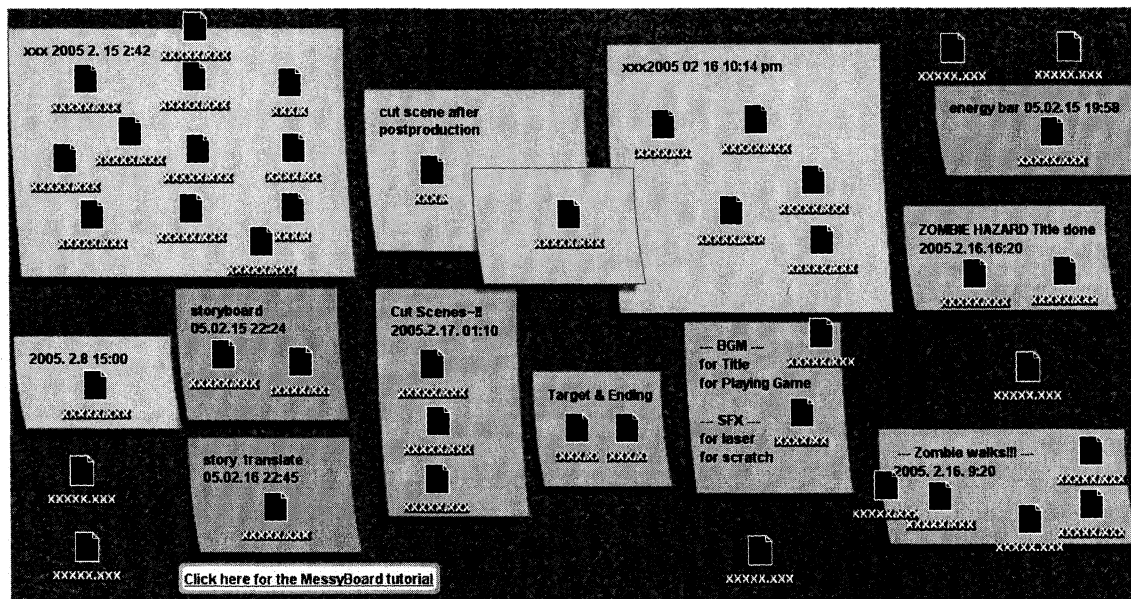


Figure 5.64. A short-term project team uses MessyBoard to share files while creating an interactive entertainment experience.

Figure 5.65 shows a MessyBoard space that is apparently used by a single user as a general-purpose storage space for pictures and files. This user is probably a member of Group C. Though the logs show that two clients had simultaneously connected on at least one occasion, there is nothing on in this space that suggests communication between two users. More likely, this user used MessyBoard on two separate computers, perhaps as a way to transfer files. This MessyBoard space is still in use as of the completion of this dissertation.

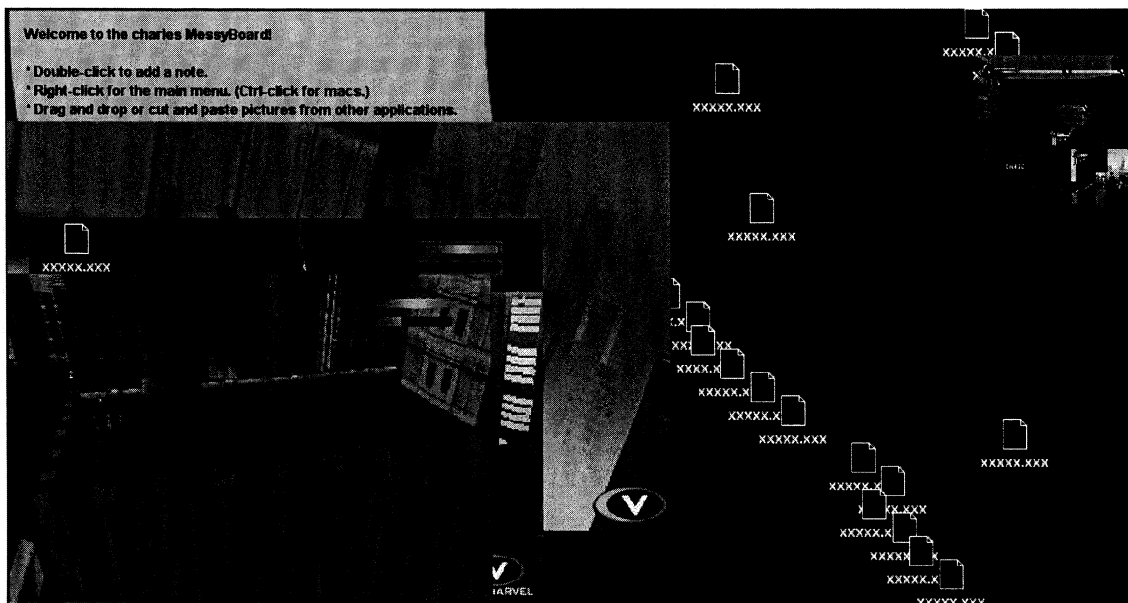


Figure 5.65. A single user uses MessyBoard as an all-purpose storage space.

5.7.3 Discussion and Conclusion

The results of the public deployment are simultaneously disappointing and encouraging. On the one hand, it is clear that very few of the MessyBoard spaces that users created were really used in any substantial way. Most users do not choose to install the screen saver, and for those that do, it does not seem to correlate with any important differences in MessyBoard use.

On the other hand, it is known that groupware adoption is an extremely difficult problem, even when organizations make concerted efforts to get their members to use new groupware systems and devote resources to persuasion and training (Grudin, 1994; Herbsleb et al., 2002; Markus & Connolly, 1990; Orlikowski, 1992). The MessyBoard public deployment has none of these advantages; A single user must create a space, figure out how to use it and what to use it for, and single-handedly convince others to use it. In this context, it is fairly remarkable that even a handful of MessyBoard spaces were used for more than a few weeks.

Despite the fact that a few MessyBoard spaces were used for interesting purposes, these results cast serious doubt over the prospect of a profitable public MessyBoard server

open to use by the general public. It might still be possible that there exists a particular demographic of users who would derive great benefit from MessyBoard, but the behaviors of the users of the interesting MessyBoard spaces in this study are too diverse to suggest a clear niche.

Chapter 6

Conclusion and Future Work

6.1 Summary

In this dissertation I have addressed the problem of the cost of communication. I have created MessyBoard, a communication medium designed specifically to reduce the authoring cost, receiving cost and the startup cost of communication. In addition, I have proposed two ways to display MessyBoard in order to further reduce the costs of communication: On a large public display and as a screen saver.

I reduce the startup cost of communication by deploying MessyBoard as a Java applet. Users can use MessyBoard by typing an easy-to-remember URL into any computer and they can create a new MessyBoard space just by navigating to the desired URL.

MessyBoard reduces the authoring cost of communication by providing a WYSIWIS space in which users can use proximity, overlap and other kinds of spatial arrangement to imply relationships between objects in the space. The authoring cost is further reduced by a fine-tuned interface that makes the most common operations quick and easy (i.e. double clicking to create a note) and allows the user to import information by dragging and dropping or cutting and pasting.

The receiving cost of communication is reduced because MessyBoard provides a single finite space with no navigation interface. The accumulation of clutter in the space is mitigated by the history feature, which gives users the confidence to clean up the board and delete old information.

