UNIVERSITY OF CALIFORNIA,
IRVINE

Social and Temporal Structures in Everyday Collaboration

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Information and Computer Science

by

Danyel Aharon Fisher

Dissertation Committee:
Professor Paul Dourish, Chair
Professor Bonnie Nardi
Professor David Redmiles

2004

The dissertation of Danyel Aharon Fisher
is approved and is acceptable in quality
and form for publication on microfilm

Committee Chair

University of California, Irvine
2004

DEDICATION

TO MY FAMILY, who carefully balanced carrot and stick

TO MY FRIENDS, who have supported me as I've changed ideas and directions

TO MY MENTORS, who saw this moment more clearly than I ever did

Um, social networks …
(mumble) and timelines …
pretty pictures,
and (mumble) email …
really, it's pretty cool.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS		X
CURRICULUM VITAE		XII
ABSTRACT OF THE DISSERTATION		XV
CHAPTER 1.	INTRODUCTION	1
1.1.	Personal Collaboration; Group Collaboration	1
1.1.1.	Email as Habitat	3
1.1.2.	Scheduling and Temporal Patterns	3
1.1.3.	Social Navigation	4
1.1.4.	Categorization and Coding	5
1.1.5.	Supporting Implicit Collaboration	6
1.1.6.	Scenarios of Failure	7
1.1.7.	Challenges in Tracking Social Information	8
1.2.	Technologies that Make People Visible	9
1.3.	Social and Temporal Structures	10
1.4.	Approach through Patterns and Structure	12
1.5.	Articulating the Approach	12
1.6.	Research Questions	14
1.6.1.	Hypotheses	15
1.6.2.	Contribution	16
1.7.	Outline	18
CHAPTER 2.	LITERATURE OF TECHNIQUES AND TECHNOLOGIES	20
2.1.	Social Network Analysis	20
2.1.1.	Social Networks for End Users	21
2.1.2.	A Framework for Discussing Network Research	25
2.1.3.	Egocentric Analysis within CSCW	26
2.1.3.1.	Contact Management	27
2.1.3.2.	Awareness and Presence	30
2.1.3.3.	Social Navigation and Collaborative Filtering	34
2.1.4.	Mining Social Networks	36
2.1.4.1.	Expertise Location	37
2.1.4.2.	Reputation Systems	40
2.1.5.	Social Visualizations	43
2.1.5.1.	Email Visualizations	44
2.1.5.2.	Visualizing Interactive Spaces	45
2.1.6.	Social Networks in the Framework	50
2.2.	Temporality in the Social Workscape	52
2.2.1.	Cross-Disciplinary Temporality	53
2.2.2.	Time in CSCW Research	57
2.2.3.	Temporality in the Interface	58

2.2.4.	Final Thoughts on Time	63
2.3.	Conclusion	63
CHAPTER 3.	VISUALIZING COMMUNICATION	65
3.1.	Designing a System	65
3.1.1.	Choosing a Corpus	65
3.1.2.	Visualizing Mail	68
3.1.3.	Building Social Network Diagrams	69
3.1.4.	Viewing Temporality	73
3.2.	The Soylent Implementation: Automated Traces	76
3.2.1.	Collecting a corpus	77
3.2.2.	The Soylent Data Collector	78
3.2.3.	The Soylent Visualizations	80
3.2.4.	Association Net	81
3.2.5.	User interface	82
3.2.5.1.	Data Filtering Tools	82
3.2.5.2.	Handling the Display Space	83
3.2.6.	Temporal Maps: Time Display and Top Ten List	84
3.2.7.	Other Soylent Views	86
3.3.	Challenges in Soylent	89
3.3.1.	Privacy Issues	89
3.3.2.	Design Trade-offs	90
3.4.	Conclusion	93
CHAPTER 4.	PATTERNS IN COMMUNICATION	94
4.1.	The User Engagement	94
4.1.1.	User Population	95
4.1.2.	Observing Respondents	96
4.1.3.	Looking at Soylent	97
4.1.4.	Redesigning Visualizations	98
4.2.	Understanding Large Datasets With Patterns	99
4.3.	Using Formal Social Network Analysis	100
4.4.	Interpreting Patterns	101
4.5.	Pattern Catalog	103
4.5.1.	About the Catalog	104
	Pattern: Disconnected Clusters	107
	Pattern: Onion	110
	Pattern: Clique	113
	Pattern: Butterfly	114
	Pattern: Nexus	116
	Pattern: Gradual Shifting Positions	119
	Pattern: Bursts of Activity	122
	Pattern: Changing Roles within a Network	124
	Pattern: Normalcy	126
	Pattern: Unusual Events	129
4.6.	The Efficacy of Patterns	130

4.7.	Discussion	131
4.8.	Conclusion	134
CHAPTER 5.	APPLYING SOCIAL INFORMATION	135
5.1.	Design implications from Patterns	135
5.2.	Aspects of Patterns	137
5.3.	Prototype Sketch: a Social Proxy	138
5.4.	TMA as Awareness	142
5.5.	Enhancing Email with EE4P	147
5.5.1.	Recipient prediction	148
5.5.2.	Message awareness	150
5.5.3.	Enhanced Address Book	151
5.6.	Discussion	152
CHAPTER 6.	THE FUTURE OF THE WORKSPACE	153
6.1.	ContactMap	153
6.2.	Comparing Soylent and ContactMap	157
6.2.1.	Distinguishing Soylent from ContactMap	159
6.3.	The Integrated Workscape	161
6.4.	Building an Infrastructure	162
6.5.	Summary	165
CHAPTER 7.	CONCLUSIONS	166
7.1.	Overview	166
7.2.	Research Conclusions	168
7.3.	Limitations and Future Work	170
7.3.1.	Speed as a Limiting Factor	170
7.3.2.	Generalized Groups	171
7.3.3.	File System Integration	171
7.3.4.	Text Processing	171
7.4.	Conclusion	172
BIBLIOGRAPHY		173
WEB REFERENCES		190
APPENDIX 1.	DATABASE FIELDS IN SOYLENT	191
APPENDIX 2.	HANDLING ALIASES AND CLEANING EMAIL	194
APPENDIX 3.	THE SOYLENT-TELLMEABOUT API	196

LIST OF FIGURES

Figure 2-1.	Babble interface.	33
Figure 2-2.	ReferralWeb.	39
Figure 2-3.	Social Network Fragments.	45
Figure 2-4.	PeopleGarden.	46
Figure 2-5.	VisualWho	47
Figure 2-6.	Conversation Map.	48
Figure 2-7.	Crosspost visualization from Netscan.	49
Figure 2-8.	ClanGraph.	50
Figure 2-9.	Time as ordering in the file system.	59
Figure 3-1.	The Soylent egocentric networks.	70
Figure 3-2.	Showing the ego.	71
Figure 3-3.	Soylent's Association Net view.	72
Figure 3-4.	Time Display.	74
Figure 3-5.	Soylent Architecture	79
Figure 3-6.	The Top Ten List.	84
Figure 3-7.	The Group Display.	86
Figure 3-8.	TextArc-based view.	88
Figure 4-1.	Disconnected Clusters	107
Figure 4-2.	The Onion pattern.	111
Figure 4-3.	The core subgroup within the Onion.	112
Figure 4-4.	The Butterfly pattern.	114
Figure 4-5.	The Nexus pattern.	116

Figure 4-6.	A second, different Nexus.	117
Figure 4-7.	Top ten list showing variations from travel.	119
Figure 4-8.	Top ten list showing a new project.	120
Figure 4-9.	Timeline showing interaction on a publication project.	120
Figure 4-10.	Timeline showing interaction in a romantic relationship.	121
Figure 4-11.	Timeline showing interaction with a professor.	122
Figure 4-12.	Timeline showing conference paper submission.	123
Figure 4-13.	A group changing over time	124
Figure 4-14.	Changing involvement.	125
Figure 4-15.	Normalcy	127
Figure 4-16.	Exceptions to normal schedules.	129
Figure 5-1.	The Workspace Social Proxy.	141
Figure 5-2.	TellMeAbout visual display.	145
Figure 5-3.	Enhanced Email for People (EE4P) Recipient Prediction	149
Figure 5-4.	EE4P main screen.	151
Figure 5-5.	EE4P Address Book Entry Maintenance	152
Figure 6-1.	Contact Map.	154

LIST OF TABLES

Table 2-1.	A Framework for Discussing Social Network Research	25
Table 2-2.	Attributes of Social Network Systems Within the Framework	51
Table 3-1.	Message Census for Three Sample Users	77
Table 3-2.	Data stored by Soylent (and corresponding RFC 822 fields)	78
Table 4-1.	Pattern Type Summary	106
Table A-1.	Envelope information.	191
Table A-2.	Recipients and Participants	192
Table A-3.	Attachment table	192

ACKNOWLEDGEMENTS

This dissertation, like most, is the work of many hands. I have been privileged to work with a variety of able teachers and mentors throughout my career. I am honored by having both David Redmiles and Bonnie Nardi on my defense committee, and having had the additional insights of both Katie Faust and Andre van der Hoek on my advancement committee. Few dissertations are ever read more than once; I am grateful to have had early comments from Steve Abrams, Jack Muramatsu, and Jeremy Fisher.

My graduate career would have been very different were it not for several generations of mentors: Barry Wellman's ideas and direction were deeply formative early in. Were it not for his invitation to the "Communities" workshop at CHI 1999, social computing and I might never have met.

The various researchers at IBM Research / Cambridge (especially Paul Moody and David Millen) worked hard to encourage my progress, and funded two internships, one specifically for some of the work of this dissertation.

I could not have made it through my dissertation without the constant support of my friends: too many to name throughout my graduate career, but unquestionably Chuck Smith, Joshua O'Madadhain, Katherine Farber, Michelle Moore, and Talana Orzechowski.

Perhaps most importantly, Paul Dourish has, at the various and appropriate times, aided, abetted, directed, and disavowed all knowledge of me. I am grateful for all of them.

Social network visualizations were generated by the JUNG visualization system; thanks to Kevin Dodd and the ICS 125 EE4P development team for the EE4P prototype described Chapter 5.

Portions of chapters 3, 4, and 5 are updated and extended material that originally appeared in *Fisher, D. and Dourish, P. Social and Temporal Structures in Everyday Collaboration, in Proceedings of the 2004 ACM Conference on Human Factors in Computing Systems.* This material appears with permission of the ACM press. Portions of chapters 3, 4, and 5 are updated and extended material that are scheduled to appear in *Fisher, D. and Dourish, P. Evolving Email into the Social Workscape* in an upcoming issue of the Human-Computer Interaction Journal. Portions of chapter 6 are a reprint of material that is scheduled to appear in Fisher, D. and Nardi, B. *Soylent and ContactMap: Tools for Constructing the Social Workscape* within the upcoming MIT Press volume edited by M. Czerwinski and V. Kaptalenin. *Beyond the Desktop Metaphor: Toward Integrated Digital Work Environments.* Printed by permission.

My co-authors for these publications, Paul Dourish and Bonnie Nardi, directed and supervised research which forms the basis for this dissertation. Some of the text within, and much of the inspiration, is theirs.

Chapter 2 of this dissertation contains a several illustrative images clarifying previous network communication. I thank Tom Erickson for figure 2-1 and Henry Kautz for figure 2-2. Figure 2-3 first appeared in *boyd, d. (2002). Faceted Id/entity: Managing representation in a digital world. Media Lab, School of Architecture and Planning. Masters thesis from Massachusetts Institute of Technology* and is reprinted with permission. Figure 2-4 is reprinted from *Xiong, R. and J. Donath (1999). PeopleGarden: Creating Data Portraits for Users. Proceedings of the 12th annual ACM symposium on User interface software and technology, Asheville, NC., ACM*. Printed by permission. Figure 2-5 is reprinted from *Donath, J. (1995). Visual Who: animating the affinities and activities of an electronic community. Proceedings of the third ACM international conference on Multimedia, San Francisco, California*. Printed by permission. Figure 2-6 and Figure 2-7 reprinted from *Lueg, C. and Fisher, D, eds. (2003). From Usenet to Cowebs: Interacting with Social Information Spaces. London: Springer*. Printed by permission. Figure 2-8 is reprinted from *Terveen, L., Hill, W., Amento, B. (1999). "Constructing, Organizing, and Visualizing Collections of Topically Related Web Resources." ACM Transactions on Computer-Human Interaction* ***6****(1): 67-94*. Printed by permission. Figure 2-9 is a screen shot from Microsoft Windows XP (a trademark of the Microsoft Corporation). Screen shot(s) reprinted by permission from Microsoft Corporation.

Financial support for this work was supported in part by the National Science Foundation under awards IIS-0133749, IIS-0205724 and IIS-0326105, by IBM Research, and by Intel Corporation.

CURRICULUM VITAE

Danyel Aharon Fisher

EDUCATION

PhD, August 2004, Information and Computer Science (ICS)
University of California, Irvine (UCI)
Advisor: Paul Dourish

MS, June 2000, Computer Science
University of California, Berkeley

BS, June 1997, Computer Science
Washington University in St. Louis

BA, June 1997, Philosophy,
Washington University in St. Louis

RESEARCH INTERESTS

Computer-mediated communication; Computer-supported Cooperative Work (CSCW); human-computer interaction (HCI); temporality; information visualization; social analysis of technology and of interaction.

RESEARCH EXPERIENCE

2000–2004. Graduate Student Researcher, ICS.

Carried out research on the development, adoption, and use of workplace collaboration tools in a variety of settings, ranging from a large distributed development and manufacturing corporation to a small academic program.

Fall 2002. Research Associate, IBM Watson Research/Cambridge. Collaborative User Experience group.

Developed and expanded "Soylent" research system and visualizations for understanding users' email collaboration patterns. Interviewed employees about their mail sending and receiving patterns. Applied results to continuing "Re:mail" project.

Summer 2000. Research Associate, IBM Lotus Research

Designed and implemented a distributed Java applet for online survey and mail analysis. Studied email use patterns, including folders, threads, and social network participation.

Summer 1997. Research Associate, IBM Research/Almaden. USER group.

Designed and implemented WebPlaces, a web-based interface for electronic social interaction, based on IBM's Web Browser Intermediary proxy solution. WebPlaces follows users' online surfing to understand their interests; allows users researching similar topics to communicate.

PEER-REVIEWED PAPERS

M. Wattenberg and D. Fisher. "Analyzing Scale Perceptual Organization in Information Graphics." Information Visualization Journal. Palgrave. In press.

D. Fisher and P. Dourish. "Social and Temporal Structures in Everyday Collaboration." ACM Conference on Human-Computer Interaction (CHI) 2004, Vienna. April, 2004.

M. Wattenberg and D. Fisher. "A Model of Multi-Scale Perceptual Organization in Information Graphics." IEEE Symposium on Information Visualization (InfoVis) 2001, Seattle. October, 2003.

K.-P. Yee, D. Fisher, R. Dhamija, M. Hearst. "Animated Exploration of Dynamic Graphs with Radial Layout." IEEE Symposium on Information Visualization (InfoVis) 2001, San Diego. October, 2001.

G. Mark, S. Poltrock, and D. Fisher. "Communication and Technical Diffusion Across Geographic Distance." IEEE Professional Communication Conference 2001, Phoenix. October, 2001.

D. Fisher, K. Hildrum, J. Hong, M. Newman, M. Thomas, and R. Vudoc. "SWAMI: A Framework for Collaborative Filtering Algorithm Development and Evaluation." Poster at ACM SIGIR 2000, Athens, Greece. July, 2000.

BOOK EDITOR

D. Fisher and C. Lueg. From Usenet to Cowebs: Usenet News and Online Information Spaces. London: Springer. 2003.

BOOK CHAPTERS

D. Fisher and B. Nardi. "Soylent and ContactMap: Tools for Constructing the Social Workscape." To appear in V. Kaptalenin and M. Czerwinski, Integrated Digital Work Environments. Cambridge: MIT Press, 2004.

P. Dourish, M. Reddy, and D. Fisher. "Temporality and Structure in Collaborative Work." In Heilesen (ed.), Digital Presence: Knowledge and Design in New Media (Det digitale naervaer: Viden og design i nye medier). Roskilde University. 2004

D. Fisher. "Communications Technologies." In K. Christiansen and D. Levinson, eds. Encyclopedia of Community: From the Village to the Virtual World. Volume 1, pp. 215-218. Sage Publications. 2003

D. Fisher. "Introduction: Studying Social Information Spaces" In C. Lueg, D. Fisher (eds). From Usenet to Cowebs: Usenet News and Online Information Spaces. London: Springer. 2003.

D. Fisher and C. Lueg. “Appendix: Studying Online Newsgroups” In C. Lueg, D. Fisher (eds). From Usenet to Cowebs: Usenet News and Online Information Spaces. London: Springer. 2003.

SOFTWARE RELEASED

JUNG (Java Universal Network and Graph toolkit; http://jung.sourceforge.net)

A byproduct of research on social networks, this library attempts to fill the need for an extensible, analytical tool for online network analysis. It functions like network analysis tools (PAJEK, UCINET), but allows a Java developer to implement additional algorithms, modify graphs on the fly, and derive information from a variety of data sources.

ORGANIZED WORKSHOPS

D. Fisher, S. Farnham, and D. McDonald. CSCW (ACM Conference on Computer-Supported Collaborative Work) 2004. “Social Networks for Design and Analysis: Using Network Information in CSCW.”

D. Fisher and D. McDonald. ECSCW (European Conference on Computer-Supported Collaborative Work) 2003. “Moving from Analysis to Design: Social Networks in the CSCW Context.”

D. Fisher, A. Bruckman, T. Erickson, and C. Lueg. CSCW (ACM Conference on Computer-Supported Collaborative Work) 2000. “Dealing with Community Data.”

ABSTRACT OF THE DISSERTATION

Social and Temporal Structures in Everyday Collaboration

by

Danyel Aharon Fisher

Doctor of Philosophy in Information and Computer Science

University of California, Irvine, 2004

Professor Paul Dourish, Chair

Everyday work frequently involves coordinating and collaborating with others, but the structure of collaboration is largely invisible to conventional desktop applications. The information is not shown as a result of a traditional, but outdated, separation between computer applications for individual tasks and those that support collaboration. In my dissertation, I argue that this separation needs to be broken down by building systems that better acknowledge the roles and relationships of other people within computer systems.

A first step in this process is to understand how those relationships are enacted. I present “Soylent,” a tool to collect and visualize the social networks and temporal fluctuations implicit in a single user's email. Soylent user tests lead to a notion of patterns, recurrent structures of interaction that were common between different groups of users. These patterns, in turn, led to the design of two tools that help show user interaction histories and recent events, “TellMeAbout” and “Enhanced Email for People.”

These two tools are used to demonstrate the mechanisms of a workscape that is better connected to social roles and histories.

Chapter 1. Introduction

This dissertation shows that social networks and temporality can be used to provide meaningful, useful descriptions of the interconnections within groups of people. These descriptions allow the development of software that support the collaborative aspects of apparently individual work, by placing the single-user experience more explicitly in the wider social frame within which works takes place.

In this chapter, I will discuss how the task of situating work within a social context has traditionally been approached within the field of computer-supported collaborative work (CSCW). CSCW tends to make a distinction between collaborative work and individual work, and, as a field, tends to look more closely at the former. I will describe some of the symptoms of this distinction, and will suggest that the answer lies in breaking it down by providing systems with information about social and temporal structures. Last, I will propose research questions that will allow us to understand what these awareness systems can provide.

1.1. Personal Collaboration; Group Collaboration

The work we carry out using computer-based tools is highly collaborative; we write documents to send to others, collate presentations from pieces supplied by colleagues, and use spreadsheets to coordinate distributed activities.

The tools with which we conduct this work remain stubbornly isolated. Typical applications provide little or no access to the other people who are closely involved in our work, as audience or as collaborators. Word processors, for instance, operate in terms of documents and paragraphs but give us no access to the people whose words we are manipulating; similarly, software development systems separate source code from views of the team working on it. Even the tools for managing email – perhaps the daily computer-based task that most obviously involves other people – are largely browsers for a database of text objects. The interface obscures the social structures that are enacted and maintained by email communication.

This division is reflected in the disciplinary division between HCI and CSCW. Although CSCW research distinguishes between single-user and multi-user technologies, the boundary between individual and collaborative work is much less clear. many tasks carried out with ostensibly single-user tools are, on closer examination, collaborative in nature. Documents are frequently assembled out of pieces provided by others; presentations may be crafted as parts of larger projects, and are designed to suit the needs of both presenters and audiences; and spreadsheets are often used to coordinate collective activity.

CSCW researchers have long argued that human activities are shaped by, and respond to, aspects of the social, organization, and cultural contexts in which they occur (Suchman, 1987). This situated nature of everyday activities applies not only in "formally" collaborative settings such as joint project work and meetings, but also in apparently "individual" activities, which nonetheless arise in coordination with others. Even ostensibly single-user applications are used to conduct work that is essentially collaborative in nature – creating, sharing and integrating documents; coordinating projects; and developing software systems. Following past work (Fisher and Dourish, 2004), I refer to this as "everyday collaboration," in contrast to traditional collaboration settings such as process-based workflow or face-to-face meetings.

Users, of course, continue to reinvent and appropriate technology to accomplish the joint social and technical tasks that they accomplish. Here, I describe several different instances where the permeable distinction between individual work and collaboration has been reflected in the literature.

1.1.1. Email as Habitat

Duchenaut and Bellotti (2001) show that, for many users, e-mail is not simply a communication tool, but has become a space for varied organizational tasks, such as document storage, version control, scheduling, note keeping, activity tracking, and maintaining calendars. The ability to organize entries by author, by date, and to search them flexibly makes email a useful medium for storing diverse personal information. The use of electronic mail (rather than, for example, a spreadsheet, database, or file system) as a central repository of task and personal information draws our attention to the way that individual work on computer systems is enmeshed in a complex of collaborative activities, social relationships, and interpersonal dependencies.

1.1.2. Scheduling and Temporal Patterns

Scheduling and calendar systems also highlight the relationship between individual work and broader social and temporal patterns. Pipek and Wulf (1999) highlight the importance of shared interaction, and the effects of this individual work on group work. Discussing the transcriptionists of a German government agency, the investigators write, "As all the sections worked in the same rhythm, it created peaks in the work load of the secretaries causing a significant prolongation of processing time" (Pipek and Wulf, 1999). The work schedule of the secretary both was dependant on the rhythm of the incoming work, and constrained the timing for the following work.