Displaying MessyBoard on a large public display further reduces the receiving cost by allowing users to see MessyBoard at convenient times without interrupting them or requiring them to remember to check it. Large public displays can be costly and require

maintenance, and I have implemented a MessyBoard screen saver for Windows as an alternative means to this end.

I have deployed the system and observed pre-existing groups using it for research, course projects and other kinds of collaborative activities. My results suggest that, in its present form, MessyBoard may be best suited to larger groups (25 or more people) who typically use it for a combination of playful and goal-directed behavior. The smaller groups that use MessyBoard and find value in it tend to use it for a specific work-related function, and they use it most often as an annotated file repository.

6.2 Principles of Communication

With the benefit of hindsight, it became clear that all of my design decisions stem from a small set of intuitive principles that I believe determine peoples' communication habits. Here I explicitly articulate those principles, discuss my main results in light of them, and relate the principles to other kinds of communication tools.

Group communication comprises individual acts of communication by group members, and the overall level of communication is determined by the frequency of these individual acts. Group members make many small decisions moment to moment about whether and how to perform acts of communication. Should I call Joe now or just wait until I run into him? Do I have time right now to compose a concise and unambiguous message? Should I start reading my e-mail right now or do it later?

The most important decision a person makes about an individual act of communication is about whether and when to begin. The decision to begin is much more important than any decision the person must make while completing the act. Once a person begins working on a task, they are unlikely to abandon it before it is complete, no matter the cost. (This is obviously truer for fine-grain communication, such as e-mail messages, than for coarse-grain communication, such as white papers.) Finishing tasks makes people feel good, even if the tasks are unimportant. Giving up or being forced to quit in the middle of a task causes stress and makes people feel like they have wasted time.

The decision to begin is influenced by the perceived cost in terms of time and effort, but not necessarily the actual cost. For novel tasks, the actual cost is unknown. People overestimate the costs of tasks that are known to be frustrating or boring. The actual cost is only important in so far as it affects the perceived cost.

All else being equal, the higher the perceived cost, the less likely a person is to begin the act right away. People generally have a number of tasks that need to be completed and they must choose one. People are rewarded by a sense of completion, and the task with the lowest perceived cost is the one that a person thinks will yield that reward most quickly.

The presence or absence of consequences also affects the decision to begin; otherwise people would never begin tasks that are perceived to be costly. As with cost, perceived consequences are more important than actual consequences. The combined effect of perceived consequences and perceived cost explain why people do eventually begin to work on tasks that are perceived as costly but they seldom begin working early enough to complete the task on time without heroic measures at the end.

The perceived cost is greatly affected by the presence or absence of a mental model for how that task is to be accomplished. When I say that a person has a mental model, I mean that they can envision a set of concrete actions that will lead to the completion of the task. The existence of a mental model is a binary variable: a person either has or does not have a mental model, and that makes most of the difference with respect to perceived cost. When a person has no mental model for how a task is to be accomplished, the task is intimidating and the person assumes the worst in terms of cost. When a person has a mental model, the task appears tractable and the person will likely underestimate the cost. The relationship between the mental model and the actual steps used to accomplish a task is almost irrelevant. The mere existence of any mental model at all bounds the perceived cost and makes the person that much more likely to begin. Once the person begins, she will likely complete the task.

The tools at hand influence the likelihood that a person will have a mental model or that the person will be able to invent one quickly. Some tools are designed to accomplish par-

ticular tasks, and the mental models are built into the interface. When the users learn to use the tools, they automatically learn the mental models. Other tools are deliberately unstructured. They provide no obvious mental models for specific tasks, but the capabilities that they provide allow users to quickly imagine models for some tasks.

A communication tool that causes people to perceive costs of communication as being low will result in more communication. A tool can influence the perceived cost in two ways:

1. By providing obvious mental models for accomplishing specific tasks
2. By enabling the user to quickly invent a mental model for a task that they need to accomplish

The latter applies to MessyBoard. By providing a medium that is persistent and freeform, MessyBoard enables users to quickly invent models for some kinds of communication tasks. The following section provides an example taken from my field observations.

6.2.1 Applying Principles to Results

My main result is that large groups (20 or more members) are more likely to adopt MessyBoard than small groups (15 or fewer members) and that the small groups that do use MessyBoard use it most often as an annotated file repository. In light of the principles that I have articulated, these results can be explained as follows.

In a large group, any given pair of members may have no working relationship at all and the entire group meets infrequently, if ever. There is no default mental model for accomplishing tasks that require input from the entire group. Existing communication tools enable low-cost asynchronous broadcast communication (e-mail) or synchronous two-way communication (chat), but these capabilities do not suggest complete mental models for all communication tasks. In such an environment, a new communication tool that suggests mental models for completing tasks can make a great difference.

For example, consider the task of collecting a list of birthdays for all members of a large group. E-mail allows a collector to ask everyone to send their birthdays to her, but the e-

mail correspondence does not accomplish all of the work. She must go through all of the replies and make the final list. If she thought about it for a moment, she might imagine this simple mental model, but she does not have that moment. If she has another task on her to-do list for which she already has a mental model, she will begin working on that task instead of thinking about the birthday list task.

Consider the same situation in a group that uses MessyBoard. Upon seeing the birthday task, the collector might immediately realize that she can start a list on MessyBoard and that people will add their birthdays to it. The process itself will create the final list, which can quickly be pasted into another software tool if necessary. A complete mental model is instantly available for this task, greatly increasing the likelihood that the collector will begin the necessary act of communication.

In a small group, each member is more likely to have close working relationships with all other members and the entire group is more likely to meet on a regular basis. This means that in a small group, all else being equal, group members are more likely to have existing mental models for communication tasks and they can easily invent new models. The default mental model for all communication tasks is "I'll bring it up the next time we all get together". New communication tools that suggest new mental models will make little difference in this environment with respect to the likelihood of individuals beginning acts of communication.

The size of the group has little effect on the existence of mental models for sharing digital media. Close working relationships or meetings at which all members are present do not automatically allow group members to share digital documents, artwork or code. MessyBoard suggests a simple model for sharing files: drag the file to MessyBoard and notify the group that it is there. Perhaps this explains why file sharing is a leading use of MessyBoard among small groups. Once a group begins using MessyBoard to share files, annotation and spatial grouping follow naturally.

6.2.2 Applying Principles to the Design of Communication Tools

The principles set forth here imply that the relationship between a new communication tool and the mental models of its users will determine whether the tool is adopted and whether it increases the overall level of communication in a group. This relationship between a tool and users' mental models depends on the characteristics of the tool and the characteristics of the group.

For small collocated groups, tool makers may do best to focus on specific, possibly domain-dependent tasks that are not addressed by current tools. Small groups are unlikely to adopt a new general-purpose communication medium because they already have mental models for accomplishing most communication tasks, including tasks that they have not encountered yet. Small groups' use of MessyBoard for sharing files is an exception. This usage owes more to the astoundingly sorry state of sharing capabilities in current consumer operating systems than it does to the design of MessyBoard.

For large and/or distributed groups, I previously mentioned two ways that tools can increase the likelihood of acts of communication. One is to design the tool around specific tasks, thereby building the mental models into the interface. Shared calendar systems such as Microsoft Outlook (Microsoft, 2005b) are a good example of this. Outlook's calendar shows a person when a colleague has free time, allows her to schedule a meeting and automatically sends an e-mail announcement to all invitees. A person who knows this and has access to Outlook is very likely to begin the task of scheduling a meeting.