Online calendars are visible tools for expressing temporal information. While they are often constructed individually, in order to allow users to track their own scheduled events, they can be used in a group context to share information about availability (Grudin, 1988). Balancing personal needs against group information can be delicate. Dourish et al. (1993) examined two different calendar systems at one organization, examining the tightly-coupled challenges of choosing how to publish and interpret information presented on the two different calendars. The distinctions was a subtle one: in one case, group members knew to trust the reliable paper calendar for the *times* of visiting talks – but to check the electronic calendar for the *titles* of the talks. In this paper, the interpretation of information in an "individual" tool (their view of the electronic calendar) depended on an assessment of the social character of the information.

1.1.3. Social Navigation

Even personally-oriented activities can produce and share information for others. Just as a worn path across the grass shows where many people have walked, social navigation technologies, such as collaborative filtering and recommender systems (Resnick & Varian, 1997) generate information about social actions implicitly, as a side effect of users' activities. These technologies capitalize on the social context of individual tasks, and relate individual actions to larger social trends by examining individual activities and highlighting some of the implications for other users.

1.1.4. Categorization and Coding

As illustrated by Dourish et al. (1999), sorting, filing and categorization involve implicit collaboration. Current users of a categorization scheme must understand the constraints of past designers, modify it to make sense to their current needs, and prepare for future users of the categorization system. In Dourish's study, a classification scheme used at a government agency must be repeatedly reinterpreted, expanded, and rearticulated to accommodate to new situations as they come. Every user of the scheme is aware that the document they file under this scheme must later be retrievable by future users: that is, the current user thinks of the system in terms of an implicit collaboration with unknown future users.

Bowker and Star (1999) have extensively examined the social embeddedness of systems of categorization. In their book, they examine the construction and use of an important medical classification scheme. They show the broad set of social factors that went into the construction of the classification scheme, ones based on seemingly-arbitrary and sometimes contradictory decisions. Unlike Dourish's example, the schema doesn't change on the fly: but the users of today are constrained by the system already in place.

Marshall (1997) also discusses the reuse of annotations in a university bookstore as an implicit, anonymous collaboration over time. While any particular annotator might not plan for their notes to be useful to anyone except themselves, future purchasers of their books seem distinctly aware of the value of old annotations and go out of their ways to find well-highlighted texts.

1.1.5. Supporting Implicit Collaboration

Each of these collaborative activities is embedded in both formal and informal social structures: Dourish's calendar users understood the calendars' maintainers and their work practices; Pipek and Wulf's transcriptionists were in the midst of several workflow pipelines. The effective and appropriate use of single-user tools in these settings relies on individuals' abilities to understand and relate their activity to broader patterns of the social and temporal organization of work. This is typically resolved on a task-by-task basis, adding collaborative features one-at-a-time to calendars, word processors, and file systems. This approach, though, suffers from three drawbacks. First, it has to be implemented anew for each application; second, it leads to inconsistency in the ways that different applications model and present social context; and third, it fails to exploit patterns in cross-application operation.

Instead, this dissertation examines the larger patterns of collective interaction. Moving beyond the scope of particular applications, I am interested in identifying broader connections between individual and collective activity.

1.1.6. Scenarios of Failure

These researchers have pointed to places where individual and group work intersect. The next step is to examine user needs to see how these intersections align with real scenarios in which a user attempts to handle their interactions with others.

- An editor tries to call up the proposal that he'd prepared a few months ago and had emailed to a friend, but it's tucked away somewhere on his hard disk. He locates it by a search on his inbox, then skims through a couple of alternatives before saving the proposal, for the third or fourth time, to disk.
- An author of an article has sent out preview copies to a series of his colleagues. A week later, he remembers some other people – has he sent them previews? What responses has he gotten to his notes so far? Did anyone send him edits?
- A manager, during his end-of-week review, once again spends an hour skimming over the email from the last week, trying to remember full set of projects he'd been involved in. The week's big crisis has masked a number of prior accomplishments, and so his report looks gloomier than it need be.
- A team member, trying to track down a decision made several months back, can't quite remember who was involved in the team at the time. Was that when the consultant was involved? Do they need to call in external help again? It takes him some time to dig through the mass of old messages and documents to figure out who was involved.
- A student has finished finals and is preparing for her job search. Knowing it's

a good time for social network maintenance, she starts to catch up on her old mail. Unfortunately, she doesn't remember that she's fallen out of touch with an old friend, now a recruiter for an organization in her field.

1.1.7. Challenges in Tracking Social Information

Each of those problems has special purpose solutions. The editor needs a better schema for saving and storing files. The author might want to maintain a spreadsheet with the attached notes and changes to help guide him through the edit cycle. The team member might want to keep better track of group decisions. And perhaps the student needs to start highlighting entries in a rolodex.

The common thread between these issues is that they all involve the ties between people, events, and time in collaborative systems. It can be hard to manage the complex aspects of interpersonal information, especially as it relates to a variety of groups of people. Nardi et al.(2002a) interviewed workers about how they manage and interact with people in their work. They found that study participants were careful managers of their personal social networks, perpetually and carefully aware of the ways in which they interacted with others.

Today, much of that work is handled within the email inbox. Whittaker and Sidner (1996) and Ducheneaut and Bellotti (2001) reported that email is used not only for its intended purpose of asynchronous communication, but for a broad variety of tasks of handling social information. A general-purpose solution is desirable; the ability to reuse social information throughout the system would address many of these difficulties.

The deeper claim of this research, however, is that all these problems are actually fundamentally related. Rather than considering this set of specialized answers – one to handle each case – I propose an approach that can unify and make progress on all of these sorts of problems.

1.2. Technologies that Make People Visible

I propose that an effective solution is not to replace single-user applications with groupware tools, nor to bolt collaborative functions onto the ends of single-user tools. Rather, my solution is to reveal the collective activity that is already being carried out through those tools. I want to help people coordinate their work by providing them with ways to see how their work is connected to that of their colleagues. In this dissertation, then, I explore the potential for using single-user tools as *technologies supporting awareness*.

Awareness is a widely noted aspect of collaborative practice. Much research into collaboration in practice has shown how the explicit, task-focused aspects of activity are complemented by and coordinated through passive mutual monitoring which provides people with an ongoing, informal awareness of each other's activity. Heath and Luff (1992), for example, showed how operators in London Underground control rooms delicately coordinated their separate activities by informally monitoring each others' actions and arranging their own work to match. Others have noted similar processes at work in electronically mediated collaboration (Schmidt, 2002). They allow people to coordinate their own work with that of others. Can support for informal awareness of everyday collaboration be incorporated into conventional applications?

1.3. Social and Temporal Structures

It In order to build support this awareness, we must have some information about the how to monitor and understand the underlying interaction. My approach to understanding this information is to exploit recurrent structures that relate the details of specific activity to broader patterns. In particular, I investigate two sorts of structures. First, *social structures* describe the patterns of contact and collaboration that emerge between people. They relate individuals to groups and collaborative activities. Second, *temporal structures* describe how patterns of interaction change over time. They highlight the rhythms and trajectories of collaboration, as group members, activities, and topics of concern come and go.

Recent empirical work has shown the relevance of temporal organizations of work to coordinate activities (Tang, 2002). Activities occur in a sequence, whether planned or not. They are connected to those that occurred before and after them, are caused by previous events and shaped in anticipation of others upcoming. This temporal context provides a means to understand and interpret information and activities.

For example, while most file systems and email clients allow us to sort email and other files by date, many other temporal aspects – co-occurrence, sequencing, regularity, rhythms, clustering, gaps, coincidence, etc. – remain largely invisible.

In everyday life, the relationships between people are essential. Our world is structured about social structures – there are important distinctions between friends, family, and colleagues, between close collaborators and passing acquaintances, between active and dormant interactions. The goal of our research is to make these relationships between people visible in computer systems.

Relating work objects to the social activity that surrounds them, and so providing users with the means to see their work within richer *social* and *temporal* contexts, can enrich the interactive experience. Social context highlights the way that individual activities are embedded in patterns of contact and collaboration; temporal context draws attention to the ways that those patterns change, adapt, and repeat.

By making social and temporal structures of collaboration apparent in interaction, our goal is to help people make sense of the activities around them, and so help them to coordinate their work with others. Rather than seeing interaction lists as statically stuck in "now," a structural perspective can derive a history of interactions; rather than seeing colleagues purely as individuals, a structural perspective helps to identify roles and relationships in social groups.

The goal is to reveal the patterns of social interaction that suffuse apparently individual work. This research can lead to ways of enhancing single-user applications with visible and meaningful information about interaction patterns. It may also lead to enhanced awareness tools that provide better information about the user's work and how it interconnects with other projects (Thimbleby, 1990).

In this work, I explore how to use lessons and techniques from collaborative system design to infuse a sense of everyday collaboration into the single-user experience. If computer-based work has a collaborative component, then can aspects of that collaboration be made visible, to connect a user's activity to the activity of the broader collective?

1.4. Approach through Patterns and Structure

Before an application can present information about interaction, it must first be able to identify the important aspects of this interaction. The initial questions for our research are, first, whether it is possible to design a system that can collect information useful to understanding user activity, and, second, whether that information is in turn valuable to users. The work reported here focuses on these questions.

Using email as a starting point, I have collected low-level activity information which can then be synthesized to create higher-level descriptions of the social structures within which users and their work are embedded. When this is presented information to users, they recognize meaningful patterns within it. In this dissertation, I describe a series of recurrent patterns that arise for a number of users. These patterns – machine-recognizable but human-comprehensible – are the basis for interfaces enriched by social context.

1.5. Articulating the Approach

To describe something of what I mean by temporal and social settings, consider how this proposed approach could help resolve some contemporary problems.

> Joe is a software salesman operating in the Western region. One morning, he gets a call from Cliff, a potential client whose name he doesn't immediately recognize. As they talk, he realizes they had been in touch by email several months ago. He wants to locate that information to remind himself how they know each other, so he quickly starts to skim through his email records. The name doesn't show in his address book; the contact had been fleeting, and hadn't seemed like a useful entry. A quick search on his name doesn't come up with anything; many mail clients are poor at checking carbon-copied names.

This is an avoidable situation. None of the information that Joe lacks about Cliff is really missing from the system as it is built today: Joe's email system has a record about Cliff; there is information available that could be used to solve this problem.

With a network and temporal awareness system, the history of interaction would become quickly apparent.

TellMeAbout -person cliffsmith@bigco.com
Cliff Smith appeared <u>three</u> times between June and July, 2001, always with **Jenny Doe.** (You have sent him <u>two</u> messages; **Jenny Doe** has cc'd him on <u>one</u>).

With this information, Joe is quickly able to pull up a history of this otherwise-obscure person, and is reminded of their past working context.

In this scenario, several things have happened. The TellMeAbout (TMA) application has tracked Joe's email history, and therefore is able to remind him when he was last in touch with Cliff. In a traditional email application, Joe might have to search through several different folders to find his past correspondence with Cliff; TMA has found it immediately and offers those message histories as clickable links. Joe has been reminded of when the correspondence occurred, and therefore might be able to remember what sorts of events were happening around that interaction. Even if he still is unsure what he should know about Cliff, he can check with Jenny for more information about the exchange: she was involved in their entire interaction, and so perhaps she can connect him with more information. Based on these social and temporal cues, Joe can retrieve his past information about John and handle his account much more easily.

1.6. Research Questions

Social and temporal contexts a clear niche for a technological intervention. Networks, dynamically calculated and derived, lead to cues about both current activity and what activities have recessed further into the background.

My research questions, then, are fourfold:

First, are there recurrent social interaction patterns in electronic communication and activities?

Second, if there are, how can those be extracted and analyzed? Is it possible to derive recognizable and salient social patterns from these electronic traces?

Third, can these patterns be used to design and develop a software system that is attentive to social and personal roles within the workspace?

Fourth, can a social software system based around these patterns bridge the gap alluded to earlier between personal and collaborative technologies?

A solution to the first two questions would be a software system that visibly illustrates interaction patterns. It would derive these patterns from electronically-measurable sources, emphasizing both social networks and temporal patterns, to find where the patterns are, and how they recur. This network exploration software would thus both illustrate and detect patterns of communication and conversation between users.

The third and fourth questions are answered with a second system, once which utilizes the information derived from the previous software. This latter system would, like the TellMeAbout sketch above, provide auxiliary data based on those social and temporal patterns to workplace applications. Such a system would present information on how the user's individual activities were involved in a social context.

1.6.1. Hypotheses

First, I suggest that the mechanisms of social networks and rhythms will prove an effective way to manage and organize contact information and communication records, including email collections, instant messaging, and other interaction systems.

Second, the tools build as a result of idea alleviate some of the failures discussed earlier, and will support a variety of other tasks. The mechanism to drive these tools, automatically-extracted social networks, will prove a clean and straightforward way of handling user interaction data. In particular, a suite of tools will derive and display salient information about social interactions to users.

1.6.2. Contribution

This dissertation introduces the idea of combining automatically-extracted social networks with temporal information in order to build contact management and awareness tools. It comes at an opportune time: current applications and operating systems are being instrumented to reflect some rudimentary social information. For instance, two major commercial mail programs (Mac OS X's "Mail" and Microsoft's "Outlook") both interface with instant messaging tools to reflect a single notion of a "person." In both applications, the mail program can provide an indicator of whether correspondents are online and available. Similarly, plans are being released for an upcoming operating system (Microsoft's "Longhorn") to contain a file system that can carry some social information about files.

There is now an opportunity for both storing and presenting meaningful social information within the computer system. While there are a number of ways to solve many of the individual problems highlighted above, this dissertation claims that the use of these tools – social networks and temporal rhythms – together help address the general class of such problems.

While there are some tasks that are easy to complete with currently-existing systems, others – such as tracking volatile group membership of groups – can be difficult. These tools help illustrate how the use of social networks and temporal patterns can helpfully describe group and individual social behavior, and thus can be used for contact management, information management, and awareness tools.

This dissertation maps a single path through a problem space. The dissertation presents a paradigm of social information; the methods described in it are one plausible way of handling this data it.

The dissertation will explore and articulate a concept of systems that track and understand social interaction. In particular, it will discuss the ways that current systems strive to highlight parts of the workscape, how this social information might be designed into new systems, and how it can be made visible and interactively meaningful for users.

Based on the idea that computer-based work enacts and responds to social and temporal patterns, I believe that this information should be available directly within the context of work. The primary research challenges, then, are, first, to determine mechanisms for understanding those structures, and the limits on those techniques; and, second, to explore ways to integrate this information into interfaces..

1.7. Outline

To address the first hypothesis questions, I have developed Soylent, an infrastructure and exploratory tool to determine what elements of social and temporal organization we might be able to identify in electronic records of everyday activity. We outline the design of Soylent, and discuss an initial user engagement, carried out at an external site, to test the utility of this tool. This user engagement generated a series of findings about social and temporal structure which we have incorporated into further development activities.

To address the second hypothesis, I have developed awareness tools based on structural information. I present TellMeAbout and EE4P, initial clients that use the Soylent infrastructure to provide end users with an understanding of the structures within which their work is embedded. TellMeAbout provides background correspondence information about users, and can be invoked as a dynamic awareness tool connected to incoming social information. EE4P provides enhanced social information within the email workspace.

The remainder of this dissertation is outlined as follows. I will discuss, in Chapter 2, the social network approach and the temporal approach further, in order to come to a better understanding of how these tools are usually used, and to outline how they can be applied to contact management issues. I will then (Chapter 3) discuss Soylent, a tool developed to help understand interaction structures, and will show some of the structures found (Chapter 4). Chapter 5 then shows the steps of system development based on these structures, including the tools TellMeAbout and EE4P. Chapter 6 compares the approach taken in this dissertation to the both the ContactMap project (Nardi et al, 2002b) from AT&T labs, and discusses the implications of using a system that follows the concepts outlined in this dissertation. Last, Chapter 7 summarizes contributions and points toward future work.

Chapter 2. Literature of Techniques and Technologies

In the previous chapter, I looked at a number of approaches to collaboration, and how it is incorporated into work practice. This chapter presents two analytical lenses through which we approach these topics. The first lens is *social network analysis.* Social network analysis, as used in this discussion, is a way of interpreting sets of individual relationships and combining them into a collective picture of the structure of social interaction.

The second lens is the *temporal* shape of relationships. This chapter examines the ways that time has been a subject of study within CSCW, and how small group research has historically handled and examined time.

2.1. Social Network Analysis

Social network analysis is a tool for understanding the shapes and contexts of group interactions. By mapping the relationships between groups of people – a map in which nodes are individuals and ties are the collected or measured relationships between them – analysts can develop both qualitative and quantitative descriptions of the group's relationships (Wasserman and Faust 1994). Network analysis has been applied to a wide variety of organizations, as well as to other forms of social relations, and has found use in sociology, organizational behavior (as in Ahuja and Carley, 1998), and epidemiology (as in Morris and Kretzschmar, 1997), to name only a few. Recently, network analysis has also become better-known in more far-flung fields and the popular press; examples have been seen in physics (where it is a tool within complexity theory) and biology (Jeong, Tombor et al. 1999). The tools of network analysis, including identifying people central to networks, locating network "holes" (Burt 1992), and matching network structures, have been broadly applicable within a variety of different areas.

A growing literature has begun to apply the analytical methods of the field to the computer- and social-networks in computer-supported collaborative work. Wellman et al (1996), for example, suggested that the patterns of connection between members of electronic communities made up parts of a social network, drawing analogies between mailing lists and dense networks. His work helped bring the use of social network analysis to CSCW studies.

This section will briefly discuss the current state of social network analysis, and how it has, and has not, been applied to CSCW. It will argue that the field of CSCW can find substantial contributions from looking at egocentric social networks. It will present a framework for discussing social networks, and will re-interpret a number of past projects in terms of a network perspective.

2.1.1. Social Networks for End Users

There is a particular challenge in applying social network analysis to CSCW: CSCW, as a design field, is accustomed to examining projects from the user's perspective. Social network analysis, as an analytical technique, is more accustomed to dealing with groups of people from the outside. Finding places where network analysis directly addresses the problems of CSCW, then, is a particular challenge. Focusing on the end-user only limits the scope yet further.

It is reasonable to ask whether there are ways to introduce network analysis more tightly into the design cycle. Can network analysis have a direct value to the members of the network, rather than only the indirect results of an analyst making structural recommendations and changes. In other words, can aspects of a network be presented to a user, either directly or after some steps of algorithmic processing, and will the user be able to make sense of it?

Users might well be interested in information derived from their networks: the raw material for network analyses is often collected directly from users, and the material is often quite salient to them. A card-sort, for example, asks a participant to stack piles of cards to correspond to her understanding of groups within the network. Data can be collected by having participants fill in matrices of connections, listing off who in an organization they interact with, or depend upon for help. Social network information can also be observed by researchers looking in, counting interactions between people (as in Gibson 1999). All of these forms depend on collecting information from the participants. Yet the participants are often involved in only the data collection phases of the network analysis process: it is their responses that are used to generate the network, but they are not involved in the results.

For the purposes of this discussion, though, the question must be how to apply these networks to the needs of end-users. While hand-prepared networks might not present new information, automatically-elicited networks often contain unfamiliar material. Both of these forms might contain information that may be usable for computer systems: information about social context can be gleaned from network interconnections, for example. It then becomes reasonable to ask: is it possible to algorithmically prepare and present networks to users based on automated collection from CSCW applications, such as email or chat applications?

In order to highlight the variety of types of network analysis prevalent in the field, this chapter reads the notion of "network" broadly. I consider a network generally as a set of users and the relationships between them; the network may be homogeneously composed of people, a bipartite graph of people and the artifacts around which they interact, or even more heterogeneous graphs.

Many technologies share underlying philosophies, if different implementations. It would be impossible to present a reasonable sample from the breadth of social networks within the field, while also striving for depth within all subfields in which networks are discussed. Instead, I will approach the core insights of the different areas of research, choosing several projects from each that illustrate the general concepts at hand.

This discussion is limited to techniques where end-users benefit from the network directly: that is, when it is exposed to them in an interface, or made a tool in a system they can use. To be sure, there often are indirect beneficial effects from a network analysis to the members of a group. For instance, a network analyst might present a visualization of organizational communication patterns to a manager, who might implement new workplace policies. While these are beneficial to the participants in the organization, I am more concerned about results that users can interact with more directly.

This discussion does not, however, require explicit networks made of ties and nodes. For example, a contact list or an address book can be seen as an egocentric network with implicit ties. Indeed, several variants of this sort of list will be discussed. This section, then, begins with a discussion of *contact management,* understood as a form of egocentric network management. When a user manipulates a contact list – whether through a rolodex, palm pilot, or cell phone autodial list – they are working within one of the most popular forms of networks available. In computer systems that make the activity of members of a contact list becomes visible, the list becomes a sort of limited *awareness system.*

Nor need the network necessarily be known to the user. Analysis on some networks is done on implicit relations, such as "is interested in similar topics". A generalization of the social network concept, that of considering a graph with weighted similarity measures between every user and every other user, leads to meaningful visualizations and useful graphs at the heart of *collaborative filtering* research. Collaborative filtering can be seen as a network linking artifacts to people: in particular, it connects people to the artifacts that they are mutually interested in.

Collaborative filtering is one aspect of *social navigation*, an attempt to derive information for users based on collective actions. Social navigation is an approach to network analysis in which activity in the network is analyzed, then presented collectively as a supplement to information-seeking. Last, this chapter will explore various network *visualizations* that provide a visual representation of a social networks or online interaction.

The various areas discussed – contact information, awareness, knowledge management – are all very broad; it would not be feasible to cover them all in great detail. Rather, this chapter strives to select examples of particular interest, and then discusses the range of variation in other related systems.

2.1.2. A Framework for Discussing Network Research

In order to categorize and discuss these technologies, I isolate several important attributes of the technologies. A combination of these questions forms rough axes to categorize and extract relevant features from the technologies.