The other way for tools to increase the likelihood of acts of communication is to enable users to quickly invent mental models for novel tasks. Media that are WYSIWIS, free-form and persistent allow users to invent mental models for a wide range of communication tasks because they allow users to instantiate or adapt models that are already familiar, such as the following:

- Everybody adds information to a list
- Everybody expresses an opinion
- Everybody updates the current state to reflect personal knowledge

MessyBoard and Wikis (Cunningham, 2005) are both examples of media that are WYSIWIS, freeform and persistent. MessyBoard spaces and Wikis differ in ways that are orthogonal to this discussion: finite vs. infinite space and one dimension vs. two dimensions. I have argued that these characteristics affect the authoring and receiving costs of communication, but they are much less relevant to the likelihood that an individual will begin an act of communication.

6.2.3 Communication Principles and Rational Decisions

I began with the notions that people are likely to complete tasks once they begin and that the decision to begin is determined by perceived costs and consequences. I postulated that the perceived cost depends a great deal on the presence or absence of a mental model for completing the task. More abstractly, I assume that an aggregate behavior (the overall level of communication in a group) is the product of decisions made by individual actors and that those actors make their decisions based on a very limited set of criteria (perceived costs and consequences).

The reader may be stricken by my focus on snap judgments of the perceived costs of tasks in the heat of the moment. What about the actual cost of the task and its importance relative to other tasks? Should not we encourage people to take these factors into consideration? My views on this subject are implicit in the principles outlined above, and I will make them explicit here: people do not make rational decisions about the relative importance of tasks and their costs. It is the perceived cost that matters most.

As system designers, we can only persuade people to make decisions that are rational in a normative sense by understanding and harnessing the real, irrational factors that influence human decision-making. I hypothesize that people do not communicate enough because the perceived cost is too high relative to the perceived costs of other activities. I aim to lower the perceived cost in order to correct for this imbalance so that the decisions of individuals will be closer to an assumed normative standard. In short, I believe we should aspire to design an environment in which the right choice is also the easy choice.

6.3 Future Work

6.3.1 Interface Improvements

MessyBoard could stand to benefit a great deal from three relatively minor additions. The first is to allow users to set “expiration dates” on their notes so that they could be deleted automatically when they are no longer useful. Several users have indicated in interviews that they would be willing to do this extra work up front in order to keep the board clean. An alternative requested by some users is a system to allow each user to “check off” a note when she has read it. A note might be automatically deleted when each user checks it off, or the check list might simply allow a user to determine which notes are no longer needed.

Another proposed improvement is a list object tailored specifically to the task of creating lists or tables of individual items. Many groups create lists and then check items off, rearrange items or moving them between multiple lists. This behavior can be tedious using a single note as a list and having a note for each item occupies too much space. A special-purpose list object could present many items in a compact list or table and provide a way to check items off or rearrange them by dragging and dropping.

Finally, several users requested the ability to set “bookmarks” at important times in the history. Users indicated that they would do this after a meeting or before deleting important information and that this would give them added confidence that they could recover the information later. Another obvious addition to the history interface is a search mechanism. Both would be beneficial, but it is interesting to note that the bookmark feature may give the user more confidence in deleting old material and that is the ultimate goal of the history feature.

6.3.2 Integration with a Repository

Since MessyBoard provides a finite single 2D space, it can not be used to organize large amounts of information. I have argued that the limited space reduces the receiving cost of communication, but some applications require large amounts of information to be shared and archived in a highly structured fashion. For example, software engineers often enter

bug reports in a special purpose database, movie and video game studios use asset management systems to keep track of 3D models and 2D images, and many businesses use document repositories to keep track of multiple versions of forms and reports. I refer to all of these applications as “repositories”.

Repositories serve two purposes: communication and archiving. For example, a bug database tells a software engineer if anyone is fixing the bug right now, and it also provides a history of modifications to the software. These workers might benefit in the short term from low-cost communication with MessyBoard, but in the long term they would suffer from MessyBoard’s lack of support for a structured archive for old information.

My proposed solution is to tightly integrate MessyBoard with a repository so that workers can reduce the cost of communication without sacrificing the benefits of a structured archive. For example, suppose that a company uses a document repository with a web front end such as Microsoft Sharepoint (Microsoft, 2005c). MessyBoard could appear on the web site and employees could use it to collaborate on a small working set of documents. The documents on MessyBoard would actually be links into the repository. When users drag new documents from their local file systems onto MessyBoard, they could be added to the repository automatically. In some businesses, the work flow is sufficiently structured that the system might guess the correct location for the document in the repository or automatically detect if it is a new version of an existing document. Users could also drag documents from the repository to MessyBoard to create links in order to facilitate collaboration or announce information to their colleagues.

MessyBoard’s history feature could provide a useful perspective on the entire repository as a visual history of the surrounding collaboration. The repository stores multiple versions of old documents, but the MessyBoard history might show the user other documents that were created or modified at the same time as well as to-do lists and announcements from that time. Even jokes or pictures might help users remember more about the circumstances surrounding modifications to documents in the archives. New employees could browse the MessyBoard history to get a sense of the company’s recent

history. All of this information would be generated by users as a natural by-product of their collaboration.

Appendix A: Third Party Libraries and Components

The MessyBoard implementation uses several third-party libraries and components. The following table lists these components and how they are used.

Component: Acme

Use: Encode GIF files for automatically resized images (server)

Home page: <http://www.acme.com/java/software/>

Component: Apache HTTP Server

Use: Serve images and files over HTTP (server), host messyboard.org

Home page: <http://httpd.apache.org/>

Component: KFC

Use: Display editable lightweight text areas for notes (client)

Home page: <http://openlab.jp/kyasu/>

Component: Java

Use: Main implementation language (client and server)

Home page: <http://java.sun.com/>

Component: mod_python

Use: Server-side creation of dynamic HTML for messyboard.org

Home page: <http://www.modpython.org/>

Component: Python

Use: Server-side creation of dynamic HTML for messyboard.org

Home page: <http://www.python.org/>

Component: Xerces

Use: Write current state and history of MessyBoard space to XML file (server)

Home page: <http://xml.apache.org/xerces2-j/>

Appendix B: Interview Questions for Bulletin Boards

These questions were asked in face-to-face open-ended interviews.

1. Who owns this bulletin board?
2. What is this bulletin board used for?
3. Are there any rules about posting material or removing it from the board?
4. Is someone responsible for maintaining this bulletin board?
5. Do you feel that this bulletin board is useful?

Appendix C: Online Survey Items

Participants filled out the following survey in a web browser and results were processed with a server-side Python script. The text in bold below was replaced with relevant names and descriptions for the groups.

This survey asks about your use of the MessyBoard space "**example**". In this survey, the term "**your group**" refers to **all of the people in the example group**. Your group may include people who never actually used MessyBoard. If you are a member of other groups or you use other MessyBoard spaces, please disregard them for this survey.

Your MessyBoard Space: **example**

Your group: **all of the people in the example group**

Please answer these questions based on your experiences with your group **during the example time period**, when you were all working together on your project. All of your answers should reflect what you were doing at that time, not anything that has happened since.

Please be sure to answer every question.

1. How many people are in your group? Please include everyone in your group as defined above, not just the people who use MessyBoard. If you aren't sure, enter your best guess.

Number of people in group:

2. How long has your group existed? Please indicate the amount of time that **your group** has existed, **not** the amount of time that your MessyBoard space has existed. If you aren't sure, enter your best guess.

- ☐ Less than 1 month
- ☐ At least 1 month but less than 6 months
- ☐ At least 6 months but less than one year
- ☐ One year or more

3. Which of the following best describes the locations of people in your group?

- ☐ Everyone works in the same room
- ☐ Everyone works in the same building but at least one person works in a different room
- ☐ Everyone works at the same site or campus but at least one person works in a different building
- ☐ At least one person works at a different site or campus

4. Does your group collaborate on any of the following activities? Include any activity that your group does **even if you do not use MessyBoard** for that activity. Please **do not** include activities that members do individually or in small subgroups. Check all that apply.

- ☐ My group schedules activities such as meetings
- ☐ My group plans events
- ☐ My group gives public presentations
- ☐ My group writes documents
- ☐ My group creates web sites
- ☐ My group creates art or visual designs using traditional media (paint, pencil, etc.)
- ☐ My group creates art or visual designs using digital media (digital painting, 3D modeling, etc.)
- ☐ My group designs physical products
- ☐ My group communicates with clients or customers

5. Please write a brief description of how your group uses MessyBoard. Please include all of the different things that you do with it.

For Example:

- We share jokes and interesting web pages
- We display architectural diagrams and comment on them
- We maintain a calendar of events for the week

Describe how your group uses MessyBoard here:

6. How often do you (yourself, not anyone else) post information on the group1_board_name MessyBoard?

- ☐ Never
- ☐ At least once but less than once a month
- ☐ At least once a month but less than once a week
- ☐ At least once a week but less than once a day
- ☐ Once a day or more

7. How often do you (yourself, not anyone else) see the group1_board_name MessyBoard? Please include all times that you see the group1_board_name MessyBoard, including when you see it projected on the wall or as a screen saver on your computer or someone else's computer.

- ☐ Never
- ☐ At least once but less than once a month
- ☐ At least once a month but less than once a week
- ☐ At least once a week but less than once a day
- ☐ Once a day or more

8. Some groups have leaders who have authority over other members of the group. Leaders may set goals for the group, make important decisions or assign tasks to other group members. A leader may have a formal title or a higher rank than other group members.

Examples of leaders include:

- A professor in charge of a research group
- A manager in charge of workers
- A group member that has been elected or appointed to make decisions or manage the group

Which of the following best describes the leadership of your group? (Your group includes all of the people in test group 1)

- ☐ My group has no leader
- ☐ My group has a single leader
- ☐ My group has a few leaders

9. Which of the following statements best describes the influence of the leader(s) on the way that your group uses MessyBoard? If your group has no leader (as defined above) then please select the last choice.

- ☐ Our leader(s) completely determined the way that we use MessyBoard
- ☐ Our leader(s) mostly influenced the way that we use MessyBoard
- ☐ Our leader(s) and other group members both influenced the way that we use MessyBoard
- ☐ Other group members besides the leader(s) mostly influenced the way that we use of MessyBoard
- ☐ Other group members besides the leader(s) completely determine the way that we use MessyBoard
- ☐ Our group has no leader

10. Please Enter all of the deadlines that you can remember for group projects that have occurred since you started using MessyBoard. Please include all deadlines even if the group didn't use MessyBoard for the project. Please **do not** include personal deadlines that only affected you or a small number of group members. If you can't remember exactly when a deadline occurred, please enter your best guess.

Deadline 1: Month:	<input type="text" value="none"/>	Day:	<input type="text" value="none"/>
Deadline 2: Month:	<input type="text" value="none"/>	Day:	<input type="text" value="none"/>
Deadline 3: Month:	<input type="text" value="none"/>	Day:	<input type="text" value="none"/>
Deadline 4: Month:	<input type="text" value="none"/>	Day:	<input type="text" value="none"/>
Deadline 5: Month:	<input type="text" value="none"/>	Day:	<input type="text" value="none"/>
Deadline 6: Month:	<input type="text" value="none"/>	Day:	<input type="text" value="none"/>
Deadline 7: Month:	<input type="text" value="none"/>	Day:	<input type="text" value="none"/>
Deadline 8: Month:	<input type="text" value="none"/>	Day:	<input type="text" value="none"/>
Deadline 9: Month:	<input type="text" value="none"/>	Day:	<input type="text" value="none"/>
Deadline 10: Month:	<input type="text" value="none"/>	Day:	<input type="text" value="none"/>

For each of the following statements, indicate how strongly you agree or disagree.

	1=Strongly Disagree				7=Strongly Agree											
11. I put a lot of time and energy into this group.	<input type="radio"/>	[1]	<input type="radio"/>	[2]	<input type="radio"/>	[3]	<input type="radio"/>	[4]	<input type="radio"/>	[5]	<input type="radio"/>	[6]	<input type="radio"/>	[7]	<input type="radio"/>	Not sure
12. I feel that I am really a part of this group. ¹	<input type="radio"/>	[1]	<input type="radio"/>	[2]	<input type="radio"/>	[3]	<input type="radio"/>	[4]	<input type="radio"/>	[5]	<input type="radio"/>	[6]	<input type="radio"/>	[7]	<input type="radio"/>	Not sure
13. I look forward to being with (or communicating with) members of this group. ¹	<input type="radio"/>	[1]	<input type="radio"/>	[2]	<input type="radio"/>	[3]	<input type="radio"/>	[4]	<input type="radio"/>	[5]	<input type="radio"/>	[6]	<input type="radio"/>	[7]	<input type="radio"/>	Not sure
14. All members of this group work together closely on a single project.	<input type="radio"/>	[1]	<input type="radio"/>	[2]	<input type="radio"/>	[3]	<input type="radio"/>	[4]	<input type="radio"/>	[5]	<input type="radio"/>	[6]	<input type="radio"/>	[7]	<input type="radio"/>	Not sure
15. MessyBoard is useful to my group.	<input type="radio"/>	[1]	<input type="radio"/>	[2]	<input type="radio"/>	[3]	<input type="radio"/>	[4]	<input type="radio"/>	[5]	<input type="radio"/>	[6]	<input type="radio"/>	[7]	<input type="radio"/>	Not sure
16. MessyBoard is enjoyable for my group.	<input type="radio"/>	[1]	<input type="radio"/>	[2]	<input type="radio"/>	[3]	<input type="radio"/>	[4]	<input type="radio"/>	[5]	<input type="radio"/>	[6]	<input type="radio"/>	[7]	<input type="radio"/>	Not sure
17. Some group members post information on MessyBoard more than other members.	<input type="radio"/>	[1]	<input type="radio"/>	[2]	<input type="radio"/>	[3]	<input type="radio"/>	[4]	<input type="radio"/>	[5]	<input type="radio"/>	[6]	<input type="radio"/>	[7]	<input type="radio"/>	Not sure

¹ Adapted from (Cammann, Fichman, Jenkins, & Klesh, 1983)

For each of the activities below, indicate how often people in your group do the activity, how important the activity is to the group, and which methods that people use.

• **Activity: sharing personal photos taken by members of the group**

18. How often do members of your group share personal photos with each other? Please include all of the times that members share photos using MessyBoard and other methods.