Table 2-1. A Framework for Discussing Social Network Research

Orientation	Egocentric	Sociometric
Data collection	Entered	Discovered
Modes	Single-mode	Multi-modal
Presentation	Displayed directly	Computationally processed
Visibility	Visible identity	Anonymous
Involvement	Members are end-users	Members are not end-users

Orientation denotes whether the network has an *egocentric,* or *sociometric* (whole) orientation. In a *sociometric* study, the researcher attempts to find the connections across a population. Such studies are common, ranging from Eveland and Bikson's (1986) early work with email at the RAND corporation to recent projects examining the interrelationship between communication patterns and organizational structure (MacDonald 2003).

In an *egocentric* study, respondents are asked to generate a list of names of people with whom they are connected, and only that smaller network is examined. For example, a study by McCarty (2002) explicitly collected the interconnections between every pair of named people from a name generator. The study could then examine the clusters in that cluster

How is the **data collected**? The network may be *entered* or *discovered*. In the former case, a system might solicit network information from users; in the latter, the network might be automatically extracted from online data sources.

How many **modes** are stored in the network? The network may be *single-modal*, tying users directly to other users, or may be *multi-modal*, tying users to each other by way of shared artifacts and activities.

How is the network **presented**? The network may be *displayed* directly to users, or the results of a numeric process may be shown.

How **visible** is participation in the network? The members of the network may be *known to the user*, or may be maintained *anonymously*.

How **involved** in the network are the end-users of the network? This chapter largely examines systems in which the end-users are members of the network, but touches on some other systems for contrast.

This questions will form a framework for discussing research on networks for end-users.

2.1.3. Egocentric Analysis within CSCW

The simplest form of a social network is an egocentric network. Indeed, when people ordinarily attend a "networking" event or speak of their "social network," they ordinarily mean their immediate contacts and friends-of-friends, and not those at greater distance. This manifests itself as a contact list or an address book. A contact list is simply an egocentric social network, without interconnections.

This section reviews general contact management systems. It then discusses awareness systems, understood as contact management with presence information. Last, it extend awareness into social navigation.

2.1.3.1. Contact Management

From this perspective, Rolodexes, address books, instant message buddy lists, cell phone number lists (Berg et al. 2003; Grinter and Eldridge 2003) and email contact lists (Ducheneaut and Bellotti 2001) are all personal views of networks. Address books are a traditional source and destination for contact information. For example, both cell phones and email programs store contact lists as a convenience to the user: an index into frequently-dialed numbers and often-sent messages. Personal information management (PIM) devices often synchronize with these address books to build a minimal social network, flattened into a small number of contexts: "work" and "personal," for example.

To place contact management within the rest of the framework, most of these networks are ones that were *entered* by their maintainers; in fact, managing a contact list is a substantial amount of work. The members are all *known to the user* and non-anonymous, are *presented* directly back to the user. Last, there is, generally, only one user of the address book.

Where traditional address books–such as those stored by email programs or PIM–collected and managed contacts in a set of categories, some forms enhance these networks with activity or presence information. It is common for instant message tools to indicate other active users on the contact list (conventionally known as a "buddy list"), often providing some measure of their activity (Isaacs, Alan et al. 2002; Grinter and Palen 2002; Herbsleb, Atkins et al. 2002). Users leave "away messages" messages for their contacts that indicate their current activities or feelings; the system also indicates whether the buddy is logged into the system, and whether they are idle, showing when a user hasn't been active on their computer for a time.

These network ties may be outgoing links only (as in AOL Instant Messenger's – AIM's-- implementation), or may be constrained to bidirectional links (as in Yahoo's implementation, which prevents users from having someone on their buddy list who hasn't approved them).

The cooperation of several users can construct a sociometric network from contact lists. With the right analysis tools, a portion of the broader underlying network can be constructed. BuddyZoo (Web: BuddyZoo) collects the network of AIM Buddy Lists from users who volunteer their buddy lists; it then allows those users to see visualizations of their buddy lists, see popularity measures, and count network distance between screen names.

A contact list ordinarily provides nothing more than a list of its members. Some research projects, however, have explored further dimensions and attempt to articulate the network more thoroughly. AwareNex (Tang et al. 2000) provides additional information about where a user is located, and attempts to provide up-to-date information about how a user can be best contacted. However, some recent work has begun to suggest that these systems might be enhanced with additional activity awareness. Thus Berg et al (2003) suggest that the list might be presented differently to users; dynamically sorted, animated, and clustered based on how a given participant is interacting with his friends. Berg's work also suggests a richer notion of activity; her work proposed a variety of ways in which the contacts might highlight their movements and actions.

Contact lists are not necessarily individual artifacts, unshared by other users. Grinter (2003) discusses teens sharing their phones to read each other's address books. One of their study participants reports that he would borrow others' phones and re-insert his name, irritated that he had been removed. A well-filled address book was treated as a sign of social status; when a participant in their study would receive a new phone, they would often initialize it by copying an entire number list out of another's phone.

As the contact list begins to expand, categorization becomes difficult – and a categorized contact list bears strong resemblance to a social network gathered through a card-sort, in which a user connects groups of contacts together. Whittaker and Jones (et al., 2002) discuss the difficulty their interviewees had tracking the associations between their contacts, and remembering who they were in touch with. The ContactMap system (Nardi et al., 2002b) was built as a partial solution to this problem. It visualizes clusters of names on the user's desktop as a way of managing contact information. It initially collects a list of names from an email record; it then allows its user to cluster the names as they will into explicit groups. This then becomes an enhanced information manager: it is possible to view contact information for each person or group, and to send email messages to the entire group. The ContactMap also functions as an index into the past correspondence, thus unifying the notion of the visible map with the actual interaction. ContactMap also contains basic awareness indicators; it can signal when new email has been received from a contact.The ContactMap, like the buddy list and the phone quick-dial list, is a fixed one; names can be manually added to it, but it does not adaptively modify its lists based on external changes. (ContactMap is discussed in more detail in Chapter 6).

2.1.3.2. Awareness and Presence

A contact list with active information becomes an awareness tool. Awareness, a popular theme within CSCW, was described by Dourish and Bly (1992). They describe awareness as "knowing who is 'around', what activities are occurring, who is talking with whom, and provides a view of one another in the daily work environments." This definition does not require computer support nor networks; an intern in the corner cubicle has good awareness of who passes into the building. Heath and Luff (1992) describe the subtle techniques used by a pair of coworkers communicating about emergency management to each other and to their various co-workers on the London Underground. CSCW systems support awareness by relaying subtle information about what remote co-workers are doing.

Within the network framework, awareness systems match contact lists: egocentric connections, entered by end users, allow other known users to be displayed on screen.

In the CSCW world, awareness tools make it possible for users to learn more information about the activities of a group of correspondents. Schmidt (2002) attempts to arrive at a taxonomy of awareness, emphasizing both active forms (such as Heath and Luff's emergency team telegraphing important facts to each other) and passive forms (in which users gain so-called peripheral awareness with tools that suggest, rather than show, the activity of their partners). Emphasizing the work practice that leads to awareness, he writes that "'Awareness' is not the product of passively acquired 'information' but is a characterization of some highly active and highly skilled practices." Portholes (Dourish and Bly 1992), for example, is a peripheral system that allows users to glance into an office some distance away. Thus, a casual glance would inform a user whether their collaborators are busy in a meeting – or idly tapping their thumbs, and that it might then be a good time to reach them. The "skilled practice," in this case, is then one of interpreting availability, activity, and interest from the video images.

Gutwin and Greenberg (1998, 1999) point out that peripheral awareness allows users to communicate more efficiently and to get more context about what other users are doing. Their awareness system gave users the ability to have rough location and activity information on their collaborators in one experimental condition; in another condition, users had only their own information. Users completed their tasks more quickly with awareness, and were more able to communicate with their collaborators.

Further information can come not only from who is available, but who is interacting with the system. The NyNex Portholes system (Lee et al. 1997), for example, provided lists of both “people in my portholes viewer” (which allowed a user to see all users who were connected to the system) and an overview of “who is looking at me.” This seems to be more fair, by a sense of reciprocal awareness, and their users liked knowing who was watching them. From a network point of view, the NyNex system then allows users to perceive incoming links as well as outgoing ones: they can be aware of how their network is shaped.

Ackerman and Starr’s Espresso (1995) provides a global level of information across the network. In their system, users can see not only the messages that are sent around in a series of chat rooms, but can see some global statistics of the system, such as how many messages are being sent and who is sending them. This allows users to anticipate which sections of the system are active enough to be worth watching. These activity cues allow more active participation.

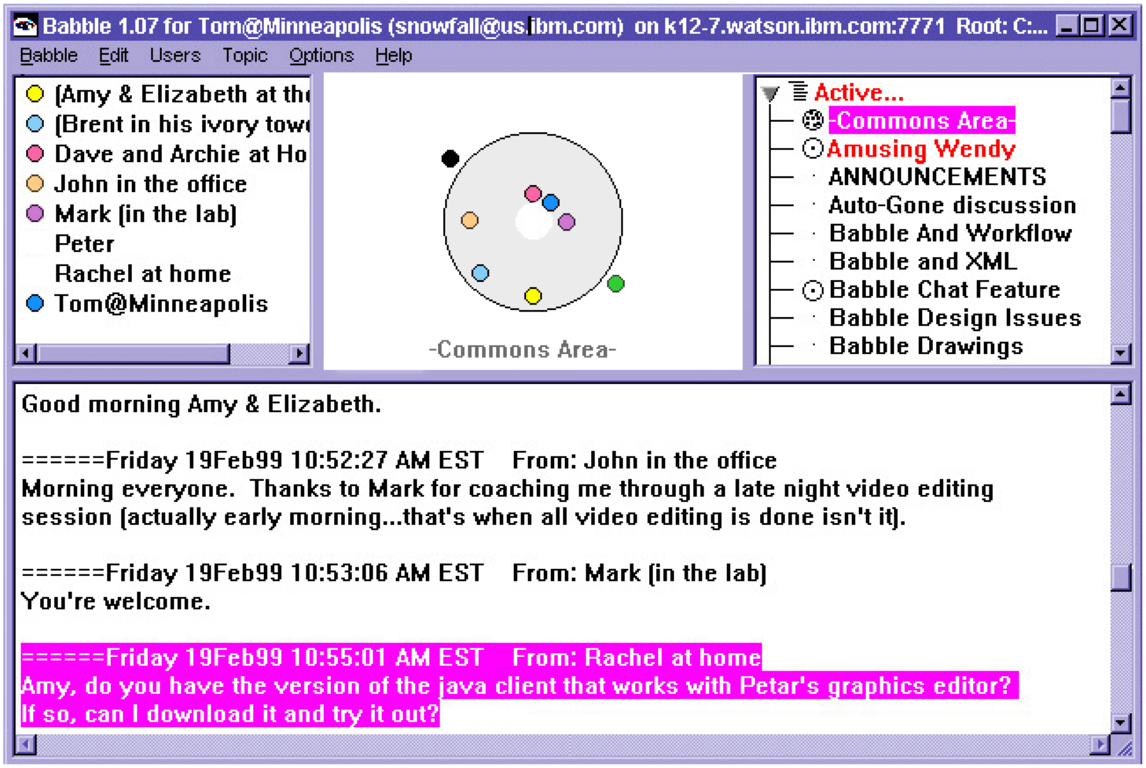

Figure 2-1. Babble interface.
Note circular "social proxy" at top representing the conversation.

Similarly, Babble (Bradner et al. 1999, figure 2-1) combines social activity indication with a simple visualization across a limited team. Based on a series of principles of transparency and mutual visibility (Erickson et al. 1999; Erickson 2003), the system allows team members to know who is contributing to a persistent conversation. The system splits conversation into a series of themed "topics", and so users can also tell which conversations have current updates. Teams tend to set up independent rooms for interaction; similarly, group members affiliated with particular interests often linger within a single topic. Babble, then, defines an affinity network from person to topic, and annotates that network with information about which people are most and least active, and which topics they are active within.

A particularly relevant form of workspace awareness is the "Anchored Conversations" system (Churchill et al. 2000). Within that system, users can attach fragments of conversation, such as instant message transcripts, to documents. Documents that had been so annotated then carry with them information about who was interested in them, as editors and commentators. Those annotations describe, in a very practical sense, a notion of who is interested in a particular topic and thus highlight an affinity network.

2.1.3.3. Social Navigation and Collaborative Filtering

One closely-related cousin of awareness is the field of social navigation (Shardanand and Maes 1995; Hook, Benyon et al. 2003). In a social navigation system, the actions of a group of people (such as book purchasers, or movie reviewers, or web surfers) are collected and used to provide insight to later people doing the same things. Users, then, gain awareness of the aggregated actions of other users in the systems. In Wexelblat's Footprints (2003), for example, links on web sites begin to show signs of change as users read links more often. The system monitors the paths that surfers take through the web and feeds them back to later users, constructing recommended paths and popular ways of finding information. In related systems, documents can begin to show their own age as users work with them, allowing later readers and editors to notice popular, important, and heavily-modified parts (Hill et al. 1992).

Within the framework, social navigation continues to present an egocentric view. The network is often a discovered and anonymous one, as users wander between sites seeing the accumulated tracks of previous users. The network is bimodal; users are joined by the documents they share, and don't see each other directly.

A related technology, collaborative filtering (as Amento et al. 2003a, and collected at Resnick and Varian 1997) is based on the principle that similar users are likely to be interested in similar things. Users' preferences are placed into a large, multidimensional space, and are clustered automatically based on proximity within the space. Nearby users are given access to collective information about each others' preferences. The online bookseller Amazon.com (web: Amazon) can, at a user level, describe the notion of "people like you" who recommend particular books or ideas. Collaborative filtering, then, defines an implicit network across the set of all participating users. As we will see, this sort of network can be mined: users at far points can be seen as isolates; users further within the network can be seen as central.

Domingo and Richardson (2001) consider the implicit network from collaboration systems to be a social network, and analyzes it accordingly, searching for communities that may be interested in a particular marketing tactic. The network ties are those of "influence," and a user is said to influence another if the recommendation of the one applies to the other. This is, he suggests, in some ways a purer network than other forms; a collaborative filtering system that compares strangers can be fairly sure that the users are joined over no other media. This network, then, is generated independently from friendship or geographical networks. It is no longer quite a "social" network, although it is understood as a way to interpret and locate influential people.

2.1.4. Mining Social Networks

Results from the automatic analysis of a social network can be fed back to users. One genre that has produced a series of results for end users comes from the idea of mining network structure to provide information about the underlying data. Some of these networks, drawn from web pages, are only implicitly social, but the results that can be mined from them highlight ways that results can be presented. For example, pages that have structurally-relevant positions within the networks can be placed higher in a list of returned values.

Page and Brin (et al, 1998) introduced the PageRank system, which is the basis of a popular search engine, Google. With the PageRank algorithm, web pages are assigned values based on their location within the network: well-ranked sites are defined recursively as those that are referred to by other well-ranked sites. A related algorithm, Kleinberg's (1998) hubs-and-authorities model, also highlights good sources of information. Kleinberg's model considers a network to be divided into two groups, "hubs" and "authorities." Again, the definition is a structural one: a good hub is one that points to good authorities; while a good authority is one that is pointed to by good hubs. Users of the algorithm have developed prototype tools to find sections of the web on particular themes. These can lead to interfaces that highlight navigation cues through the space of web pages. One application of these technologies is the TopicShop (Amento et al. 2003b) system. It uses so-called "social data mining" to sort out important pages from large lists, and then allows users to gather and cluster the data in ways that are particularly relevant to them.

Social networks may also be derived from other socially-driven web pages, such as weblogs. In a broadly non-academic trend, a series of weblogs have ranked each other using PageRank-like algorithms (Web: Blogosphere Ecosystem), automatically rated each others' importance in the "blogosphere" by mutual links, and displayed the results publicly as indices into the weblogs. Weblog authors can, and do, check these ratings to see how their weblog ranks; web surfers check the rankings as ways to find their own reputation.

Other systems search weblog links to find a consensus on important pages. If enough weblogs point to a particular site, than it must be of some importance: an idea that has captured weblog community's collective imagination. The network this searches is a bipartite one (the "hubs" are all weblogs; the "authorities" are the pages they search for, and the hubs aren't ranked) but the interface does allow users to know who is important (Web: Popularity)

2.1.4.1. Expertise Location

One popular use of network information is for expertise location. Based on the observation that the expertise in many organizations is broad enough that the organization does not know what it knows, expertise location attempts to describe and locate expertise within an organization. One mechanism for locating this knowledge is to produce a directory of employees and their areas of expertise. These directories can be assembled in a variety of ways; however, they still retain the social challenges of fostering cooperation between employees. Network analysis attempts to help with both of these problems, by examining both what communities the employees are connected to, and by examining what knowledgeable employees are close enough to be a friend-of-a-friend. In an example of the latter, Tyler et al. (2003) use email network to identify close-knit interest communities at a research lab, separating groups on the basis of their most central connections. The communities were found by examining pairs of users who emailed each other frequently.

Research into expertise location uses sociometric networks, not egocentric ones, based on information accumulated across an organization. The information can be accumulated through automatic means (as in Tyler), or manually. It handles closed organizations; even if the members are not be familiar with each other, they can find out information about each other through business directories and organizational charts.

Some uses of expertise location have been very explicit about their use of networks. For example, Busch et al. (2001) use social network analysis to try to detect patterns of tacit knowledge. Their network is collected from a name-generator instrument; their analysis seeks out top collaborators and the ways that they collaborate with each other. They then generate diagrams of networked collaboration and try to find crucial players within the organization. In the study described in the paper, they find that the group secretary forms a crucial gate-keeping role between two dense networks within the office. Busch presents the network as an analytical tool, but does not present a tool for end-users to work with the network themselves.

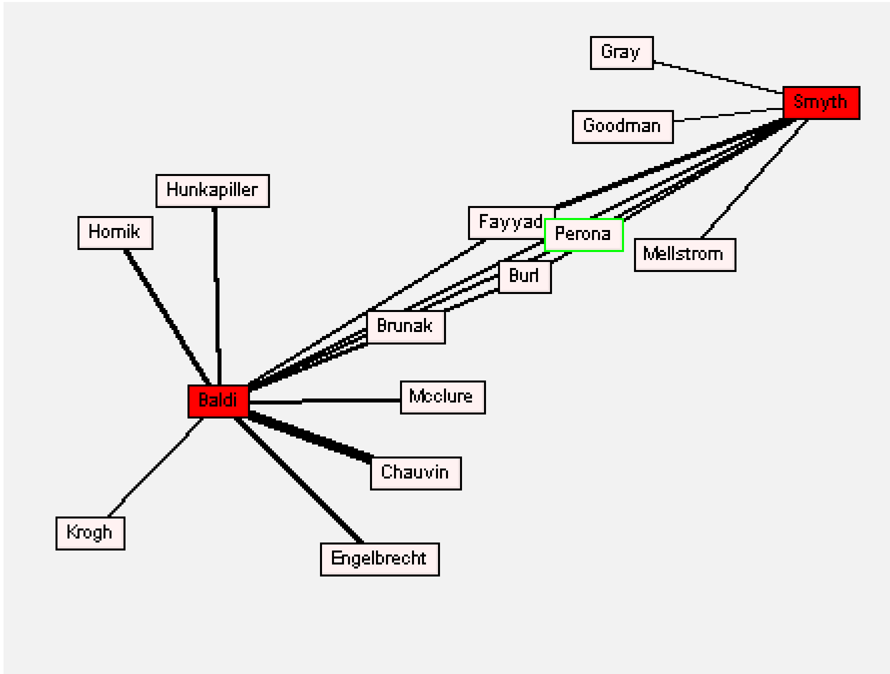

Figure 2-2. ReferralWeb.
Note co-citation connections between two (red) researchers.

Implicit networks – derived from socially published information – can be tied to more explicit user information in order to produce results that can be meaningful to users. ReferralWeb (Kautz et al, 1997; figure 2-2) follows traditional network co-citation models, but allows end-users to browse between different members of the network. Each author is associated with past co-authors for papers (and other names that may co-occurred on web pages), as well as a list of keywords. ReferralWeb then makes an explicit and visible social network from this implicit information. The ReferralWeb interface allows users to view portions of the network, and thus can find important authors in the field, locate shortest paths between important authors, and search for communities of knowledge: groups of authors who work on similar topics and are closely connected to each other. Since the ReferralWeb system uses publicly-available information and posts results to public websites, the users of the ReferralWeb network are not the same as the participants in the network.

The IKNOW system (Contractor et al., 1998) also captures social information to display networks to users. It collects information about areas of knowledge and network ties from within a restricted community of users (who either enter their data manually, or have it scanned off of home pages). It then presents information in the form of several different networks, based on common interests, common referrals (to external web sites or other users), and common vocabulary. It presents the information back to users as networks, and, like ReferralWeb, displays networks as well as network measures to the users

2.1.4.2. Reputation Systems

A number of systems use network analysis as a way to synthesize a "reputation" figure. In some systems, a well-thought-of person has more weight in rating others; in others, recommendations are only valid within a 'network of trust', the people who have been rated, and who rate those raters. The former is exemplified by systems such as Slashdot's "karma" (Web: Slashdot) system. In the framework presented above, reputation systems accumulate information from a network of connections. They use information entered by users, and largely do not explicitly display the networks they use to calculate their connections. Members of the network know each other, and are the end users of the system.

That genre of reputation system was tested by a series of experiments at Microsoft Research. Jensen et al. (2002) extracted network information to present reputation information to users. They then were able to test the system on a series of users, asking them to predict how likely they would be to interact with certain alters based on their online reputation. Their study found that reporting similar interests was a strong determinant of a user's desire to enter a chat, while implicit location on a network (as derived from network ties) was less useful.

The "Advogato" system (Web: Advogato) evaluates the abilities of open source programmers to work with each other by extending what the system refers to as a "web of trust." Programmers who have witnessed other programmers' skills are able to rate them on a scale ranging from unskilled to expert. Users of the system – the programmers – can then view the ratings of other programmers with whom they have not personally collaborated by being shown a compound path rating. (Thus, if I rate a programmer as "apprentice", and he rates a collaborator of his as "expert", his review will have lower credibility in the system then the equivalent rating by a programmer who I have rated well.)