- ☐ Never
- ☐ At least once but less than once a month
- ☐ At least once a month but less than once a week
- ☐ At least once a week but less than once a day
- ☐ Once a day or more

19. How important is sharing personal photos to your group?

1=Not important at all

7=Extremely important

- ☐ [1] ☐ [2] ☐ [3] ☐ [4] ☐ [5] ☐ [6] ☐ [7]

20. Which of the following methods does your group use to share personal photos? Check all that apply.

- ☐ MessyBoard
- ☐ E-mail attachment
- ☐ Post printed pictures on a wall, door or bulletin board
- ☐ Face to face conversation
- ☐ FTP
- ☐ A web page
- ☐ A chat program such as ICQ or Instant Messenger
- ☐ A network file sharing protocol like Windows File Sharing, AppleTalk or NFS
- ☐ A physical device such as a floppy disk, USB key or CD-ROM
- ☐ A web picture sharing service such as MSN photos or PictureTrail
- ☐ A web file sharing service such as Groove or XDrive
- ☐ Other:

• **Activity: sharing files such as documents, presentation slides, code, 3D models and textures, etc.**

21. How often do members of your group share files with each other? Please include all of the times that members share files using MessyBoard and other methods.

- ☐ Never
- ☐ At least once but less than once a month
- ☐ At least once a month but less than once a week
- ☐ At least once a week but less than once a day
- ☐ Once a day or more

22. How important is sharing files to your group?

1=Not important at all

7=Extremely important

- ☐ [1] ☐ [2] ☐ [3] ☐ [4] ☐ [5] ☐ [6] ☐ [7]

23. Which of the following methods does your group use to share files? Check all that apply.

- ☐ MessyBoard
- ☐ E-mail attachment
- ☐ FTP
- ☐ A web page
- ☐ A chat program such as ICQ or Instant Messenger
- ☐ A network file sharing protocol like Windows File Sharing, AppleTalk or NFS
- ☐ A physical device such as a floppy disk, USB key or CD-ROM
- ☐ A web picture sharing service such as MSN photos or PictureTrail
- ☐ A web file sharing service such as Groove or XDrive
- ☐ Other:

• **Activity: sharing interesting, fun or humorous material such as news articles or comics**

24. How often do members of your group share interesting, fun or humorous material with each other? Please include all of the times that members share this material using MessyBoard and other methods.

- ☐ Never
- ☐ At least once but less than once a month
- ☐ At least once a month but less than once a week
- ☐ At least once a week but less than once a day
- ☐ Once a day or more

25. How important is sharing interesting, fun or humorous material to your group?

1=Not important at all

7=Extremely important

- ☐ [1] ☐ [2] ☐ [3] ☐ [4] ☐ [5] ☐ [6] ☐ [7]

26. Which of the following methods does your group use to share interesting, fun or humorous material? Check all that apply.

- ☐ MessyBoard
- ☐ A physical blackboard, whiteboard or bulletin board
- ☐ e-mail
- ☐ A standard web page
- ☐ A blog, wiki or other kind of editable web page
- ☐ A chat program such as ICQ or Instant Messenger
- ☐ A web collaboration tool such as Groove or Blackboard
- ☐ Other:

• **Activity: Maintaining a shared list of tasks or items such as a todo list or shopping list**

27. How often do members of your group create or modify shared lists? Please include all of the times that members maintain shared lists using MessyBoard and other methods.

- ☐ Never
- ☐ At least once but less than once a month
- ☐ At least once a month but less than once a week
- ☐ At least once a week but less than once a day
- ☐ Once a day or more

28. How important are shared lists to your group?

- 1=Not important at all 7=Extremely important
- ☐ [1] ☐ [2] ☐ [3] ☐ [4] ☐ [5] ☐ [6] ☐ [7]

29. Which of the following methods does your group use to create or modify shared lists? Check all that apply.

- ☐ MessyBoard
- ☐ A physical blackboard, whiteboard or bulletin board
- ☐ e-mail
- ☐ A standard web page
- ☐ A blog, wiki or other kind of editable web page
- ☐ A web collaboration tool such as Groove or Blackboard
- ☐ Other:

• **Activity: Brainstorming or group discussions**

30. How often do members of your group brainstorm or have group discussions? Please include all of the times that members brainstorm or have group discussions using MessyBoard and other methods.

- ☐ Never
- ☐ At least once but less than once a month
- ☐ At least once a month but less than once a week
- ☐ At least once a week but less than once a day
- ☐ Once a day or more

31. How important is brainstorming or having group discussions to your group?

- 1=Not important at all 7=Extremely important
- ☐ [1] ☐ [2] ☐ [3] ☐ [4] ☐ [5] ☐ [6] ☐ [7]

32. Which of the following methods does your group use to brainstorm or have group discussions? Check all that apply.

- ☐ MessyBoard
 - ☐ Talk in person
 - ☐ Talk on the phone
 - ☐ A physical blackboard, whiteboard or bulletin board
 - ☐ e-mail
 - ☐ A standard web page
 - ☐ A blog, wiki or other kind of editable web page
 - ☐ A chat program such as ICQ or Instant Messenger
 - ☐ A web collaboration tool such as Groove or Blackboard
 - ☐ Other:
-

Click the button below when you have completed the survey.

Submit Survey

Appendix D: Interview Questions for Group A

These questions were asked in face-to-face open-ended interviews.

1. What is your official position? (Staff, faculty, student, etc.)
2. What department are you in?
3. Who is your supervisor/boss/advisor?
4. Who are your students/subordinates?
5. Please give me a brief summary of your education and work history.
6. How did you come to work in this group?
7. How many hours per week would you say that you spend in the graphics lab?
8. How many hours per week would you say that you spend working on lab-related projects?
9. Please describe the projects that you have worked on in the last month.
10. Imagine a typical day some time in the last month and describe it.
11. Do you work closely with anyone else on your projects?
12. Describe the contact that you have with other members of the graphics lab.
13. Do you consider other members of the graphics lab to be close personal friends?
14. Do you see or talk to members of the lab in a social context outside of work?
15. Are there “key” members of the lab in terms of getting work done?
16. Are there “key” members of the lab in terms of having fun?
17. How often do you attend the graphics lab meetings on Tuesdays?
18. Do you think that the meetings are important? Why or why not?
19. What big events (conference deadlines, visits from funding agents) for the group have happened in the last month or so?
20. How will these events affect you?
21. What big events for the group are coming up in the next few months?
22. How will these events affect you?
23. What would you most like to change about the group?
24. What do you like most about the group?

Appendix E: Interview Questions for Course Alpha

These questions were asked in face-to-face open-ended interviews.

1. Briefly describe your role in the group.
2. How much did you use MessyBoard? What for?
3. How often did you look at MessyBoard?
4. Did you use your own computer or a group computer?
5. How often was projector on?
6. In general, what activities do you think MessyBoard is good for?
7. Were there times that you wanted to use MessyBoard but something prevented you?
8. What percentage of the activity on MessyBoard was work-related?
9. What percentage of the activity on MessyBoard was just for fun?
10. What are the best 3 things about MessyBoard?
11. What are the worst 3 things about MessyBoard?
12. Do you have any suggestions for improvements?