A similar network is embodied in the key exchange system of private-key infrastructure (PKI) systems. Within the secured world of PKI networks, all information must be transferred either by hand, or encrypted with a known key. These keys, though, are information themselves. Once a user has collected a small number of keys by hand that she trusts, she may collect other keys safely only by getting them from trusted users. These new keys can then also be trusted. The recommendations and degrees of trust, then, implicitly imposes a network of connections (Web: PGP). In practice, the space is structured as a strongly-connected hub; a small number of trusted repositories and users are well-rated and act as certifying authorities for most other users (as shown by Guardiola et al. 2002).

Other sorts of network-based reputation input and output systems are available. The LambdaMOO Cobot (Isbell et al. 2000) wandered about its online world, collecting "social statistics": what users interacted with other users, and how much. This information was relayed back to the users through a social player, the Cobot, who could answer simple queries such as "Who is like me?" Cobot would respond with messages like

> cobot [to Gabaldon]: Here are your relationships with Sparklebug... Sparklebug is ranked 7 on your list of playmates. You are ranked 19 on Sparklebug's list....
> (Isbell, Kearns et al. 2000)

Although the Cobot collected pair-wise ties across a number of different modalities (who hugs whom, who talks to whom, and so on) – perfect for network data – the information was not explicitly grouped into network information.

A related series of online tools, among them "Friendster" (Web: Friendster; boyd 2004) and "SixDegrees," (Web: Sixdegrees) present networks directly to users. Users of the software select a list of friends by mutual ties, and then can view their immediate friends, the friends of their friends, and so on. Both services overlay a dating and connection service over the network; users can search their friends-of-friends (to within three or four removes) to find dates or new friends, and can arrange introductions through the sometimes-tenuous chains that connect them. There is, however, no built-in structural analysis: users are not automatically rated based on their friends' connections, for example.

2.1.5. Social Visualizations

The last set of uses for network information for end users have come from visualization areas. Social networks, seen by users, become in themselves interesting: users find their own networks interesting; networks serve as a tool for recall, for storytelling, and for introspection. Visible networks (as shown with ContactMap) can be a tool for sorting and organizing information.

These networks are both egocentric and top-down, are both entered explicitly by users and discovered. The information in them, though, is displayed directly to users; the members are known to each other. All networks discussed here are meant to be used by the people who are depicted: the visualizations are meant for end-users.

Most of these systems describe visualizations of collaborative social spaces: spaces where users interact with each others. They then try to visualize some aspects of the conversations within these spaces, understood as parts of a network of conversation. Email-based systems are a way to gain a new perspective on personal interaction information, while visualizing online conversation spaces allow a group to share a common view of a space. Each of these domains have important features.

2.1.5.1. Email Visualizations

Farnham (2002) has presented a variety of systems that develop social networks out of email records. One project centered on the implicit networks that can be derived from co-membership on email lists, and visualized networks based on that information. The networks are portrayed as traditional social networks; nodes are connected to each other explicitly. In addition, the project extracted a list of "closest co-workers," and annotated corporate information pages with the list of close connections. A second project highlighted the networks that can be derived from an email database; using data extracted from contact lists in the inbox, the system displayed connected networks of users. This view roughly clustered the list of contacts into groups, and allowed the user to change various thresholds of connection in order to make the groups more or less connected.

A series of projects from MIT Media Lab have explored different aspects of conversational network visualization. Social Network Fragments (boyd 2002 pg. 75-94, Figure 2-3) illustrates a single users' email spool, visualized as a social network. The network, assembled through a combination of automatic collection and manual labeling, draws several different sorts of ties, ranging from "is weakly aware of" to "trusts." In this case, "weakly aware" is a general tie for pairs of users who have both been carbon-copied on a single message. The most trusting tie, in contrast, is reserved for those who have been placed on a "BCC"

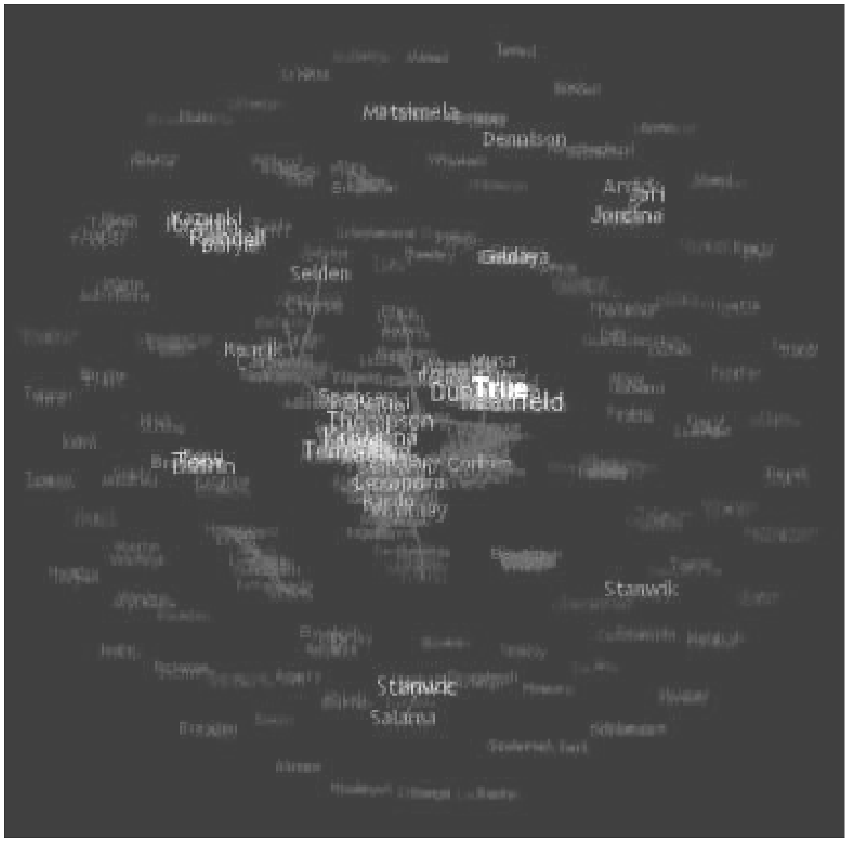

Figure 2-3. Social Network Fragments.
The network surrounding one cluster of names is shown.

(blind carbon-copy) list; those are the contacts who can be given non-reciprocal access to the names of others.

The "PostHistory" project (Viegas, boyd et al 2004) visualizes activity on a contact list – specifically, in an email archive – but declines to explicitly show network interactions.

2.1.5.2. Visualizing Interactive Spaces

While mapping email networks is a study of a localized set of interactions, visualization projects have also striven to show the size and characters of broader interactive spaces. A number of projects – collected, for example, in the Atlas of

Cyberspace (Web: Atlas; Dodge and Kitchin 2000) – have attempted to illustrate aspects of the topography of cyberspaces. The subset of these that examine online conversation spaces have more explicitly-social aspects, and thus are more relevant to this discussion.

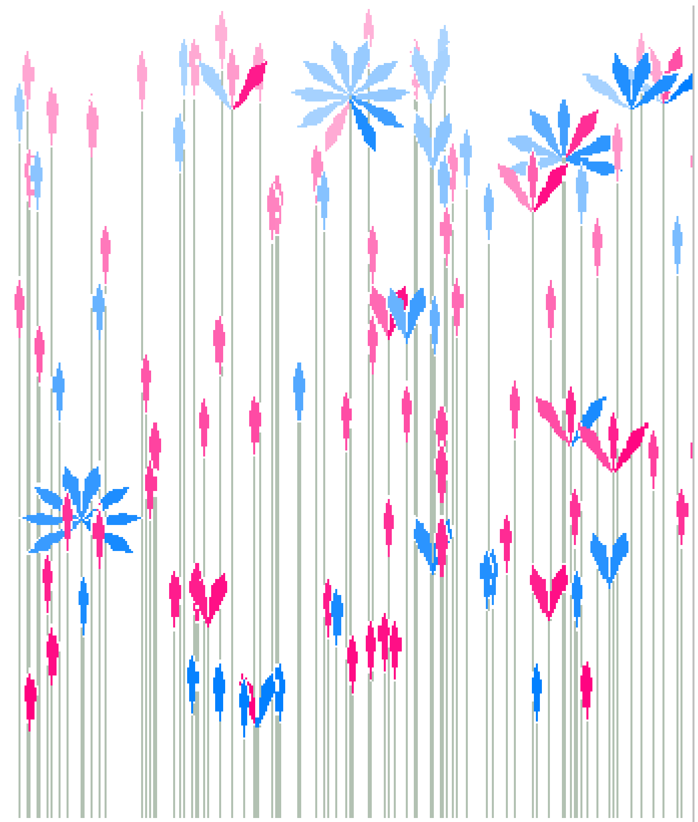

Figure 2-4. PeopleGarden.
People are represented by flowers on stalks; larger flowers have participated more.

"PeopleGarden" (Xiong and Donath 1999, Figure 2-4) attempts to create glyphs of users as individual flowers; their shared interaction patterns are seen as a garden. Each flower-shaped glyph represents the time since the user last posted to the group, the number of messages to which the user received responses, and the amount of time a user has been in the group.

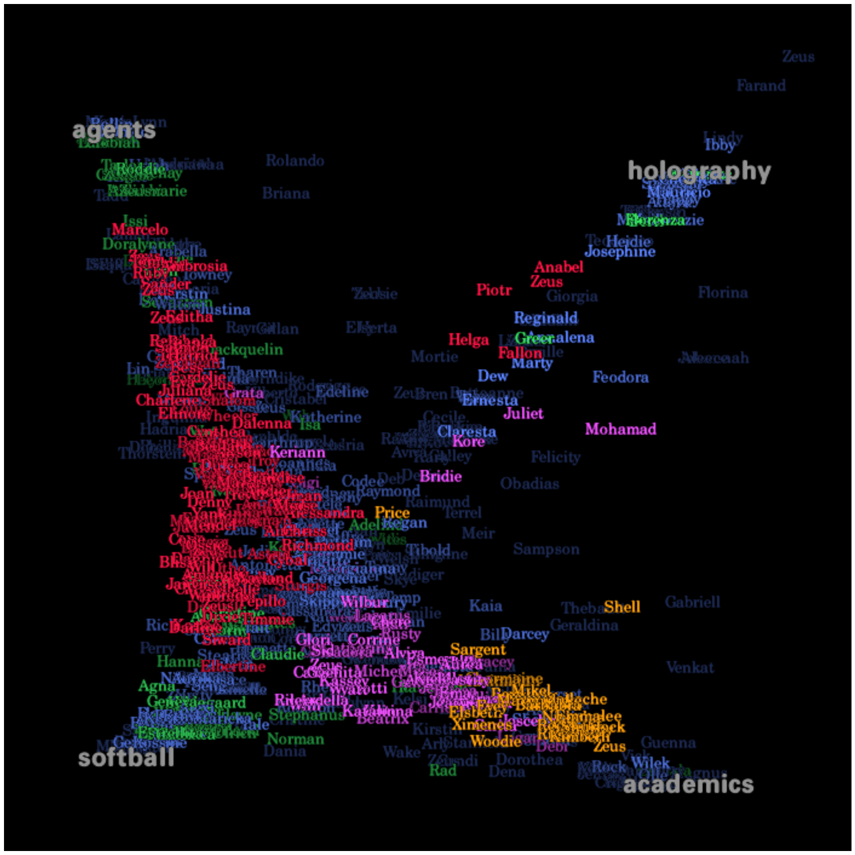

Figure 2-5. VisualWho

Four mailing lists – agents, holography, academics, and softball – align a group of users inside the broader space.

"VisualWho" (Donath 1995, Figure 2-5) shows individual participation on mailing lists – much as does Farnham's visualization – but doesn't attempt to find linkages between individuals. Instead, it broadly pulls users into regions of a display based on their affiliation with mailing lists, and their affiliation with individuals on mailing lists. The "VisualWho" display is sensitive to some forms of awareness; it dims the display of members who are not logged in, allowing the user to locate those who are active.

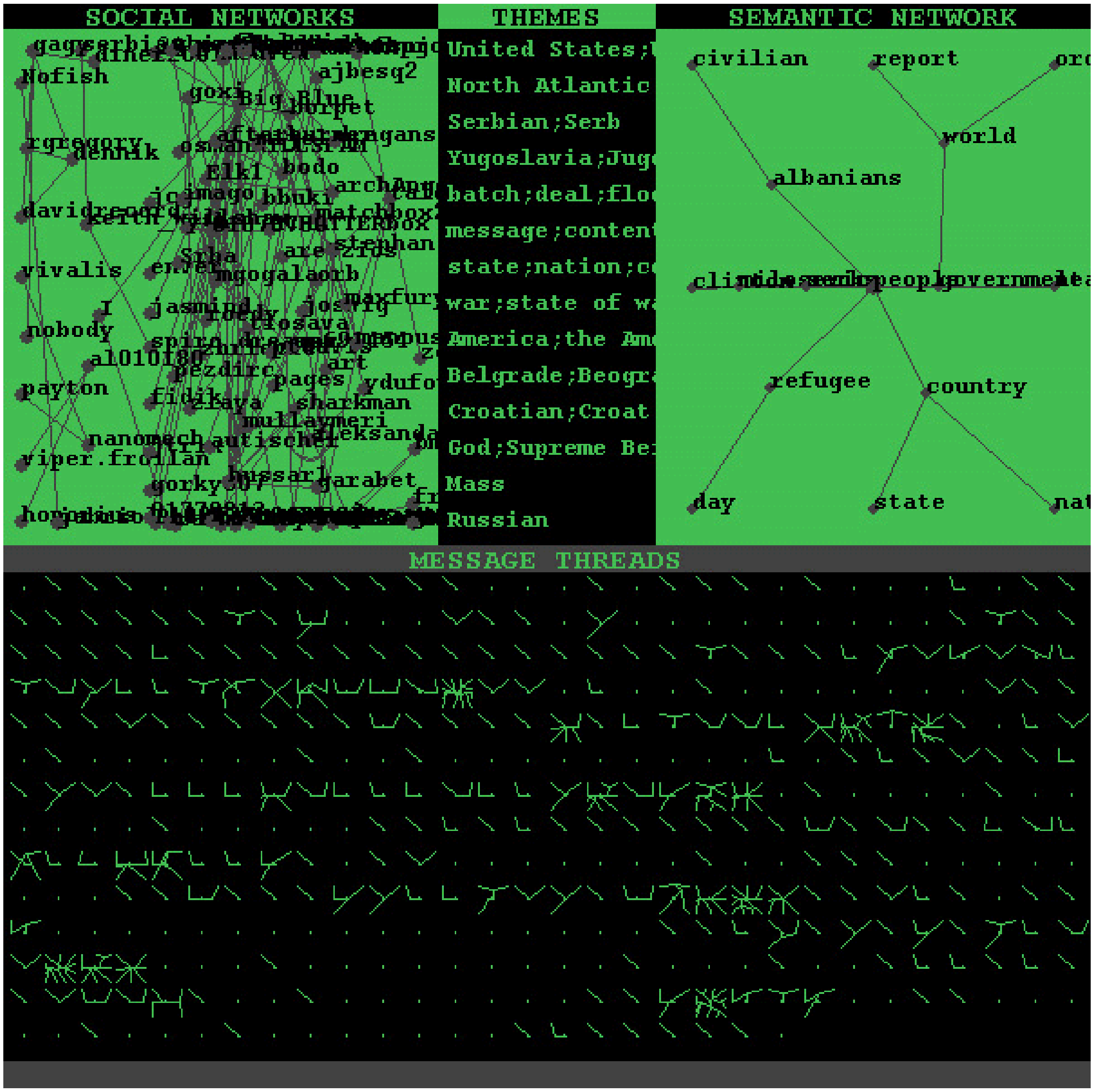

Figure 2-6. Conversation Map.
Note person-to-person connections at top left, word-to-word connections at top right, and threads at bottom.

Other views of conversational networks include Sack's Conversation Map (2003, Figure 2-6), which visualizes Usenet newsgroups. This tool shows the implicit networks derived from which users have responded to other users. They are meant both as tools for analysts, and eventually to be placed in the hands of end-users – members of the newsgroup can view their own participation, and can

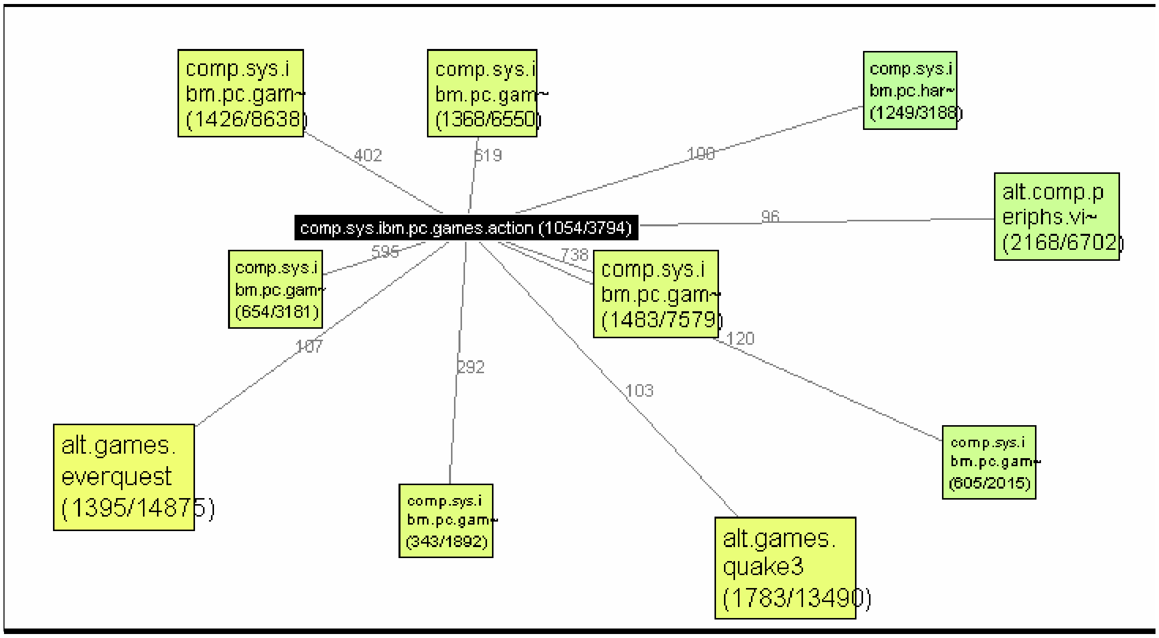

Figure 2-7. Crosspost visualization from Netscan. The most-connected groups around comp.sys.ibm.pc.games.action are displayed.

begin to understand the interactions within the newsgroup. Similarly, Ducheneaut's Open Source Visualizer (Ducheneaut, 2003) shows Open Source mailing list communication. Group members can review their own contributions to the mailing list, can examine important conversation themes, and can understand how the project breaks up into implicit teams.

Smith's "Netscan" (2003, Figure 2-7) is a real-time web service that also dispatches "newsgroup report cards." While it does not show inter-user network ties, it does highlight relevant network statistics about people interacting. It provides information, per user and per group, on how many people responded to messages, how many groups they posted to, and other similar statistics. "Netscan" also contains a network-like visualization based on cross-posts; pairs of groups that receive many messages posted to both groups are thought to be more closely related, and get stronger ties

AT&T's Clan Graph (Terveen, et al. 1999; Figure 2-8) attempts to present a hybrid between displaying a network directly, and collecting information based on the network. The system organizes web sites based on their conceptual (and geographical) adjacency. Interlinked web pages are sorted into sites (based on common URL) and combined into clusters (based on network interconnections). The system then presents back to users a network of web pages; as the user moves from one page to another, the system presents concentric circles of pages relevant to their initial query.

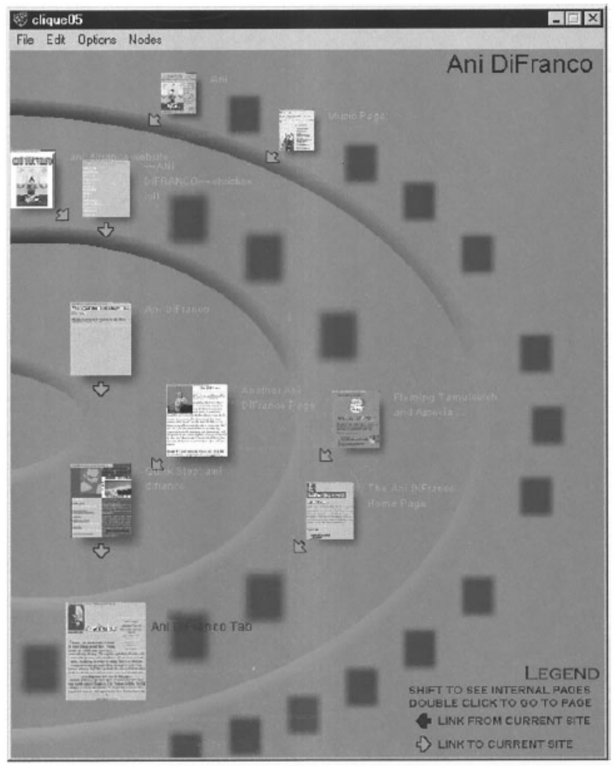

Figure 2-8. ClanGraph. Pages with decreasing levels of similarity surround the central topic.

2.1.6. Social Networks in the Framework

Social networks are frequently used as an analysis technique, but are less frequently directed toward end users. The field of CSCW, already oriented toward end-user concerns, may find social network analyses a useful tool. Designers can develop applications that collect networks from user interaction, process them into results, and feed them directly back to users. While this is promising, the few forays into social network analysis have not yet fully caught on as an end-user design technique. This is partially result of cultural differences: social networks is a technique from analytical fields, and particularly sociology; the tracks of CSCW that build systems are more accustomed to seeing themselves as design-based fields.