Bibliography

- Agamanolis, S. (2002a). Designing Displays for Human Connectedness. *CSCW 2002 Workshop: Public, Community and Situated Displays*.
- Agamanolis, S. (2002b). *iCom: a multipoint awareness and communication portal for connecting remote social spaces*. Media Lab Europe.
www.medialabeurope.org/hc/projects/icom/
- Akiva. (2005a). ideaCenter. <http://www.akiva.com/products/ideacenter/ideacenter1.cfm>
- Akiva. (2005b). WebBoard and WebBoard Meeting.
<http://www.akiva.com/products/webboard/index.cfm>
- AOL. (2005). AOL Instant Messenger. <http://www.aim.com/>
- Apache Software Foundation. (2005a). Apache HTTP Server. <http://httpd.apache.org/>
- Apache Software Foundation. (2005b). Mod_python. <http://www.modpython.org/>
- AT&T Laboratories Cambridge. (1999). VNC. <http://www.uk.research.att.com/vnc/>
- Bardram, J. (2000). Temporal Coordination. *Proceedings of CSCW 2000*, 157-187.
- Bellotti, V., & Bly, S. (1996). Walking away from the desktop computer: distributed collaboration and mobility in a product design team. *Proceedings of CSCW 1996*, 209-218.
- Bellotti, V., & Rogers, Y. (1997). From web press to web pressure: multimedia representations and multimedia publishing. *CHI 1997 Conference on Human Factors in Computing Systems*, 279-286.
- Benford, S., Bowers, J., Fahl, L. E., & Greenhalgh, C. (1994). Managing mutual awareness in collaborative virtual environments. *VRST 1994 Virtual Reality Software and Technology*, 223-236.
- Bercowicz, D. A., Barnett, G. O., & Chueh, H. C. (1999). eWhiteBoard: A Real Time Clinical Scheduler. *American Medical Informatics Association Annual Symposium (poster abstract)*.
- Birch Grove Software. (2005). Screen Pass. <http://www.bgrove.com>
- Bishop, G., & Welch, G. (2000). Working in the Office of Real Soon Now. *IEEE Computer Graphics and Applications*, 20(4), 76-78.
- Bly, S. A., Harrison, S. R., & Irwin, S. (1993). Media spaces: bringing people together in a video, audio, and computing environment. *Communications of the ACM*, 36(1), 28-46.

- Brennan, S. E., & Ohaeri, J. O. (1999). Why do electronic conversations seem less polite? the costs and benefits of hedging. *WACC 1999 International Joint Conference on Work Activities Coordination and Collaboration*, 227-235.
- Brinck, T., & Gomez, L. M. (1992). A collaborative medium for the support of conversational props. *Proceedings of CSCW 1992*, 171-178.
- Burbeck, S. (1992). *Applications Programming in Smalltalk-80(TM): How to use Model-View-Controller (MVC)*.
<http://st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html>
- Buxton, W., Fitzmaurice, G., Balakrishnan, R., & Kurtenbach, G. (2000). Large Displays in Automotive Design. *IEEE Computer Graphics and Applications*, 20(4), 68-75.
- Cabecal. (2005). PinBoard. <http://www.cabecal.com/Products/PinBoard/>
- Cammann, C., Fichman, M., Jenkins, G. D., & Klesh, J. R. (1983). Assessing the attitudes and perceptions of organizational members. In Seashore, S. E., Lawler, E. E., Mirvis, P. H., & Cammann, C. (eds.), *Assessing organizational change*. John Wiley & Sons, Inc.: New York, 71-138.
- Caucus. (2005). Team. <http://www.caucus.com>
- Centra. (2005). Symposium. <http://www.centra.com/products/symposium/index.asp>
- Cheverst, K., Fitton, D., Dix, A., & Rouncefield, M. (2002). Exploring Situated Interaction With Ubiquitous Office Door Displays. *CSCW 2002 Workshop: Public, Community and Situated Displays*.
- Churchill, E. F., Nelson, L., Denoue, L., & Girgensohn, A. (2003). The Plasma Poster Network: posting multimedia content in public places. *INTERACT 2003: IFIP Conference on Human-Computer Interaction*, 599-606.
- Clark, H. H., & Brennan, S. E. (1991). Grounding in communication. In Resnick, L. B. & Levine, J. M. (eds.), *Perspectives on socially shared cognition*. American Psychological Association: Washington, DC, 127-149.
- Conway, M., Audia, S., Burnette, T., Cosgrove, D., & Christiansen, K. (2000). Alice: lessons learned from building a 3D system for novices. *CHI 2000 Conference on Human Factors in Computing Systems*, 486-493.
- Covi, L. M., Olsen, J. S., Rocco, E., Miller, W. J., & Allie, P. (1998). A room of your own: What do we learn about teamwork from assessing teams in dedicated project rooms? *Co'Build 1998 Conference on Cooperative Buildings*.
- Cunningham, W. (2005). Wiki Wiki Web. <http://c2.com/cgi/wiki>

- Cutrell, E., Czerwinski, M., & Horvitz, E. (2001). Notification, Disruption and Memory: Effects of Messaging Interruptions on Memory and Performance. *Interact 2001 IFIP Conference on Human Computer Interaction*, 263-269.
- Cybozu. (2005). Share360. <http://cybozu.com>
- Davenport, T. H., & Beck, J. C. (2002). *The attention economy: Understanding the new currency of business*. Harvard Business School Press.
- Dobrowolski, T. (2005). MoonEdit. <http://me.sphere.pl/indexen.htm>
- Documentum. (2005). eRoom. <http://www.documentum.com/>
- Dourish, P., & Bellotti, V. (1992). Awareness and coordination in shared workspaces. *Proceedings of CSCW 1992*, 107-114.
- e-motional.com. (2005). Active Bulletin Screen Saver. <http://www.e-motional.com/activebulletin.htm>
- Everitt, K. M., Klemmer, S. R., Lee, R., & Landay, J. A. (2003). Two worlds apart: Bridging the gap between physical and virtual media for distributed design collaboration. *CHI 2003 Conference on Human Factors in Computing Systems*, 553-560.
- Facilitate.com. (2005). Facilitate.com. <http://facilitate.com/>
- Fass, A. M., Forlizzi, J., & Pausch, R. (2002). MessyDesk and MessyBoard: Two Designs Inspired By the Goal of Improving Human Memory. *DIS 2002 Designing Interactive Systems*, 303-311.
- Fass, A. M., & Pausch, R. (2002). Adding Scripting to a Public Bulletin Board. *UIST 2002 Symposium on User Interface Software and Technology (poster abstract)*.
- Fitzpatrick, G., Kaplan, S., Mansfield, T., David, A., & Segall, B. (2002). Supporting public availability and accessibility with Elvin: Experiences and reflections. *Computer Supported Cooperative Work*, 11(3), 447-474.
- Fogarty, J., Hudson, S. E., & Lai, J. (2004). Examining the robustness of sensor-based statistical models of human interruptibility. *CHI 2004 Conference on Human Factors in Computing Systems*, 207-214.
- Frohlich, D. M., & Kraut, R. E. (2003). The social context of home computing. In Harper, R. (ed.), *Inside the smart home*. Springer-Verlag: London, 127-162.
- Fuchs, L., Pankoke-Babatz, U., & Prinz, W. (1995). Supporting cooperative awareness with local event mechanisms: The GroupDesk system. *ECSCW 1995 European Conference of Computer-supported Cooperative Work*, 247-262.