Table 2-2.Attributes of Social Network Systems Within the Framework

	Orientation	Collection	Modes	Presentation	Visibility	End-User
Contact Management	Ego	Enter	1	Display	Yes	Yes
Awareness	Ego	Enter	1	Display	Yes	Yes
Social Navigation	Socio	Discover	2	Calculate	No	Yes
Mining	Socio	Discover	1 or 2	Calculate	No	Both
Social Vis.	Both	Both	1 or 2	Display	Yes	Yes

There are, however, many techniques within CSCW that use information analogous to social network analysis. As this section has shown, it is common to maintain the contact lists of users, or to examine a user’s set of choices in a recommender system. Many of these techniques could be strengthened by a social networks perspective. This chapter has illustrated a number of opportunities for future research: many currently-existing systems could be enhanced by either understanding and using network information, or by presenting it to end users.

This section has highlighted a series of features of networks that are worth recapitulating. Automatically-generated networks have been the basis of systems that are comprehensible by users and produce useful results. Awareness systems are notable for the idea of changing a display with current activity. While multi-modal networks seem to be rarer than person-to-person networks, the heterogeneous network of links between people and the documents they work with are implicit themes in systems that connect people with documents. Last, research into egocentric sources and person-centered views opens directions for future work with tracking and visualizing personal information.

2.2. Temporality in the Social Workscape

A second basis for this work is an increasing interest in the relevance and role of temporal patterns in supporting coordination and collaboration. Where social networks are a valuable tool to understand the structure of the social world within which users operate, temporal information can be used to understand patterns and how those structures change over time. Supplementing structural patterns with their temporal aspects is critical to making information relevant to the user *in situ*. Social networks are not fixed, but change over time, and these changes are relevant for the work that is carried out. Moreover, even in stable networks, patterns of activity and of relevance will change with respect to the rhythms and patterns of working relationships on scales ranging from hourly and daily, to monthly and yearly. Combining temporal and social information makes each more salient.

This section will attempt to present a perspective on time as a basic way of understanding and placing work and activity. There are longstanding traditions within computer interfaces of looking at time merely an ordering: as a way of assigning a unique ordering to each item in a computer system. These systems, though, take John Archibald Wheeler's old joke – that "time is what prevents everything from happening at once" – too literally. Time is not only a way of separating or ordering events, but points to the locations of those events. Experience has a *temporal texture*; events are *signposts* that break the uniformity of time and are *shaped* by the time in which they occur.

To expand on this concept of time, as texture for experience, I will draw on several different research directions. I will first offer a cross-disciplinary overview of how temporality has been addressed within the social sciences. I will follow this with a brief overview of how CSCW research has, as a field, approached temporality. I will then turn to the question of how time is organized, and will offer a framework for how time can be used in the interface, reviewing the interfaces available for time in both information retrieval and personal information.

2.2.1. Cross-Disciplinary Temporality

Although it has received relatively little direct attention in the collaborative systems community, the question of temporality as a fundamental feature of social life has been investigated by a number of social scientists. Much of the work was started by Sorokin and Merton (1937), who coined the term "social time" to distinguish between temporal organizations whose meaningfulness is socially constructed rather than being governed by nature and the wall-clock. These conventional notions are ways of measuring out socially-prescribed periods, such as the "semester" or "adolescence." While these periods may be measured out in clock-time (as Adam, 1988 usefully points out), their significance is independent of the figures on the clock: July 4th, to pick an example from popular American culture, is a very different day than the adjacent, but far less significant, July 3rd. Zerubavel (1985), for example, examines one such conventional social construction, the week. While various cultures have chosen to adopt the week as a standard unit, the significance they lend to it as a unit for organizing work, for demarcating holiness, and for scheduling life has greatly varied. In particular, he points out that Orthodox Jews structure their time around the fixed, weekly Sabbath. Indeed, Zerubavel argues that the observance of the Sabbath as a holy time, with its specific rituals, are a way that Jews identify and segregate themselves as a community: in other words, time is used as a tool for forming group identity.

One important aspect of time is as a structuring principle. The workday, with a conventional start and end, is familiar within a culture, but contains its own arbitrary choices. This culturally-defined rigor, as Zerubavel (1981, pg. 31) and Thrift (1988) point out of monks and peasants respectively, has been developing since medieval time, with increasingly-demarcated schedules. These schedules define a notion of work, and as such demarcate days precisely between work-times and leisure-times. Roy (1959) discusses how factory workers relieve the stress and tedium of the day by creating for themselves a temporal structure organized around seemingly meaningless events and rituals: breaks at "peach time," "window time," and "banana time," among others. These brief breaks are ways that the workers could reclaim a pattern of time to themselves, imposing their own schedule over a brutal schedule imposed by external work-shifts and assembly lines. Similarly, Fine (1990) describes the short-term temporal patterns at work in restaurant kitchens, and how kitchen workers interpret and orient towards them. In the kitchen, the dominant activity is one of synchronization: trying to ensure that dishes are ready simultaneously for tables of customers, trying to ensure that customers do not wait too long for meals, trying to keep busy enough to be active, but not so busy as to be overwhelmed.

Reddy and Dourish (2002) study information seeking in a hospital's Surgical Intensive Care Unit (SICU), and report that much of the information-related activity was oriented about the characteristic rhythms and trajectories of their field site. These rhyhtms include the daily schedules of the various hospital staff, time lags for test results to come in, and the developing conditions of the patients. Like Fine's kitchens, the hospital is an environment of carefully controlled synchronization: a test must be ordered early enough, for example, for updated results to be ready for the doctor on rounds. The SICU, however, also features longer-term management issues: doctors must manage cases to ensure that there is room in the SICU, should it be needed; patients must be circulated out of the SICU as their condition is stabilized to make room for future patients.

The converse of this short-term schedule is the longer-term calendar. Orlikowski and Yates (2002) examine the ways that organizations change and adapt. They look at a time period measured in months and years, the schedules during which new methodologies are adopted. While they show that the window for adopting new ideas can be surprisingly narrow, it is a narrowness measured in months. This article, too, casts a light on the importance of temporality in giving meaning to working activity in organizations.

Time has turned out to be an important aspect of small-group research. Similarly, teams and groups in organizations form and diffuse in sequences of distinct stages (Tuckman and Jensen, 1977, cited in Moreland and Levine, 1988). Groups *form*, as members establish an identity within the group. The groups then *storm*, in which members attempt to relate the group to their own needs, and *norm*, in which the groups converge on acceptable group behavior. Only then do they *perform*, accomplish their tasks, and last *adjourn*, or disengage from the group. While these stages can take different periods of time, and depend partially on the goals and composition of the group, the sequence is universal and identifiable. This suggests that is can be meaningful to talk about the "duration" of a group, or events that happened at different stages of the process.

McGrath and Kelly (1986) adapt the notion of "entrainment" from psychology to refer to the way that groups work to common schedules. Animals entrain to the schedules of their environments: they spend more time awake or asleep depending on the season, and can, under laboratory settings, be coerced to take on different schedules. So, too, team workers entrain to the schedules of their environments and groups. They choose conventional hours to attend work, for example, in order to maximize their opportunities to coordinate. Dramatically, during NASA's recent robotic mission to Mars, scientists took on a shared schedule of a slightly longer, and offset, day (24 hours and 39 minutes) in order to maintain synchronization with their experiments. In order to do so, they actually formed their own community in a distinct virtual time zone (Mallis, 2003). In each of these cases, a member's membership in a group was reflected in their schedules: a situation interestingly parallel to Zerubavel's Orthodox Jews.

2.2.2. Time in CSCW Research

Within the last several years, CSCW research has begun to develop an interest in the importance of temporality in collaboration. The growing body of work has begun to examine how people manage their time and their reachability, and what important temporal factors drive their schedules.

Hudson et al. (2002) explore patterns of availability and accessibility amongst managers in a research organization. By querying their subjects with brief survey questions across different times of day, they try to find how managers distribute their time. The research finds some degree of uniformity between different managers, including common patterns of when might be the best time to interrupt them. It also found a strong preference not to be interrupted during meetings and gatherings.

This work might be contrasted against Begole et al.'s (2002, 2003) work on work rhythms. In both papers, the authors use automated techniques to build up more detailed information about daily activity structures for specific individuals. The activity they examine, that of being active at the keyboard of their computer, is easily collectable data. From it, the authors develop a picture of when the user is likely to be available. Users fall into characteristic patterns, and so they are able to present a tool that predicts both future *availability* and its converse, the time at which point the user is next likely to become unavailable. These are, of course, probabilistic measures: but they provide cues to personal behavior. Tyler and Tang (2003) use similar techniques to predict when a user may expect an email response. They find that past measures of responsiveness are good predictors of future messages. Indeed, while a person will often control their responsiveness for different correspondents, they will often be rather uniform for each correspondent. In these cases, then, people are carefully projecting a responsiveness image, trying to ensure that they are perceived as the right balance of available and important.

These studies, and others, point to the significance of temporal patterns in coordination, and begin to provide a deeper understanding of their role.

Social activity is inherently poly-rhythmic, incorporating activities that happen on a variety of schedules. For example, Begole et al. looked primarily at daily cycles, focusing on when workers were likely to be in or out of their offices. This research studies broader rhythms, too: collaboration groups can shift month to month; over the duration of a year, members of a team may work with different groups. Previous work has suggested that the time scale of email conversations shows changes in social networks; over the duration of an online conversation, additional participants are gradually incorporated or removed as the topic becomes relevant or irrelevant to them (Fisher & Moody, 2002).

2.2.3. Temporality in the Interface

Because of its strong social cues, time also is an important cue for memory. We mark our interactions with each other on a social calendar more than a paper one. Influenced by the annual publication cycle, researchers might try to wrap up

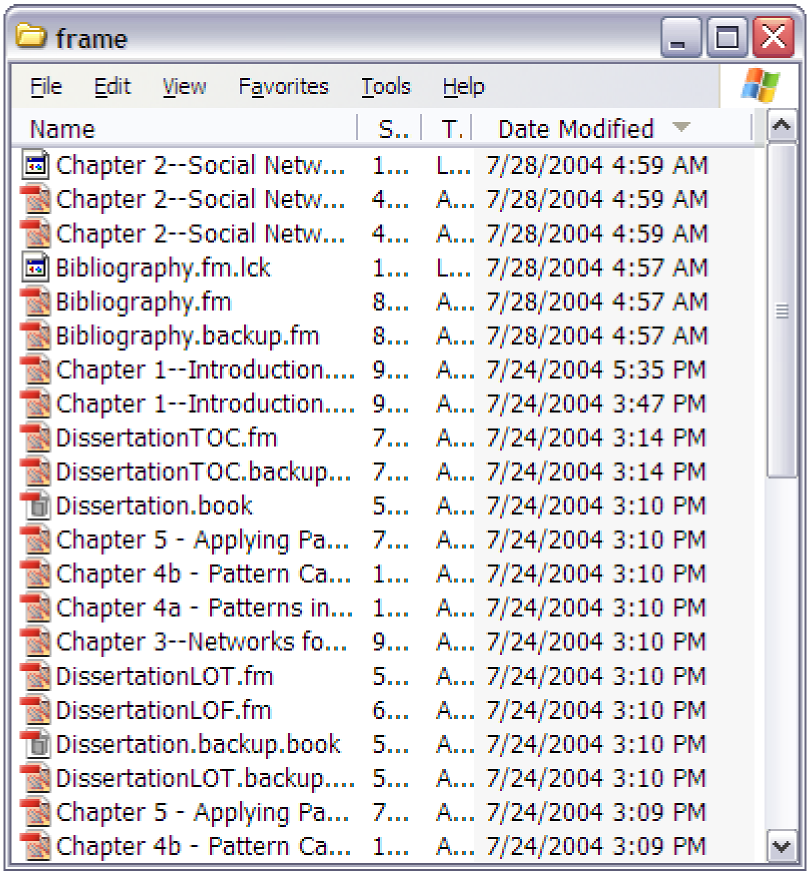

Figure 2-9. Time as ordering in the file system.

aspects of work on a project each year in mid-winter; with work patterns and collaboration bursting into activity in early spring as manuscripts are prepared for publication.

Time sits, however, uneasily in the user interface. In most common operating systems and email clients, time is portrayed solely as an ordering: resources are lined up in sequence by time of creation, or last modification (see figure 2-9). While this is an efficient form of display to minimize space, it does little to clarify the social or mnemonic aspects of the schedule.

This issue has been addressed in several alternative interfaces. In particular, past research projects have attempted to portray the operating system or a group of active documents along a time line.

"Lifestreams" (Freeman and Fertig, 1995) is an original approach to reorganizing the user's workspace around temporal information. In the Lifestreams project, the user's complete history (their file system, their email messages, and so on) are arranged in a linear stack on screen. Users may then access past documents by scanning through the pile in order. While the pack is linear, it supports an ability to implement full-text search. Searched entries are then highlighted and separated slightly, providing a cue to users where in time the events that involve those words happened.

A related project, TimeScape (Rekimoto, 1999), also attempts to present the user with documents keyed to their history. A user places their current documents on a modified desktop, and removes them when they are no longer current. Later, the user may "travel back" through time to see the desktop as it had been at any past time. With this system providing flexible control over time-indexed files, the user may easily recall the time associations of past files. Note that in this project, a user can scroll through past events, seeing the configuration as a cue to what events were happening and what information is available. This project might be compared to the PARC Perspective Wall (Mackinlay et al., 1991). The Perspective Wall presents documents along a timeline; users may arbitrarily place documents vertically on the page. Like TimeScape, the Wall then allows the user to scroll forward and backward through time to see different parts of the data's history. A third, closely related project are "Space-Time Maps" (Chiu and Trong, 2002). These maps attach a timeline to a two-dimensional map; resources placed on the map enable the user to triangulate a desired object in time and space. Each of these examples shows a way of using the objects placed in time acting as both *signpost* and the *goal*.

A third important temporal visualization is the Milestones interface for the recent Stuff I've Seen (Ringel et al., 2003). Stuff I've Seen (SIS) is an engine that tracks and indexes a user's files, emails, and web pages in order to allow them to access the information more quickly. In the original SIS interface, query responses are returned in several possible orders, including by date. In contrast, the "Milestones" interface to SIS attempts to remind the user of the context of documents by displaying memory cues. Rather than presenting an ordered list, it presents a timeline; next to elements in the timeline, it shows important news headlines, elements from the user's calendar, and entries from the user's photography archive. Milestones provides temporal cues that align with the user's own experiences. This interface is a more refined version of the "Dynamic Timelines" (Kullberg, 1996), which placed photographs next to an interactive timeline as a browsing interface. This system uses two different sets of objects: the set used as cues (the newspaper headlines, the pictures, the calendar entries), and the set used as the targets of the query. The former set is used to orient the latter.

So far, I have discussed visualization systems that work to replace the user interface to the file system, or that display responses to queries along timelines. It is worth highlighting several query interfaces that reflect changes over time that deal with different types of datasets. Both “HistoryFlow” (Viegas et al., 2004) and “ThemeRiver” (Havre et al., 2002) present linear views of the ways that text develops over time. In ThemeRiver, a body of time-indexed text (such as newspaper articles) is examined for multiple hits on important keywords, and “bends” of the “river” are drawn more broadly when that theme is more prominent in the dataset. If the corpus is time indexed, than it becomes a tool for monitoring changes in themes and topics over time. Where ThemeRiver looks at a timestamped corpus of information, HistoryFlow is a visualization that shows revisions and changes to a single document over time. It vividly shows at what stages the document has changed, and labels the parties who did the changing. In both systems, the metaphor of the “texture” of time is given a more literal definition.

As yet, these visualization techniques have portrayed a linear sense of time: time progresses serially, from early to late, along a straight line. As I have pointed out above, however, time is not merely linear: rhythms and cycles are important aspects to make visible. Carlis and Konstan (1998) suggest placing time-indexed points on a regular spiral. In contrast to traditional linear visualizations, in which rhyhtms and recurrent are infrequently visible, the spiral format allows recurrences to be aligned. For example, a spiral with a frequency of one week makes regularly-scheduled appointments visible. This unusual visualization helps emphasize the cycles that featured prominently in a great deal of the research discussed above, from Fine’s kitchens to Reddy and Dourish’s hospital.

2.2.4. *Final Thoughts on Time*

This section has attempted to present an idea of time as a way of structuring information. It has discussed ways that events in the past can be used as cues that point to the social context, and the ways that a social context influences the use and understanding of time. Within the social science studies and the CSCW work, time is seen as a way that people structure their lives and work. Among the user interfaces, time is presented as a way of experiencing reality. In all of them, time mediates the experience of people.

While time is often seen as a simple ordering within computer systems, this section has pointed to a number of projects oriented around placing both user and external events on various types of manipulable perspectives and timelines for users. While most of the systems are linear, nonlinear ways of portraying cycles also bear promise, especially in light of the work suggesting that much of the way that workers do their work is organized around these cycles.

2.3. Conclusion

This chapter has laid out two toolsets that will be used to analyze the problem presented in the first chapter. The first tool, social network analysis – and, in particular, the notion of the egocentric network – can be used to explore personal networks; these networks can produce results that can be fed back to end-users. The second tool, temporal patterns, provides an important dimension to understanding how groups and interactions change

Both of these perspectives are ways of understanding and analyzing the ways that people interact. Social networks are a way to translate the subtle interactions between people into a usable, analyzable, and visualizable form. These networks, when captured, can display a variety of interactions. The first portion of this chapter showed a selection of views of group interaction. It presents technologies that both explicitly and implicitly present networks to their users. Throughout, the section convergeed on tools that are used to visualize interaction with an underlying sense of the network. It found a variety of presentation tools that show the network or a processed version of it. This lends credence to the general intent of this discussion: to first develop prototypes that allow us to prototype network views, then to design tools that *use* the network information, not necessarily directly.

The section on temporality introduced the importance of the social study of time as a research direction. It pointed out the uses of temporal research in small-group research, and the values of temporality in the workplace. From this work, several themes emerge: people maintain schedules in the short term (hours and days); groups and organizations move and develop over a longer term.

The temporality section also criticized contemporary computer interfaces for presenting temporal information too sparsely. It pointed to a variety of query systems and user interface views that are directed around the idea of time, and began to collect their insights as a direction for design.

In the next chapter, these two analytical perspectives will be combined and applied to design a system, “Soylent,” as a tool to understand individual interaction histories.

Chapter 3. Visualizing Communication

In the previous chapter, I looked at a broad variety of ways that social networks and temporality can be used to interpret and present a variety of individual and group activities. In this chapter, I begin to consider how they can be applied to the topic at hand: how they can be used to interpret and make visible the traces of activity left in everyday interaction.

In this chapter, I will describe one way of reconstructing the social networks and temporal rhythms are assembled from traces available in the computer system. It discusses an infrastructure for data collection and visualization tools collectively known as Soylent. It will first discuss general issues in the design of a data collection mechanism and a visualization tool, before focusing specifically on the implementation of the Soylent system, looking in some detail at the set of tools implemented and studied in this project.

3.1. Designing a System

This section deals with the general design of a way to handle electronic traces: it attempts to broadly discuss the design space and possibilities, and to explain the rationale behind the Soylent system. It first discusses ways of collecting and storing electronic communication traces, and then moves to a discussion of how to present and visualize them.

3.1.1. Choosing a Corpus

The first hypothesis in Chapter 1 stipulates that patterns of contact and collaboration leave meaningful electronic traces. Those traces are derived from social activity in computer systems; the challenge is to show that those traces are potentially and actually meaningful, and that they come in sufficient quantity to provide meaningful data about interaction. Traces are generated by user activity: sending and receiving email messages, for example, each generate substantial amounts of data in both the messages themselves and their header information. Examining the attributes of these messages – who wrote them, when, and to whom – are cues to the social shape of the users' workspace.

These are not the only traces available to computer systems. Other sorts of communication acts are recordable and usable as information about social context: instant messaging (Begole, 2002), voice mail (Whittaker, 2000), and even physical movement sensors (Hudson, 2003) have been used as cues to understand the dimensions of interaction. Implicit communication, through file sharing (Gummadi et al., 2004), code editing (Froehlich and Dourish, 2004), or collaboration around shared artifacts (Muller et al, 2004) are all possible areas where meaningful traces can be interpreted and examined.

The traces examined in this project are the *headers* of *archived, outgoing email.* This choice is based on a combination of the value that can be obtained from email, the cleanliness of outgoing email (as opposed to incoming), the detailed social information available in the headers, and the convenience of accessing mail archives.

Previous research (Whittaker and Sidner, 1996; Fisher and Moody, 2002; Tyler and Tang, 2003), has shown useful results based on email activity. While email records cannot fully cover all communications between a pair of people, they do present a view of workplace communication importance: certainly, in some organizations, email is a communication form of record, and as such is used to maintain many important contacts.

Archival email stores have the advantage of being consistently recorded and, for many users, thoroughly collected (Whittaker and Sidner, 1996); therefore, it is possible to look at broad interaction histories. In contrast, monitoring current email activity requires the adoption or installation of a prospective data collection system. Such systems provide a much more limited range of time available for study, and require more sustained work by the users. On the other hand, many users keep very limited mail archives, and so have more limited data.

Email *headers* contain specific social information. By definition (Resnick, 2001), each email message is labelled with a unique identity, a time stamp, a sender, and a list of receivers. Conventionally, email messages are often sent to a group of concerned recipients, sometimes distinguished by their presence on a "to" or "cc" ("carbon-copy") line. While a majority of messages are sent to only one recipient (Fisher and Moody, 2002), a substantial number is still sent to a larger groups. The header information, then, is sufficient information to reconstruct social network structure as well as to label messages with their temporal aspects.

While both outgoing and incoming mail contain valuable information, the *outgoing* mail has some particular advantages. Incoming mail is used for a wide variety of communication tasks: routine announcements, mailing lists, spam, and other conversations can be found in incoming email. A substantial body of research has gone into trying to figure out which email is relevant in which context, and how to separate it out (Sahami et al, 1998, among many others). In contrast, outgoing mail has the particular property that it is all relevant to the user: it was mailed presumably because it was important enough to send.

3.1.2. Visualizing Mail

The intent of this system is to uncover users' social structures and patterns of contact and collaboration. One possible strategy for doing this might be to apply techniques drawn from machine learning and data mining research in order to analyze collaboration information and uncover the statistical patterns that characterize it (e.g. Getoor, 2002). This system, however, is designed to make activity visible in terms that make sense to the user. The underlying statistical calculations are valuable only as far as they are recognizable to the users whose activity they describe. Unfortunately, while many statistical interpretations may be accurate, they may not be salient to the user.

Instead, this work presents a visual approach that may be more recognizable to users, and allows users to apply their own interpretations. The intent is to provide a range of ways to display and process email records, each of which is designed to highlight temporal and social structure in the stored communication. In particular, I will emphasize a display that is primarily intended to present interconnections between people, and a set of displays that try to display several types of temporal information.

3.1.3. Building Social Network Diagrams

The first visualization is the "association net." This net presents an aggregated, egocentric social-network view of the users' workspace based on outgoing messages. It is intended to help understand the structure of the relationships in the user's communications. If it is true, as suggested above, that users send email to sets of correspondents who are logically clustered, then a tool that draws out these clusters would be a visual way of understanding the general shape of the communication space.

Most social network studies have examined collective data. Network visualizations of email (such as Eveland and Bikson, 1988; Tyler et al., 2003) traditionally examine pairs of names, tied by who sends email to whom. These techniques provide a collective and global view of email records, and are analyzed using a "to-from" approach, drawing directed links between sender and receiver. Those social networks are used to connect a great many people together, and provide a broad view of how people are connected. A tie in such a network diagram from A to B is interpreted strictly as "A has sent email to B," and loosely as "A connects to B." Edges, then, connect paths of communication: one might read an indirect link as a way of sending a message through an intermediary. For example, a connection from A to B to C might suggest that the best way to get a message from A to C might be through B.

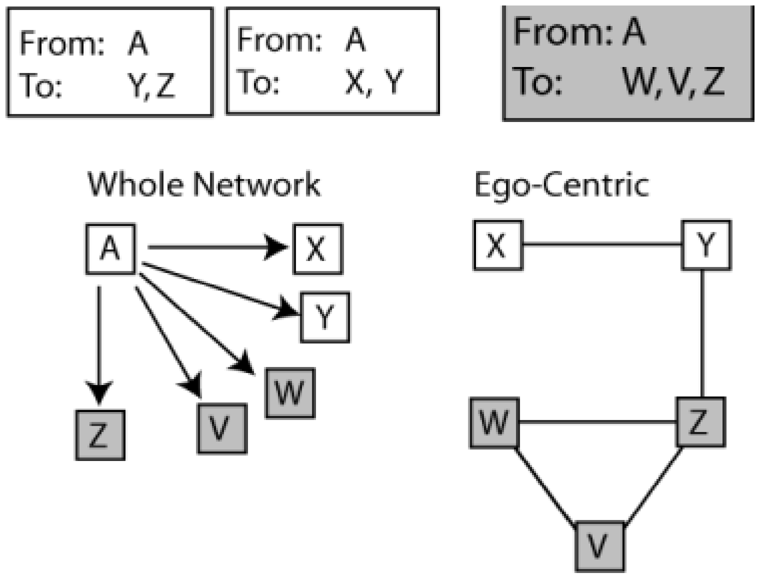

Figure 3-1. The Soylent egocentric networks.
This diagram compares the Soylent network with a traditional sociometric network. A sample set of three messages from A is shown.

A different approach within social network analysis is the notion of "egocentric" networks. Egocentric social network studies (Wasserman 1994, Wellman 1993, Newman 2003) typically connect a single person (referred to as "ego") to a set of "alters" around them. Unlike the broader sociometric studies, egocentric analyses are interested in relationships between the ego and its set of immediate alters.

My interest is to understand a *single user's* workspace. As such, the egocentric approach appeals: it requires no more information than is in that single users' mail. A diagram showing interconnections between the user's correspondents gains a nuanced view of that single user's interactions. Within this project, and as in boyd's (2002) work, a message co-addressed to two different persons, whether via a "to", "cc", or "bcc" line, is understood as implicitly tying those persons together: the sender believes that they share an interest in that message. This mechanism is shown schematically in Figure 3-1. This network diagram therefore ties these two people together. The tie from X to Y is read strictly as "the user has sent the same message to both X and Y," and loosely as "the user sees X and Y as connected." This is a statement about the *user*, not about the true state of the world: X and Y may find each other of no interest whatsoever. In these networks, it is difficult to read an indirect path. In these egocentric network, where the user links X to Y, and also links Y to Z, the best way to get a message from X to Z is, in fact, through the user him- or herself; Y is not really a bridge between the two populations.

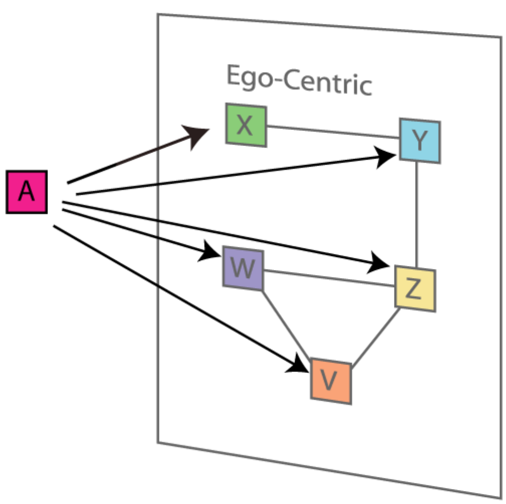

Figure 3-2. Showing the ego. The egocentric network from figure 3-1, enhanced to show A (ego).

Note that the user is not visible in the diagram in Figure 3-1. The user, were he or she to be pictured, would be somewhere outside of the image, connected to *all* of the nodes in the diagram. Figure 3-2 clarifies the location of ego, A: outside the network entirely, and invisibly connected to everyone who is visible.

The frequency of co-occurrence gives a notion of how strongly tied two users are. This dissertation will occasionally speak of a link that has been multiply reinforced through many messages as a

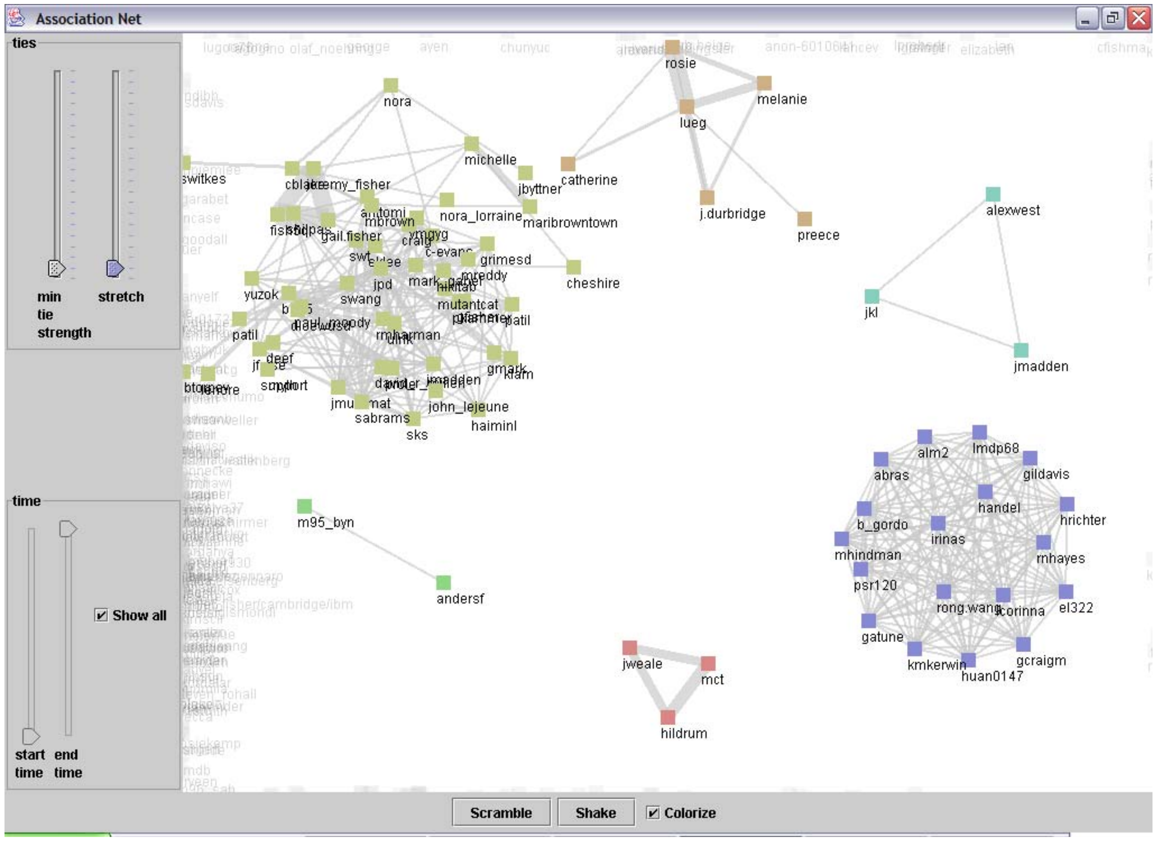

Figure 3-3. Soylent's Association Net view.

"strong" or "close" tie; in contrast, pairs of people who are connected by few messages are referred to as "loosely" connected.

Figure 3-3 shows one association net for email data. The archive shown belongs is my own. Contacts are represented by a short name and a colored box; lines are drawn between linked names. Very infrequent connections have been filtered out; pairs of people must have been sent three or more messages to be shown here. Not all names are reachable from all others, so each connected component is shown separately.

Such an association net could be assembled for both incoming and outgoing mail. With incoming mail, a tie from A to B can mean that A sent a message to the user and B, or vice versa, or that some other party sent a message to A, B, and the user.

3.1.4. Viewing Temporality

There are a number of different perspectives on temporality suggested by the work in chapter 2. A system can attempt to bring out recurrent rhythms, trajectories, and changes over time at a variety of different time scales: brief bursts of activity, long-term patterns, and others in between. In practice, the workplace is regulated by its own rhythms; this project takes advantage of the conventional entrainment to days, weeks, and months to help choose default time scales for the views that will be discussed.

A popular tool from the natural sciences, an *actogram* brings out recurrent habits. Traditionally used to examine sleeping and feeding cycles for animals, our approach visualizes mail-sending activity for humans, instead. In an actogram, events are plotted hourly on one axis and daily on another in order to compare day-to-day activity.

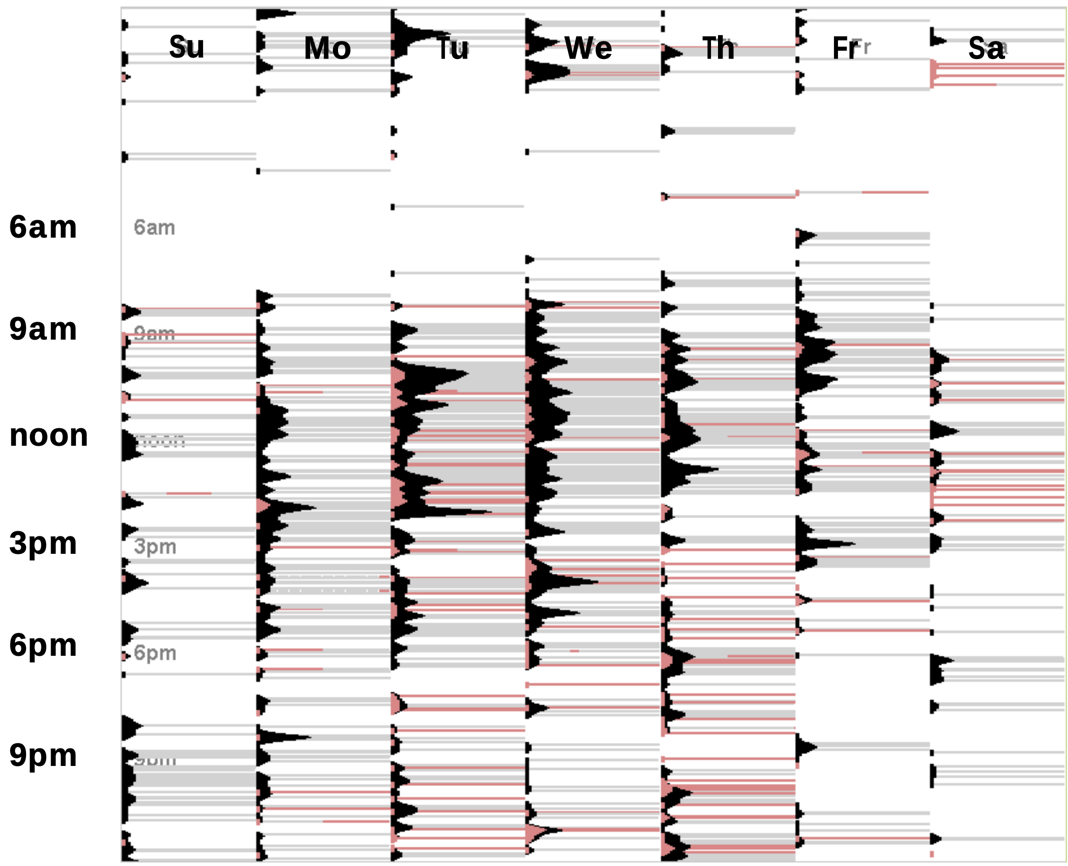

Figure 3-4. Time Display.
Days of the week proceed horizontally; time of day is shown vertically. Small gray bars indicate patterns of contact. Chart shows 6 weeks worth of information.

One such chart is shown in figure 3-4. Each thin bar indicates a message; the bars are placed horizontally by the day of the week, and vertically by the time of day they were sent. Multiple days are superimposed in order to build up a cumulative picture of longer-term activity: in this case, seven days of the week are repeated, so that this cumulative picture shows many weeks. Filled black curves on the left side indicate cumulative counts of messages: effectively histograms. A personal name can be associated with this chart, too: the name is set to a color as used as a highlight. In addition, histograms on the left side are accumulated to indicate cumulative contact with just the one person.

Some broad patterns in this data that begin to suggest ways in which this information might be usefully mined. For instance, we can see a rough structure to the day; at the least, it is clear when it begins and, to a lesser extent, ends. The user can be characterized by preferred sleeping habits, likeliness to be available on weekends, and unlikely times to be reached. By aggregating information over many weeks, we can build up an even more detailed view of daily activity.

This aggregated display is very similar to (and in part inspired by) that of Begole et al (2002), who generate activity graphs based on idle time information on workstations. There are two important ways in which this information extends the analysis that is possible using Begole's data, however. The first is that the information we are capturing here is directly linked to semantically meaningful data; this contains information about patterns of communication to specific individuals, rather than simply idle time. The second is that, since this is also integrated with social network information, we can begin to make generalizations about temporal activity that concern not only single individuals but larger social groups. This allows us to differentiate between the different roles and different activities in which people are involved.

A second way of viewing temporal changes is to look at long-term activity. While the actogram above obscures changes over time by focusing on recurrent short-term patterns, other views are designed to show long-term changes instead. Long term activity can be shown with a visualization that compares behavior over periods of time. One way of doing this is to focus on the *cumulative* behavior in the data, by comparing the activity in one time sample to another. A converse approach is to focus on *individual* behavior, by looking at the pattern in communication with a single other person.

The cumulative form gives an idea of the user's own behavior. It gives a sense of their shifting communication priorities: what people were recipients of the most communication in that month? While this may not be a perfect proxy for importance, it does provide a snapshot of that months' interactions. Comparing month to month yields information on changing events in life: a class or project ends; a new person becomes involved.

The individual approach to long-term behavior is to examine interactions with one person, repeatedly. It presents a view of how a single relationship has changed over time. As such, it can highlight both specific events (which might cause a burst of activity), steady states, and general trends toward increasing or decreasing communication.

3.2. The Soylent Implementation: Automated Traces

The designs discussed above are formative steps in developing a series of visualizations centered around records of social activity. In this section, I discuss "Soylent", a tool designed to collect and display social information.

3.2.1. Collecting a corpus

In order to confirm the hypothesis, it is necessary (although not sufficient) that there be enough information in the traces to reconstruct meaningful interaction patterns.

Table 3-1. Message Census for Three Sample Users

	User 1	**User 2**	**User 3**
Messages	42855	3333	7831
Outgoing messages	16919	3175	7831
Outgoing messages with > 1 recipient	2101	285	1146
Dataset duration	March 22, 2001 to January 19, 2004	April 2, 2001 to March 24, 2004	October 7, 1999 to November 11, 2003
Distinct outgoing addressees	1709	848	2778

This table lists some basic statistics about several users' archives collected with the Soylent system. Note that users 2 and 3 stored only their outgoing mail in the system, and so their "message count" and "outgoing message count" are close to the same. In User 1's case, and in many others, outgoing mail makes up noticeably fewer than half of the messages in the archives (in most cases, around one-third). Most outgoing mail is sent to only one person: outgoing mail sent to more than one recipient, the messages that can be shown on the association net, are a small fraction of those. However, in all these cases, that subset was enough to show network connections.

Note that even over a three year period for User 1, only two thousand messages were sent to an outgoing list of more than one person. Clearly, then, one limitation of this work is that the number of outgoing messages is fairly sparse; the groups need to be built up over time. This may cause problems with this technology in some settings. For example, some organizations have policies of purging mail messages older than three months. This would be problematic for generating long-duration histories of interaction.

3.2.2. The Soylent Data Collector

Soylent collects information from electronic mail activity, but is designed to record other sources of information flexibly, and to integrate across them (see *Technical Appendix section 1*). In the implementation discussed in this dissertation, Soylent gathers information from stored email record headers: the system records both incoming and outgoing messages in the database. This infrastructure builds a collection of interaction records. Each record contains a single message, which connects a single message sender to a group of receivers at a particular time and date. Each message is also associated with a group of message attachments, a series of file names that may have been attached to the message.

Table 3-2.Data stored by Soylent (and corresponding RFC 822 fields)

- Unique message ID (Message-ID)
- Message sender (From)
- Message recipients (To, CC)
- Date and time of message (Date)
- Attachments to message

The diagram in figure 3-5 schematically suggests the organization of the Soylent system. Specific headers of email messages are collected from an email archive and stored in the Soylent database; the database is mined at a later point to extract the visualizations.

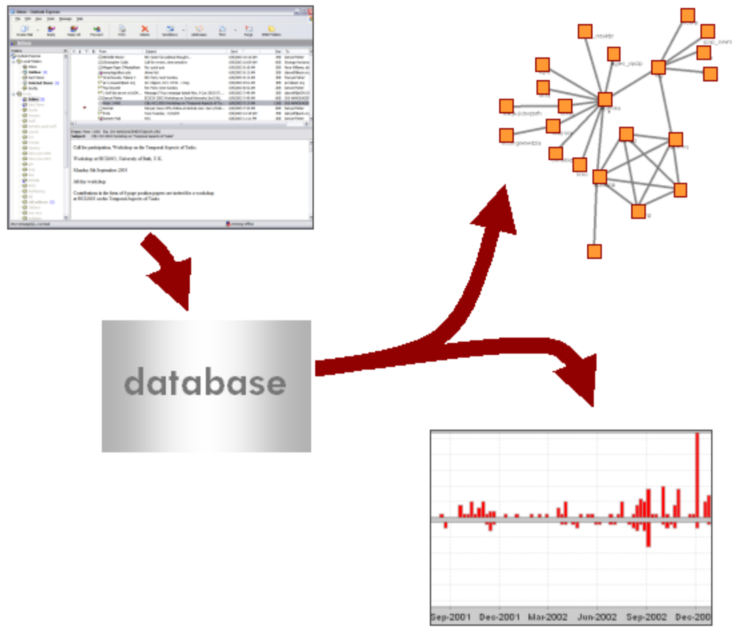

Figure 3-5. Soylent Architecture
A mail client feeds information to a database, which is visualized.

Soylent has been adapted to read from a variety of mail sources, including SMTP mail logs, IMAP mail servers, and Lotus Notes mail stores.

Soylent also contains a mechanism for updating the database with more recent messages; a user can supplement messages already in the database with additional entries. In theory, then, Soylent can maintain a near-real-time message store, updating with additional messages as they arrive, and thus provide up-to-date information about messages.

A single installation of Soylent, then, is a way of storing information about, and then examining, a single user's email. This chapter will also discuss a visualization framework built over Soylent that shows a number of different views of the underlying data.

The data is stored in a relational database, and so can be queried in a variety of ways: information can be collected centering around people, around messages, and around temporal sequence. In this project, the database was dominantly examined as a bipartite graph connecting people to messages, as a time-ordered sequence of interactions, and as a time-ordered interpersonal communication log. The Soylent application provides an API that allows several of these perspectives to be accessed (see *Technical Appendix 3*).

3.2.3. The Soylent Visualizations

All Soylent views are linked with each other; manipulations in one view (selections, colors, and other dynamic properties) are reflected in others. This makes it easier to trace patterns and see how, for instance, structural patterns are related to temporal ones. As a result, users can explore an information space that is richer than the space visible in any given view.

Each view provides a range of ways of exploring the data from a particular perspective. All of the views have a number of attributes in common: they connect *people*, *messages*, and *time,* by laying out these attributes on different dimensions. For example, the "association net" view connects people, holding constant a particular time range and number of messages. In contrast, the "TextArc" view maps people against time, and holds their messages constant.

The views are designed to be used together; they operate in concert and allow an analyst to triangulate on the data and highlight connections between different perspectives. This triangulation is accomplished largely by maintaining color consistency on the names that the system displays: when a person's name is colored in one view, they are re-colored in parallel in other views. In this way, individuals and groups can be colored in one view and traced through to other displays.

3.2.4. Association Net

In the Soylent "Association Net" visualization, the database is queried to collect pairs of names that co-occurred in outgoing messages. (The specific labels "to" and "cc" are disregarded in this visualization). Linked entries are laid out with an animated spring-embedder algorithm (Eades, 1984); the algorithm ensures that closely-connected clusters bunch together, while disconnected clusters generally slide apart from each other. (The algorithm is adjusted so that disconnected clusters arrange themselves in a rough circle around the screen). The thickness and compactness of the lines between any pair of names indicates the frequency of co-occurrence in the data set.

Because the Association Net visualization is meant to explore relationships between sets of people, names of individuals that are not linked to any other names are not shown.