- Gallo, P. S., & McClintock, C. G. (1965). Cooperative and Competitive Behavior in Mixed-Motive Games. *Journal of Conflict Resolution*, 9(1), 68-78.
- Gamma, E. (1995). *Design patterns: elements of reusable object-oriented software*. Addison-Wesley: Reading, Mass.
- Google. (2005a). AdWords. <https://adwords.google.com/select/>
- Google. (2005b). Google Image Search. <http://www.google.com>
- Grasso, A., Roulland, F., Snowdon, D., Muehlenbrock, M., Mesenzani, M., & Morici, R. (2002). Supporting Informal Communication across Local and Distributed Communities. *CSCW 2002 Workshop: Public, Community and Situated Displays*.
- Greenberg, S., Bohnet, R., Roseman, M., & Webster, D. (1992). Human and technical factors of distributed group drawing tools. *Interacting with Computers*, 4(1), 364-392.
- Greenberg, S., & Roseman, M. (1998). *Using a Room Metaphor to Ease Transitions in Groupware*. Research report 98/611/02. Department of Computer Science, University of Calgary: Calgary, Alberta, Canada.
- Greenberg, S., & Rounding, M. (2001). The Notification Collage: Posting information to public and personal displays. *CHI 2001 Conference on Human Factors in Computing Systems*, 514-521.
- Groove Networks. (2005). Groove Workspace. <http://www.groove.net/>
- Groupboard. (2005). Groupboard. <http://www.groupboard.com/>
- Grudin, J. (1994). Groupware and social dynamics: eight challenges for developers. *Communications of the ACM*, 37(1), 92-105.
- Gutwin, C., Stark, G., & Greenberg, S. (1995). Support for Workspace Awareness in Educational Groupware. *Pro ACM Conference on Computer Supported Collaborative Learning*, 147-156.
- HelpMeeting.com. (2005). HelpMeeting. <http://www.helpmeeting.com/>
- Herbsleb, J. D., Atkins, D. L., Boyer, D. G., Handel, M., & Finholt, T. A. (2002). Introducing instant messaging and chat in the workplace. *CHI 2002 Conference on Human Factors in Computing Systems*, 171-178.
- Herbsleb, J. D., & Mockus, A. (2003). An Empirical Study of Speed and Communication in Globally-Distributed Software Development. *IEEE Transactions on Software Engineering*, 29(6).

- Hindus, D., Mainwaring, S. D., Leduc, N., Hagström, A. E., & Bayley, O. (2001). Casablanca: designing social communication devices for the home. *CHI 2001 Conference on Human Factors in Computing Systems*, 325-332.
- Horvitz, E., Kadie, C., Paek, T., & Hovel, D. (2003). Models of attention in computing and communication: from principles to applications. *Communications of the ACM*, 46(3), 52-59.
- Houde, S., Bellamy, R., & Leahy, L. (1998). In search of design principles for tools and practices to support communication within a learning community. *SIGCHI Bulletin*, 30(2).
- Huang, E. M., & Mynatt, E. D. (2003). Semi-public displays for small, co-located groups. *CHI 2003 Conference on Human Factors in Computing Systems*, 49-56.
- Intranets.com. (2005). Intranet Suite. <http://www.intranets.com>
- Izadi, S., Brignull, H., Rodden, T., Rogers, Y., & Underwood, M. (2003). Dynamo: A public interactive surface supporting the cooperative sharing and exchange of media. *UIST 2003 Symposium on User Interface Software and Technology*, 159-168.
- Jackson, M., & Johansson, C. (1997). Real time discrete event simulation of a PCB production system for operational support. *29th Conference on Winter Simulation*, 832-837.
- JDH Technologies. (2005). Web-4M. <http://www.jdhtech.com/pages/business.html>
- Kidd, A. (1994). The marks are on the knowledge worker. *CHI 1994 Conference on Human Factors in Computing Systems*, 186-191.
- Kiesler, S., Zubrow, D., & Moses, A. M. (1985). Affect in computer-Mediated Communication: An Experiment in Synchronous Terminal-to-Terminal Discussion. *Human-Computer Interaction*, 1(1), 77-104.
- Klemmer, S. R., Newman, M. W., Farrell, R., Bilezikjian, M., & Landay, J. A. (2001). The Designers' Outpost: A tangible interface for collaborative web site design. *UIST 2001 Symposium on User Interface Software and Technology*, 1-10.
- Klemmer, S. R., Thomsen, M., Phelps-Goodman, E., Lee, R., & Landay, J. A. (2002). Where do web sites come from? Capturing and interacting with design history. *CHI 2002 Conference on Human Factors in Computing Systems*, 1-8.
- Klößner, K. (2000). BSCW - educational servers and services on the WWW. *C4-ICDE Conference on Distance Education and Open Learning*.
- Kraut, R. E., Fish, R., Root, R., & Chalfonte, B. (1990). Informal communication in organizations: Form, function, and technology. *Human reactions to technology: Claremont symposium on applied social psychology*, 145-199.

- Microsoft. (2005a). NetMeeting. <http://www.microsoft.com/windows/netmeeting/>
- Microsoft. (2005b). Outlook. <http://www.microsoft.com/office/outlook/>
- Microsoft. (2005c). SharePoint. <http://www.microsoft.com/sharepoint/>
- Moran, T. P., McCall, K., van Melle, B., Pederson, E. R., & Halasz, F. G. (1995). Some design principles for sharing in Tivoli, a whiteboard meeting-support tool. In Greenberg, S., Hayne, S., & Rada, R. (eds.), *Groupware for Real-Time Drawing*. McGraw-Hill Book Company Europe: Berkshire, England, 24-36.
- Moran, T. P., Saund, E., Melle, W. V., Gujar, A. U., Fishkin, K. P., & Harrison, B. L. (1999). Design and technology for Collaborage: collaborative collages of information on physical walls, *UIST 1999 Symposium on User Interface Software and Technology*. ACM Press, 197-206.
- Morkes, J., Kernal, H. K., & Nass, C. (1998). Humor in task-oriented computer-mediated communication and human-computer interaction. *CHI 1998 Conference on Human Factors in Computing Systems*, 215-216.
- MyCorkboard. (2005). MyCorkboard. <http://www.mycorkboard.com/>
- Myers, B. A. (1985). The importance of percent-done progress indicators for computer-human interfaces. *CHI 1985 Conference on Human Factors in Computing Systems*, 11-17.
- Mynatt, E. D. (1999). The writing on the wall. *Interact 1999 IFIP Conference on Human-Computer Interaction*.
- Nässla, H., & Carr, D. (2003). Investigating Intra-Family Communication Using Photo Diaries. *HCI International 2003, 10th International Conference on Human-Computer Interaction*, 984-988.
- Netstat. (2005). Nedstat Basic. <http://www.nedstatbasic.net>
- netomat, Inc. (2005). netomat. <http://www.netomat.net>
- Netpresenter. (2005). Netpresenter. <http://www.netpresenter.com>
- Netscape. (2005). Conference. <http://wp.netscape.com/communicator/conference/v4.0/>
- Nichols, J., Wobbrock, J. O., Gergle, D., & Forlizzi, J. (2002). Mediator and medium: Doors as interruption gateways and aesthetic displays. *DIS 2002 Designing Interactive Systems*, 379-386.
- Oceanview Consultancy. (2005). Marquee Plus Version 2.4. <http://www.oview.co.uk/marquee.htm>