A screen shot of an Association Net based on my data can be seen in figure 3-3, page 72. The groups that emerge are recognizable as clusters in my history. The large cluster at the lower right are participants in a workshop, for example, while the group at top middle is the set of people involved in a publishing collaboration: a co-author connects to contributors, an editor, and a publisher; the editor and the publisher are strongly tied to each other. The large cluster at the left is a tightly connected cluster of work and social interactions..

3.2.5. *User interface*

The association net can be used to examine networks across the duration of collected records, or can be focused on specific time periods. The view can also be adjusted to show only strongly-tied users, and to ignore weakly-connected pairs.

This section briefly highlights all the major manipulations that could be done on this view, separated by user tasks:

3.2.5.1. Data Filtering Tools

Data filtering tools allow users to examine the data at both different *tie strengths*, and across different *time ranges.* By adjusting sliders, users can remove or add data to the screen. Data that is filtered out would turn gray and "drift" off screen; data that is filtered back in would darken and drift back on-screen. The drifting is an animation in order to make changes clearer to the users of the package. The options available are:

- *Filter by tie strength.* At top-left is a slider for "minimum tie strength"; this is the minimal threshold for the number of messages that two users needed to share in order to be shown on screen.
- *Filter by date range.* At bottom right are starting and ending dates for a time-limited slice. Messages that were exchanged out of this visible range would be

hidden; thus, it would give an image of interactions over a given period.

3.2.5.2. Handling the Display Space

The display can easily become congested with several different disconnected components. In order to reduce crowding, the user has several different tools: they can remove connected components, or push nodes further apart from each other. In addition, users can color nodes by cluster in order to help make it clearer what the distinct clusters are. These options are:

- *Hide a particular cluster*. A single connected component can be hidden, and completely disappears. Each of its nodes would be marked as "hidden"; even if the tie strength was weakened (and so the cluster should show), those nodes would continue to be hidden. Conversely, all nodes *except* the current cluster can be hidden. There is also a function for showing all nodes.
- *Changing the spring constant* allows nodes to stretch apart further. In particular, it relaxes a dense cluster, allowing the nodes to slide a little further apart.
- *Drag or freeze individual nodes* allows the user to pull nodes in particular directions or to freeze one node at a given location.
- *Color all nodes* by their cluster tints all nodes to the same color within a given cluster, and attempts to label each cluster with its own distinct color. This node-level coloring is persistent across changes in showing nodes. Conversely, the user can reset colors on all nodes to turn them all back to the basic gray.

This cluster of functions are all intended to give the user better access to the information conveyed by the visualization. They do so by allowing the user to view specific times or strong connections only, to separate dense sections, to hide overlapping information, and to color distinct quadrants.

3.2.6. Temporal Maps: Time Display and Top Ten List

The second Soylent display is the temporal view. The Time Display shown in figure 3-4, page 74. This image is a log of the dates and times of sent email messages. While messages are, by default, indicated with gray bars and black "cumulative" marks, the view can highlight interactions with a single person. This information can indicate routine communication schedules with another person: for example, an emailed afternoon progress report to a team would show prominently.

The top-ten list is a comparative, long-term view of temporal shifts. It was an additional feature designed to show changes in individual interaction over time (Figure 3-6). It shows the top ten correspondents in a mailbox from top to bottom for each month: the most frequent correspondent is at the top, while less-frequent correspondents are further down. The top-ten list shows changes in

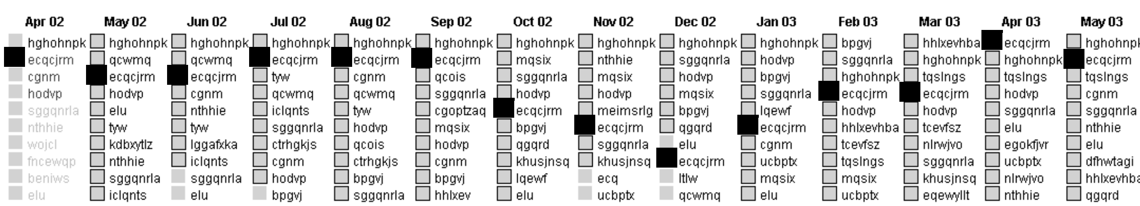

Figure 3-6. The Top Ten List.
Each month is represented by a column of the top ten outgoing contacts. One name is highlighed across the list.

interaction clearly. Presence at the top of the list is not a perfect indicator of importance, as a single burst of messages could be enough to top a month; however, transitions up and down a list from month to month can highlight meaningful changes in salience.

The top-ten list is instrumented to make individual names selectable. When a name is selected, it turns to a contrasting color all the way across the display. In this way, it is possible to easily track a single person's changes in relative position across the months displayed.

The top ten list provides some additional data about absolute numbers of messages. A name is drawn with a *black-outlined box* next to it if the recipient was sent more than 25 emails in the month; the name is written in black text if the recipient was sent more than 10 messages.

The top ten list ignores co-recipient information; all messages sent to a single address are ranked equally, whether they were sent to a single person or to a list.

3.2.7. Other Soylent Views

To give a general sense of the sorts of information that users could experience while using Soylent, we briefly describe some of the other Soylent views. One of the earliest displays was a "group display" that showed all the people who were co-sent a single message. The display placed a selected message at screen center, and then would organize other messages by the number of names they had in common with that message. The degree of darkness of a

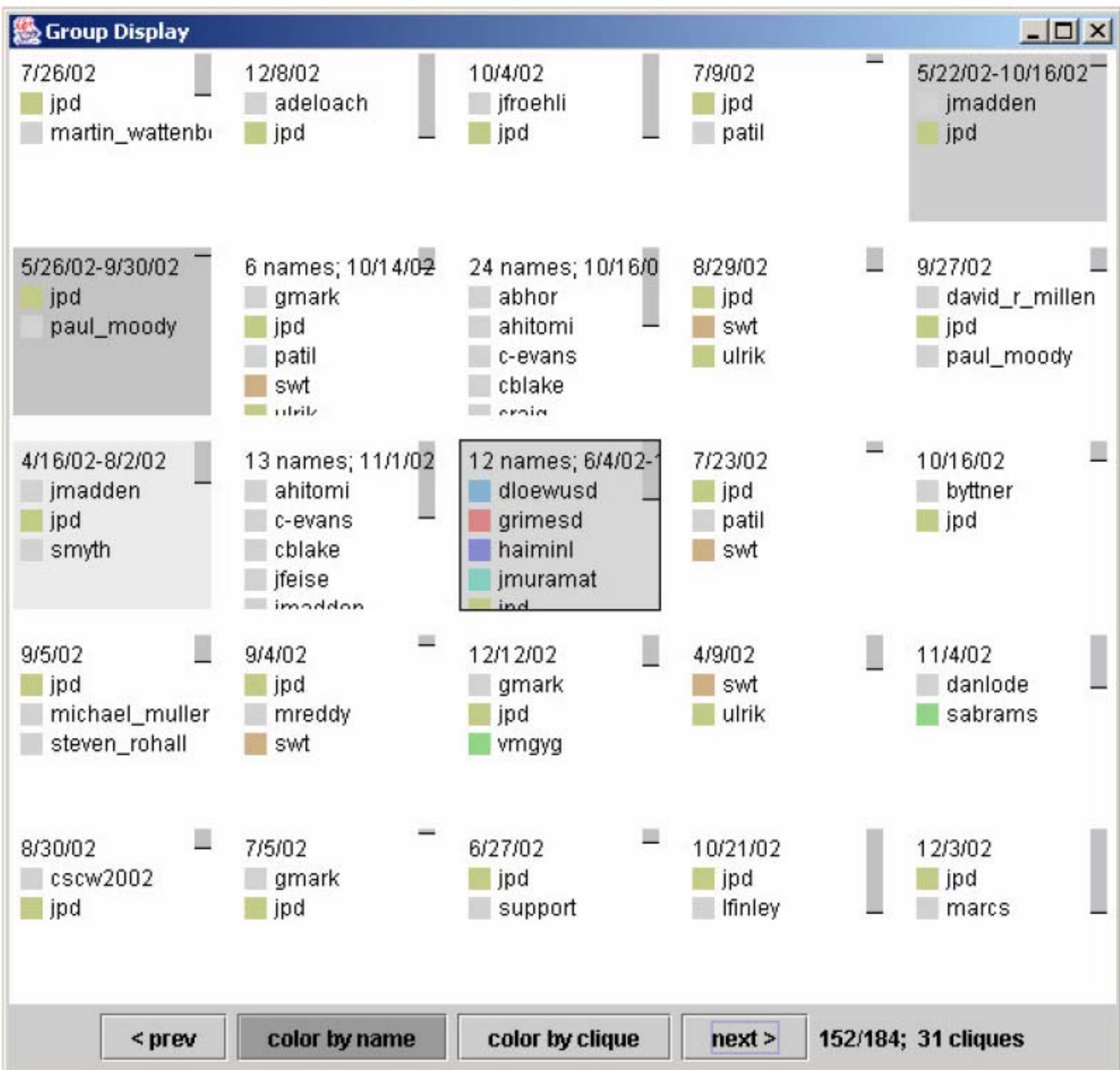

Figure 3-7. The Group Display.

The group display shows groups with overlapping membership to a central group. Boxes next to overlapping names are colored; other names are left in

square indicated how many messages had been sent to that exact list. In this manner, overlapping clusters of names were visible near the center of the screen. This visualization (Figure 3-7), which shows clusters of users, but obscured individual roles, was meant to help find common attributes of group co-membership.

In early phases of the research, the groups display was important for developing certain insights about the texture of interactions between users. For example, people who were involved in many overlapping contexts (see pattern Nexus on page 116) would have many groups around them. However, users found the display both confusing and uninteresting; it was dropped during user testing.

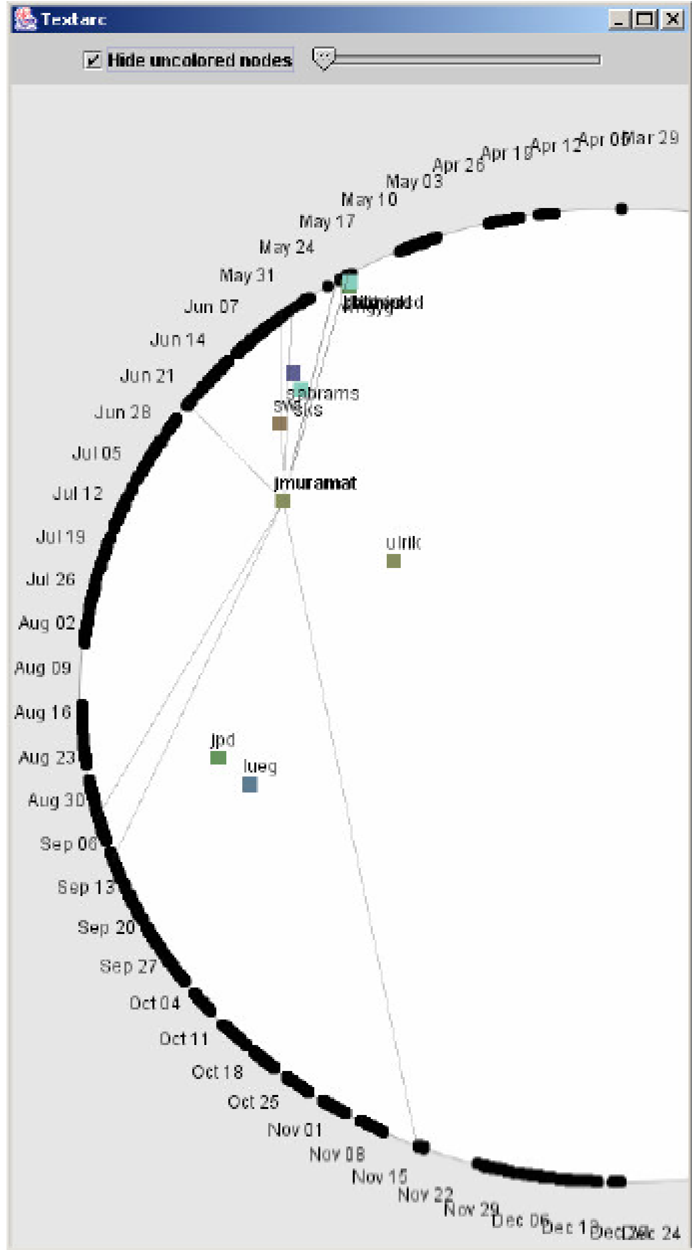

Figure 3-8. TextArc-based view.
Time proceeds around the half-circle from top to bottom; the selected user was contacted in a number of messages in late May.

A second view was based loosely on Paley's TextArc model (Paley, 2002). The design spreads a list of messages across 180 degrees of arc, and then plotted names at the spatial average of their dates. Vertex colors were tied to the other views, so a user could color a cluster in the network display, then show only those colored names in the TextArc display. The TextArc proved to be somewhat useful in showing the distinctions between the clustered names in the social network view: individuals who had a distinct role from a group would pull away from the groups in which they were involved.

3.3. Challenges in Soylent

In this section, I outline several of the technical and conceptual challenges associated with the Soylent system. The section first addresses issues with privacy, and then discusses other, more technical design trade-offs that were addressed in Soylent development.

3.3.1. Privacy Issues

Clearly, automatically gathering and analyzing traces of individual activity poses significant potential for invasion of privacy. It is important to note two things here. First, our system is constructed so that each user runs an application which gathers information purely on their own behalf and purely for their own use; the database in which records are stored is generally a private database for each individual, and information about each person's activity is shared with no others. Soylent takes an "egocentric" approach to data management and visualization, providing users with access only to their own data.

Second, users only see information about their own activities; information about one user is not combined with information about any other (and, indeed, to do so would be to defeat the object here.)

The technical privacy risks, therefore, of using Soylent are no worse than using a standard email client; it merely offers a different presentation of the same information.

Certainly, there are other privacy concerns. Users have asked whether this *accumulation* and *presentation* of data in visual form is not, in itself, a privacy risk. There is a danger that someone might casually glance over a shoulder, seeing more information about a user's interactions than might be otherwise desirable. For example, some ties that are visible on screen might be personally compromising; knowing how late a person tends to answer email (as shown in the time display) might give an undesirable notion of their schedule and availability; and knowing ones' own rank (as shown in the "top ten" list) might be distressing. While these issues are real, they are more concerns about the *general concept* of visualizing an interaction space than about this particular instance. This risk, known as *social privacy*, is discussed in more detail in the section 6.2.

3.3.2. Design Trade-offs

Of course, messages that are not stored cannot be recorded. While many email programs store all outgoing mail by default, Whittaker and Sidner (1996) identify a number of different strategies for handling messages. People who do not archive their entire mail histories might find Soylent, in its current form, unacceptable as a tool. In a production version of the system, Soylent would track and store messages as they were composed or received, and thus would work independently of archives.

The Soylent mail reader contains a small set of heuristics meant to repair bad email messages, and a mechanism to map together aliases (see *Technical Appendix, section 2)*. Because mailing list membership can change – while the purpose of the list often does not – Soylent treats mailing list addresses as individual entities, and does not attempt to expand out list membership.

Each of these visualizations is associated with a series of design decisions; parts that were designed but not implemented; parts that were considered by rejected. Several of these rejected decisions that are worth discussing are the use of looking at *incoming* association network information; distinguishing the *to* field from the *cc* field in email; storing and viewing *threading* information; and examining *keyword and textual* information.

The association net visualization has been tested on incoming as well as outgoing mail. Within incoming mail, there are three persons in relationship: a message from and sent by A, to B, C, and the user displayed as three undirected ties between A to B, and C. Other implementations are possible: for example, the co-recipients might not be connected together. While this would make a sparser graph, it would also lose much of the group structure now visible As mentioned above, incoming mail is complicated by a great deal of noise. In particular, networks based on incoming mail often featured spurious connections based on mass email lists, and seemed to disproportionately weight large discussion groups, which would carbon-copy many people at once. So, too, the time display for incoming mail reflected a sort of odd "communal" schedule, rather than showing any individual's activity. While these systems might be built to make the heuristics more accurate, the strengths of the outgoing-only network were sufficient to demonstrate the validity and importance of the traces.

boyd (2002) uses the three different recipient lists of a message ("to", "cc", and "bcc") as a way of identifying intimacy: a correspondent who been bcc'd now shares a secret, of sorts, with the sender; it is a closer relationship than the open "to" and "cc" lists. Initial research suggested that this was not particularly helpful, as email practices are too varied. While the first message in a conversation thread often was careful about distinguishing the roles of different receivers, later messages often did not track this information carefully. For example, different mail programs treat "reply" differently and inconsistently for the "to" and "cc" recipients. The "bcc" field is used for a variety of tasks: while boyd privileges it as intimate, others use it as an informal awareness tool, letting a third-party know of progress on an issue that concerns them.

Threading has been a common theme for a variety of email based systems (Venolia, 2003; Kerr, 2003). While this is an important organization principle, Soylent does not currently reflect thread information. A thread-aware system might attempt to use threading information as additional social data. The collective group of messages across a single thread might be weighted differently in the association net view, too: a recurrent group might be under-counted compared to numbers of individual messages. Other visualizations, following Kerr or Venolia, might try to give more information about the changing participation within a thread.

3.4. Conclusion

The Soylent system consists of three components: a mechanism to read email into a database; a structured relational database to store messages; and visualizations that use the information in the database to show interactions between users.

The Soylent email-reading component collects messages from an email store, and saves them to a relational database, storing time stamps and authorship information. This database is then used as a back-end for the visualization systems. While the Soylent implementation supports a variety of different visualizations, the implementation discussed here concentrates on a social network-like graph and on several temporal views. The temporal views show a user's changing interaction patterns: an actogram which shows times of interaction; a top-ten list which shows comparative involvement with different people. Together, these components give a multi-faceted view of a person's email interactions and network; they present a perspective on how the user interacts with their correspondents. Privacy issues are largely resolved by presenting only a view of information that the user already has: the system is not collecting others' information, but rather aggregating the users' own data.

This system is the basis for the user tests which will be discussed in the next chapter.

Chapter 4. Patterns in Communication

The previous chapter described Soylent, a system for collecting information about communication, and described particulars of how Soylent visualizes information. In this chapter, I discuss the process of the Soylent user engagement, and characterize its results as patterns. These patterns are recurrent, recognizable sequences and clusters of interactions.

4.1. The User Engagement

Chapter 1 presented four research questions. The first of them asked, *are there recurrent social interaction patterns in electronic communication and activities?* The second asked, *if there are, how can those be extracted and analyzed? Is it possible to derive recognizable and salient social patterns from these electronic traces?* Soylent was designed as a tool to address these questions, not as an end-user tool. The user engagement was structured to determine whether Soylent could be used to find structures and roles in electronically-gathered information. The questions would be answered in the affirmative if the traces show aspects of the social structure that are *recurrent and meaningful* to the user

The user engagement was centered around an interactive "visualization walkthrough" session. The concept is related to the "inbox walkthrough" technique described by Tyler and Tang (2003), in which a user's mailbox is explored by discussing individual messages in reverse chronological order. In the visualization walkthrough, the investigator and the user discussed the collective groupings of messages and relationships that they observed. Users discussed and identified the clusters they saw on screen; the researcher gathered information on the regularities observed.

4.1.1. User Population

In the winter of 2002-2003, Soylent was distributed at the field site for a user engagement, during an internship at IBM Research in Cambridge. During the internship, the Soylent system was modified to use their mail system, Lotus Notes, and was adapted to save to IBM-based databases.

Fifteen participants were recruited at the site using mailing lists and personal recommendations. Their job roles varied: the participants included managers, software developers, designers, and administrative assistants. All had extensive experience with email, using it as a communication tool continuously during their work for at least five years. The respondents provided access to their mail histories, dating back at least one year (and, in one case, five). A typical mail history for these participants would contain something over three thousand outgoing email messages; the participant with the five year archive had stored approximately twenty thousand outgoing messages.

Since this initial deployment of Soylent, the technology has been modified and extended; a second, briefer engagement occurred on an academic research site. Respondents at the academic site have tested the Soylent visualizations, and have also provided feedback on the TellMeAbout application (Chapter 5). The major intent of this follow-up was to confirm the patterns gathered from the first site and to direct future development work.

4.1.2. Observing Respondents

The first step in the interviews was to install the software on the participants' computers and initiate the mail-reading process. This process ran for approximately one hour (often more) as a batch file copied mail headers from the mail server into a database. The participants were left with a workbook that both explained the basic functions of Soylent and invited further exploration; the process was allowed to run unsupervised.

After a few days to allow the participant to explore the system on their own, the interviewer met with the participant to walk through the visualizations that Soylent displayed based on their email. During the discussion, the participant was invited to explain the network structures they saw. The discussion took the form of an semi-structured interview, centered around exploring the visualizations and describing the social texture of the interactions. In addition to those aspects that the participant spontaneously raised, the interviewer would also guide the discussion toward aspects that seemed to be of particular interest, either as an interesting network structure or as an unusual network configuration. The methodology is similar to that in Viegas (2004), in presenting visualizations of email records to users, and inviting their comments on the patterns and clusters that they observe.

Participants were encouraged to examine the association net at many different tie strengths. They were directed to examine both loose ties, pairs of contacts that were infrequently brought together in email, and tight ones, those that were frequently sent shared messages.

4.1.3. Looking at Soylent

Participants were easily able to recognize groupings, and identify them as meaningful to themselves. "*This is reflecting stuff that I felt are important*," said one participant, a researcher. They would pick out clusters and identify the activities that created them, and would connect ties to events and relationships. They were eager to tell the stories associated with the visualizations on screen. They were also quick to reject spurious clusters, such as those generated by accidental name collisions. In one instance, a respondent quickly noted when two different people with similar names were mistakenly displayed as the same person. He was quickly able to separate the distinct clusters around these people into two different groups.

On the other hand, there were some difficulties with the visualizations. Some participants had trouble understanding the layout of nodes relative to each other; their confusion at the arbitrary (and non-repeating) location of vertices in the social network view made it frequently confusing for them.

Participants were not always easily able to identify the history of a particular connection. In a number of cases, they wanted to go back to their email archives and chase down the original interaction: a number of the interviews included a few occasions of digging through the email archive, and finding the person and date that originated a given connection. While this happened several times, it was never a dominant aspect of the interview, and the number of unrecognized connections was far smaller than the number of connections that the users *did* intuitively recognize.