- O'Conaill, B., & Frohlich, D. (1995). Timespace in the workplace: dealing with interruptions. *CHI 1995 Conference on Human Factors in Computing Systems*, 262-263.
- O'Hara, K., & Perry, M. (2002). Using a situated display appliance for socially mediated space management. *CSCW 2002 Workshop: Public, Community and Situated Displays*.
- OpenText Corporation. (2005). Livelink. <http://www.opentext.com/livelink/>
- Orlikowski, W. J. (1992). Learning from Notes: Organizational issues in groupware implementation. *Proceedings of CSCW 1992*, 362-369.
- Pausch, R. (1988). Adding Input and Output to the Transactional Model. Doctoral Dissertation. Carnegie Mellon University, Pittsburgh, PA.
- Poole, M. S., & DeSanctis, G. (1990). Understanding the Use of Group Decision Support Systems: The Theory of Adaptive Structuration. In Fulk, J. & Steinfield, C. (eds.), *Organizations and Communication Technology*. Sage: Thousand Oaks, CA, USA, 175-195.
- Prinz, W. (1999). NESSIE: An awareness environment for cooperative settings, *ECSCW 1999 European Conference on Computer-Supported Cooperative Work*. Kluwer Academic Publishers: Copenhagen, Denmark, 391-410.
- Rick, J., Guzdial, M., Carroll, K., Hollaway-Attaway, L., & Walker, B. (2002). Collaborative Learning at Low Cost: CoWeb Use in English Composition. *CSCL 2002 Computer-Supported Collaborative Learning*.
- Rogers, Y., & Brignull, H. (2002). Subtle ice-breaking: encouraging socializing and interaction around a large public display. *CSCW 2002 Workshop: Public, Community and Situated Displays*.
- Roseman, M., & Greenberg, S. (1996). TeamRooms: network places for collaboration. *Proceedings of CSCW 1996*, 325-333.
- RSS-DEV Working Group. (2005). RDF Site Summary (RSS) 1.0. <http://web.resource.org/rss/1.0/spec#s1>
- RTZ Software. (2005). The Virtual Meeting. <http://www.rtz.com/products/TheVirtualMeeting/>
- Russell, D. M. (2002). Large Interactive Public Displays: Use patterns, support patterns, community patterns. *CSCW 2002 Workshop: Public, Community and Situated Displays*.
- Russell, D. M., Trimble, J., & Wales, R. (2002). Two paths from the same place: Task driven and human-centered evolution of a group information surface. *Make IT Easy Conference*.

- Schmidt, K. (2002). The problem with 'awareness'. *Computer Supported Cooperative Work (CSCW)*, 11, 285-298.
- Sega. (2005). Zero Wing. <http://www.toaplan.com/zerowing/>
- Shiozawa, H., Okada, K., & Matsushita, Y. (1999). Perspective layered visualization of collaborative workspaces. *SIGGROUP 1999 Conference on Supporting Group Work*, 71-80.
- Simon, H. A. (1971). Designing Organizations for an Information-Rich World. In Greenberger, M. (ed.), *Computers, Communications, and the Public Interest*. Johns Hopkins Press: Baltimore and London, 37-72.
- Smart Technologies. (2005). Bridgit. <http://www.smarttech.com/products/bridgit/>
- Smith, R. B., Hixon, R., & Horan, B. (1998). Supporting flexible roles in a shared space. *Proceedings of CSCW 1998*, 197-206.
- Smith, S. M., & Vela, E. (2001). Environmental context-dependent memory: A review and meta-analysis. *Psychonomic Bulletin and Review*, 8(2), 203-220.
- Spoon, L., & Guzdial, M. (1999). MuSwikis: A Graphical Collaboration System. *CSCW 1999 Computer Support for Collaborative Learning*.
- Spreitzer, M., & Theimer, M. (1993). Providing location information in a ubiquitous computing environment (panel session). *ACM Symposium on Operating Systems Principles*, 270-283.
- Stefik, M., Bobrow, D. G., Foster, G., Lanning, S., & Tatar, D. (1987). WYSIWIS revisited: early experiences with multiuser interfaces. *ACM Transactions on Information Systems (TOIS)*, 5(2), 147-167.
- Sun Microsystems. (2005a). Abstract Window Toolkit. <http://java.sun.com/products/jdk/awt/>
- Sun Microsystems. (2005b). Java Web Start. <http://java.sun.com/products/javawebstart/>
- Sun Microsystems. (2005c). JFC/Swing. <http://java.sun.com/products/jfc/index.jsp>
- Tan, D. S., Stefanucci, J. K., Proffitt, D. R., & Pausch, R. (2001). The Infocockpit: Providing location and place to aid human memory. *Proceedings of the PUI 2001 Workshop on Perceptive User Interfaces*, 1-4.
- Tang, C., McEwan, G., & Greenberg, S. (2003). A taxonomy of tasks and visualizations for casual interaction of multimedia histories. *Graphics Interface*.
- Thruport. (2005). HotOffice. <http://www.hotoffice.com/>

- Trimble, J., Wales, R., & Gossweiler, R. (2002). NASA position paper for the CSCW 2002 workshop on public, community and situated displays: MERBoard. *CSCW 2002 Workshop: Public, Community and Situated Displays*.
- van Rossum, G., & de Boer, J. (1991). Interactively testing remote servers using the Python programming language. *CWI Quarterly*, 4(4), 283-303.
- van Solingen, R., Berghout, E., & van Latum, F. (1998). Interrupts: just a minute never is. *IEEE Software*, 15(5), 97-103.
- Walker, D. (2004, November 9, 2004). *Why are meetings so boring?* (UK Edition), [World Wide Web]. BBC News Magazine. Available: <http://news.bbc.co.uk/1/hi/magazine/3993483.stm>.
- WebEx. (2005). WebEx. <http://www.webex.com>
- Welch, G., Fuchs, H., Raskar, R., Towles, H., & Brown, M. S. (2000). Projected imagery in your "office of the future". 20(4), 62-67.
- Wheeler, B. C., Dennis, A. R., & Press, L. I. (1999). Groupware comes to the Internet: charting a new world. *SIGMIS Database*, 30(3-4), 8-21.
- Whittaker, S., & Schwarz, H. (1995). Back to the future: pen and paper technology supports complex group coordination, *CHI 1995 Conference on Human Factors in Computing Systems*. ACM Press/Addison-Wesley Publishing Co.: Denver, CO, 495-502.
- Wolf, C. G., Rhyne, J. R., & Briggs, L. K. (1995). Using a Shared Pen-Based Tool for Meeting Support. In Greenberg, S., Hayne, S., & Rada, R. (eds.), *Groupware for Real-Time Drawing*. McGraw-Hill Book Company Europe: Berkshire, England, 81-95.
- Yahoo. (2005). Yahoo Groups. <http://groups.yahoo.com/>
- Yahoo! (2005). Yahoo! Mail. <http://mail.yahoo.com>
- Yasumatsu. (2005). Kazuki Yasumatsu's Foundation Classes (KFC). <http://openlab.jp/kyasu/>
- Zhao, Q. A. (2001). Opportunistic Interfaces for Promoting Community Awareness. Doctoral Dissertation. Georgia Institute of Technology.