It was immediately clear during the testing that there was a great deal of data to handle. Users were frequently startled by the size and inter-connectedness of their network diagrams. While a network diagram does provide a way of organizing the information, it is also plausibly confusing. The participants developed a number of different techniques to understand their networks. For example, some started their explorations at a very high minimum tie strength to extract the cores of networks, then would slowly decrease that level to add connections.

After the engagement, it became clear that pattern diagrams were best understood at distance one. That is, the diagrams were all understood to describe the relationships between closely-connected persons. Participants were more interested in finding the relationships around a single person than they were in interpreting indirect and distant links. It is (as pointed out in section 3.2.4) difficult to interpret indirect connections in the Soylent egocentric visualization; therefore, the patterns largely center around a single person and a group of people who are a single link away.

4.1.4. Redesigning Visualizations

The engagement acted as a preliminary evaluation of the display technology. Participants generally found that the display required substantial study to understand and use. As the most interesting and relevant parts of the display turned out to be views of a *single* user and their immediate neighbors, later versions of the graph visualization (described in more detail in section 5.4, page 142) were modified to show only the immediate area around a single user.

4.2. Understanding Large Datasets With Patterns

It was in attempting to understand and break down this data that the notion of "patterns" became relevant. Since Alexander (1977), the notion of the "pattern language" has been a popular one in a number of fields, including computer science. Alexander uses the idea of a "pattern" as a way to collect observations, problems, and solutions in one place. Erickson (2000) suggests adapting the notion of pattern to encapsulate the parts of a field study that might be reusable or adaptable for design. Erickson's patterns include emergent patterns as well as intentionally designed steps. Martin et al. (2001) pursue a notion of "finding patterns in the fieldwork." For them, the idea of the pattern is a way of organizing and reusing fieldwork data. Their patterns concentrate on contextual placement, trying to factor out common elements of the variety of sites they study. They collect individual stories into patterns, looking for detailed commonalities. It is from Martin's example that this work draws its approach to patterns.

The balance of this chapter presents the patterns that were found, and discusses their meaning in terms of understanding users' social groups. The specific patterns described in this following section may not be universally applicable; different work patterns and organizational structures may display different characteristic patterns. However, as these particular patterns emerged in data drawn from two study sites at differing sorts of organizations, and so they constitute a useful basis for experimentation and design. After the first round of the user engagement was complete, the notion of patterns was extended from the field notes; the second round of tests helped verify and clarify this notion

4.3. Using Formal Social Network Analysis

Social Network Analysis has some tools for analyzing groups; throughout this study, we have considered ways to use formal network analysis to extract meaningful groups from user data. Network analysis, however, is difficult to apply for two reasons.

First, the formal analysis techniques assume certain attributes to social networks; the networks that we collect do not match those assumptions. The networks that Soylent displays show two parties of a three-party interaction: the ego is removed from the cluster. As such, the analytic properties of this sort of network, egocentric without an ego, are ambiguous. Because these networks do not show information transfer, for example, the notion of "betweenness" – a common way of dividing groups (Girvan, 2002; Tyler, 2003) is not applicable.

Second, social network analysis has a continued interest in finding a *single* assignment for an individual, or a *specific* group. Popular techniques include searching for cliques, k-plexes and k-clans (Wasserman, 2004). Freeman (2004), for example, in a recent literature review, lists dozens of methods to find specific groups in the classic "southern women" dataset. As our results show, however, many participants have several contacts at different tie strengths; these contacts play a variety of roles.

One recent method (Moody and White, 2003) finds several different groupings for each person; however, the method insists on breaking the network into a tree of groups; groups can only contain subgroups. This, too, is not applicable to this dataset, as the "butterfly" and "onion" patterns will show.

4.4. Interpreting Patterns

While these recurrent structures were repeatedly visible during the interviews, they were not then characterized as particular patterns. Rather, the notion of pattern is a way to explain and clarify these recurrent themes in the data. As a result, these patterns would occur in concert with others, and could be found at different tie-strength levels. This is not to suggest that the patterns at different levels were spurious. Rather, the different patterns at different levels mean different things. At high strength, patterns reflect repeated and significant connections; at low strength, patterns highlight a general topology of interactions.

For example, "June's" immediate work group looks like an Onion pattern when looking at all her mail, but shows a Nexus pattern when looking at the most frequent recurrences. While June had sometimes mailed her whole team (forming the onion's outer layer), she also worked specifically with different sets of people in the group, rotating between tasks with a designer, the software team, and so on (forming the nexus' center).

Nor did the patterns occur in isolation. Several different patterns could occur over the same set of people. This might be problematic, were we attempting to uncover social structures for the purpose of rigorous analysis. It suits, though, the use that we are making of these structures, which is to give user-meaningful accounts of patterns of interaction. Indeed, the fact that specific individuals occur in multiple groups and multiple contexts allows us to generate richer accounts of their relationships to each other.

Nonetheless, the occurrence of these patterns throughout the study suggests that they have important properties for analysis. They are *recurrent*, in a number of ways. For a single user, they arise regularly through time and with different groups. In organizations, they appear with different individuals. Although the precise details of each instance of the patterns change, the general structures operate across multiple samples.

They serve to *render work meaningful*; that is, the structures provide a framework for interpreting and understanding what is going on. People can make sense of particular events or particular objects by relating them to a larger whole through the structures that tie them together.

These properties suggest that these structures might be a useful basis for interactive technologies that can help situate individual activities within a social setting. Studies of social networks and temporal patterns show not only that these structures exist, but also that people actively employ them in the course of their work. This further suggests, then, that they can provide an effective foundation for designing novel awareness technologies. Accordingly, the second challenge for our investigation is to harness this information as part of a tool for presenting information about everyday collaboration.

4.5. Pattern Catalog

This section of the chapter presents a "pattern catalog": a listing of the patterns that were identified by collating the structures observed in the users' observations. This catalog lists social patterns, temporal patterns and combinations of the two.

The first set of patterns are those that are developed in social networks. These patterns draw their chief meaning from interpreting the structural interconnections between groups. They are visualized with excerpts from the network diagrams generated by Soylent and TellMeAbout (Chapter 5).

While some of the social network patterns refer to groups of people as a whole, others refer to an individual's connections to the other people around him. In those cases, we refer to two named persons: *Ego* is the name of the user whose mail records are being visualized, and *Focus* is the name of the person whose relationship is being visualized. Those images portray the messages sent from *Ego* to both *Focus* and others.

It is important to note that even though these patterns may bear close resemblances to common social network diagram terminology, they are interpreted very differently. Indeed, one reason for using this new terminology is to help make it clear that these patterns must be understood differently. The visual appearance of the standard social network "star," for example, closely resembles the "Nexus": but while the "star" refers to a single person at the center of a set of disconnected others, the "Nexus" refers to a *Focus* being interconnected with *Ego* to a set of disconnected others.

Edges in these structural images are not interpreted as information-sharing edges, but rather "co-knowledge" or "co-involvement".

This section also identifies distinctly temporal patterns: changes over long and short periods of time, referring to both individuals and groups. Temporal rhythms are also a way of identifying and separating *ordinary* time from *exceptional* time. An event might be understood as digressing from a usual temporal or network pattern ("these people infrequently work together at this time") and thus be particularly worthy of note.

4.5.1. About the Catalog

Each page of the catalog mentions six aspects of the network pattern: its **name,** a **type** (social or temporal), a brief **description** of the pattern, one or more **examples** of how the pattern can be found, a note on the **frequency** of the pattern, and a note on **interpreting** the pattern.

Six different types of visualizations are used to illustrate the pattern catalog. The association net diagrams generated by "Soylent" and introduced in Chapter 3 are used extensively. In these diagrams, an edge between two persons represent one or more messages sent by *Ego* to both those persons. Edges that represent more messages are said to be at a higher "tie strength."

Network diagrams generated by "TellMeAbout" (section 5.4, page 142) are also used heavily. TellMeAbout (or TMA) diagrams are very similar to Soylent's "association net" diagrams, however, they are limited to "distance 1". "Distance 1" diagrams are centered on a single correspondent, *Focus.* People in the diagram who are not connected to *Focus* are hidden, so that the diagram shows *Focus*, the people who *Ego* connected to *Focus*, and the interconnections between those people. Some of the TMA diagrams also collapse "structurally equivalent" names together and show them as a vertical row; those names are all interconnected (see figure 4-5 for examples of all of these).

TMA diagrams can also have their edges and vertices colored to indicate temporal extent. In the case of these illustrations, connections and individuals are colored by the most recent message exchanged. The most recent connections, through February of 2004, are dark gray, while the oldest, starting in March 2001, are white.

The monthly "top ten" list (section 3.2.6, page 84) is also used to show differences in relative positions. Most of those diagrams are annotated with a dark color to aid in tracking *Focus*.

TellMeAbout generates histograms of messages; these displays show the number of messages and the number of attachments per week exchanged with focus. The diagrams have four rows; those four rows represent:

- messages sent *from Ego to Focus* (that is, *outgoing* mail)
- messages received *from Focus by Ego* (that is, *incoming* mail)
- message attachments sent *from Ego to Focus* (that is, *outgoing* attachments)
- message attachments received *from Focus* by Ego (that is, *incoming* attachments).

An annotated histogram can be found at Figure 4-9.

Last, actograms (section 3.1.4, page 73) are used to both show regular variation over a week, and long-term changes in interaction patterns.

Images reproduced in this chapter have been anonymized for publication, and were derived from both the initial field site and at the academic site.

Table 4-1.Pattern Type Summary

Name	Social extent	Temporal aspects?	Visualization
Disconnected	Network	-	Association Net
Onion	Network	-	Association Net
Clique	Network	-	Association Net
Butterfly	Network	yes	TMA & Net
Nexus	Network	yes	TMA & Net
Gradual Shift	Individual	yes	Top-ten & histogram
Bursty	Individual	yes	Top-ten & histogram
Normalcy	Individual	yes	Actogram
Usual action	Individual	yes	Actogram

Pattern: Disconnected Clusters

Type: Network, non-temporal

Description: Clusters of names that (visually) do not connect with each other at all—entirely disconnected components, without messages or people connecting them. (Underlying, *Ego* ties to all of them, but there are no interconnections between the groups.)

Example: For the participants who used their work email account for external non-work communication, there would often be no overlap between their social ties and their work ties. One participant showed the different

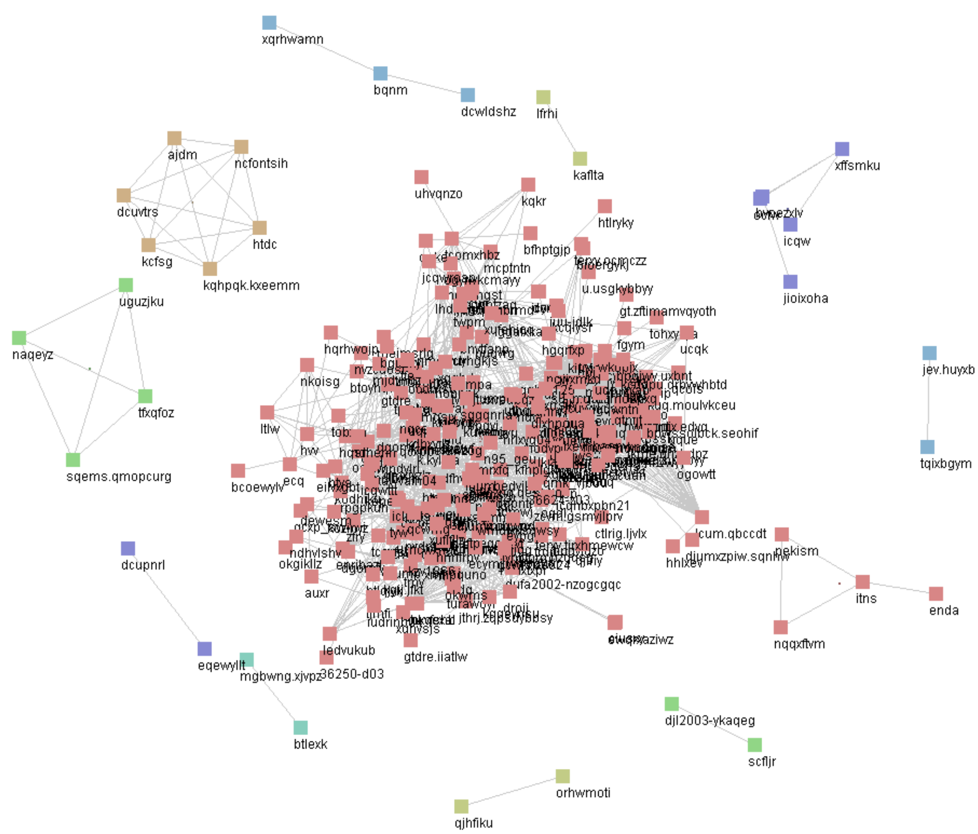

Figure 4-1. Disconnected Clusters

disconnected clusters that his mail fell into: a carpool mailing list, a school-related group, a religious group, a group of friends, a work group. Another participant's mail was separated by research projects in different disciplines: his studies in visualization were disconnected from his studies in interaction design. The display in Figure 4-1 illustrates a variety of contexts as disconnected groups.

Frequency: Virtually all respondents demonstrated some sort of disconnection: none of our respondents had completely-interconnected mail. Because our respondents looked at their mail at a variety of tie strengths, they would see clusters separate from each other at the higher strengths. The one participant who did not see is was an administrative assistant, who ran Soylent program on a mail account that was closely tied to her manager. As most of her work was in response to his requests, almost all of the mail in that account was carbon-copied to the manager.

In the figure, a screen-shot for one user shows a large cluster at the center, and more weakly-tied disconnected clusters surrounding it.

Interpretation: Members of disconnected clusters are likely to be in different contexts, to have different needs. This organizational principle helps understand how contexts can be separated. It is important to note that members of a single cluster may not be all in the same context (see Butterfly and Nexus); similarly, separated clusters may merge over time.

It is interesting to note that the Disconnected Clusters are equivalent to the traditional social network “star” pattern: a single person (*Ego*) is tied to a set of others, who are not interconnected.

Pattern: Onion

Type: Network, non-temporal

Description: The onion pattern represents a core of tightly-connected people, surrounded by a periphery of loosely-connected ones. Peripheral people are attached to fewer core members, less frequently. Many team projects that our interviewees were involved in were onion shaped. Interviewees would often recognize the core as a central group, and the periphery as occasional contributors, part-time members, or people less directly involved in the group. As the minimum tie strength criterion increases, peripheral people—those with fewer connections—fall away, leaving the central core. The overall effect, then, is of an onion gradually being unpeeled; contacts fall away from the outside, showing other patterns within.

Example: The onion is illustrated here with a pair of groups (Figures 4-2 and 4-3) from a workshop. The inner group, a subcommittee at the workshop, continued working together after the workshop ended and so exchanged many messages. In the second image, the periphery has begun to fade away, leaving the inner group. The sequence of illustrations was generated by raising the threshold at which ties would be displayed; the inner clique is more highly inter-connected than the outer group.

Frequency: Every user observed several onions in their mail system; indeed, it was the successive peeling-away of layers that allowed participants to understand and unpack the context of the mail system. The visualization

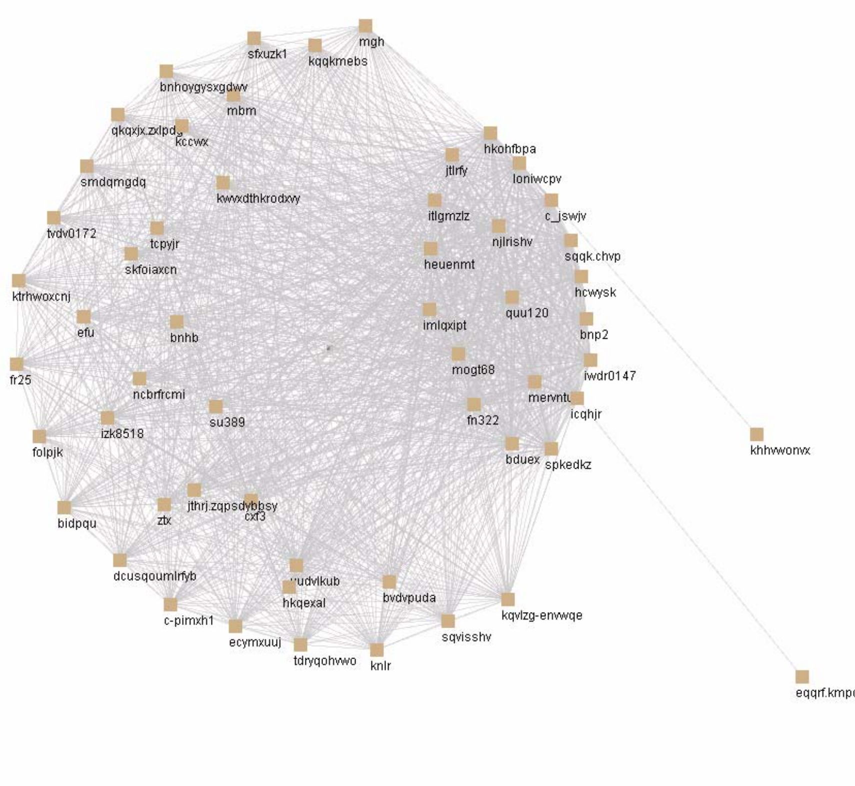

Figure 4-2. The Onion pattern.

suggests that there was no sharp distinction between core and periphery; rather, the core revealed itself within an unfolding series of layers.

Interpretation: The Onion shows successive layers of involvement. In particular, members of an onion help specify and understand *subgroups.* In particular, messages to outer members may be relevant for the context of inner members. Conversely, inner members may be understood as participating in multiple groups: the inner-most group, and all the layers around it.

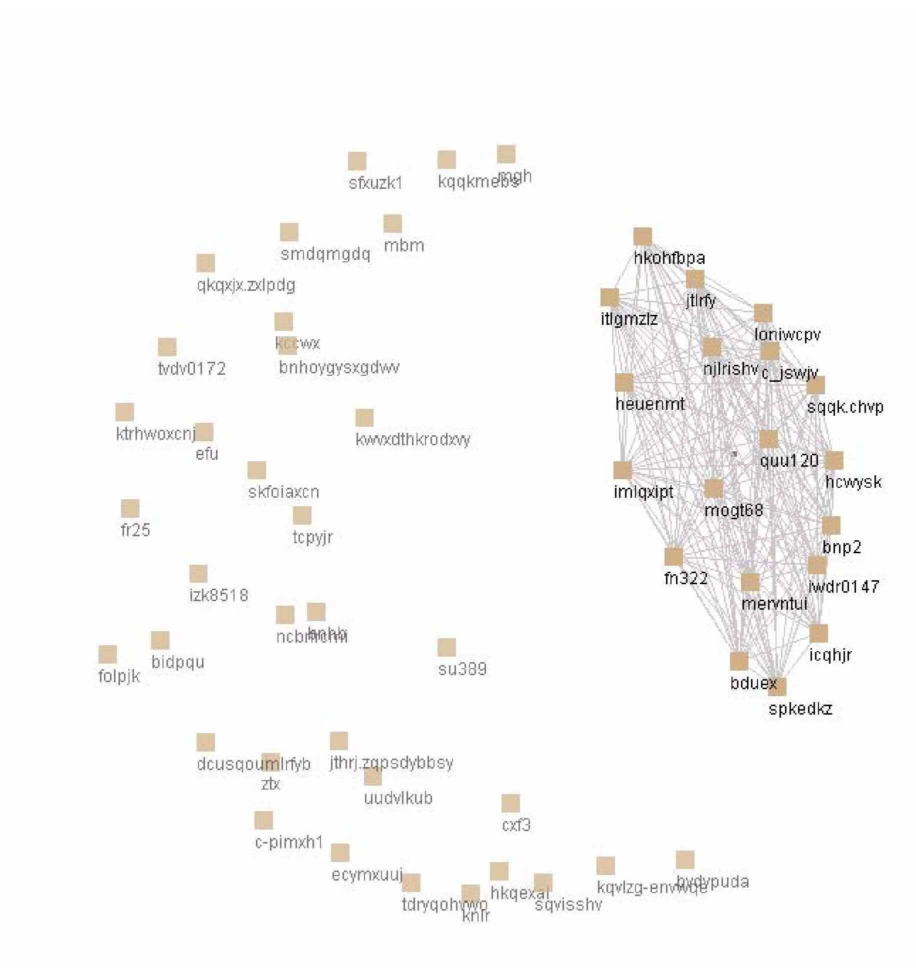

Figure 4-3. The core subgroup within the Onion.
This image is the same as figure 4-2, but with a higher tie strength. At right, note the inner clique.

Pattern: Clique

Type: Network, non-temporal

Description: A fully-connected component of a graph. Ties between group members carry equal, or near-equal, weight and are fully interconnected. Note that an Onion may have a clique at its center; more interestingly (and as in the "onion" example), a clique may itself be an outer layer.

Example: In a clique, ties between group members carried equal, or near-equal weight and were all interconnected. This usually came from messages being carbon-copied to an entire list of people, so that all pairs of them were connected. The onion example, above (figure 4-2), shows two nested cliques: the inner clique is strongly tied together; the outer clique, too, is loosely tied. In the "disconnected clusters" example (figure 4-1), several of the outer clusters are also cliques.

Frequency: A mailing list or carbon-copy list would generate a clique; as such, all users showed cliques of one sort of another. Cliques are frequently visible as the core of onions.

Interpretation: A clique is a structurally uniform cluster as a carbon-copy list; the members of the clique are interconnected and, as such, can be treated equally among this cluster.

Pattern: Butterfly

Type: Network, possibly temporal.

Description: Named for its two large wings surrounding a small *Focus*, the butterfly pattern finds a single person bridging two different contexts. The two "wings" of the butterfly may be simultaneous, or may be temporally separated. (That is, they may occur over different time periods, or have different durations.)

In the butterfly pattern, the *Focus* and the *Ego* are, in some sense, duals. The *Focus* and the *Ego* are the only two bridges between the two groups. Therefore, just as the participant sees the *Focus* as bridging the two groups, so too would the *Focus* see the participant as bridging the same two groups.

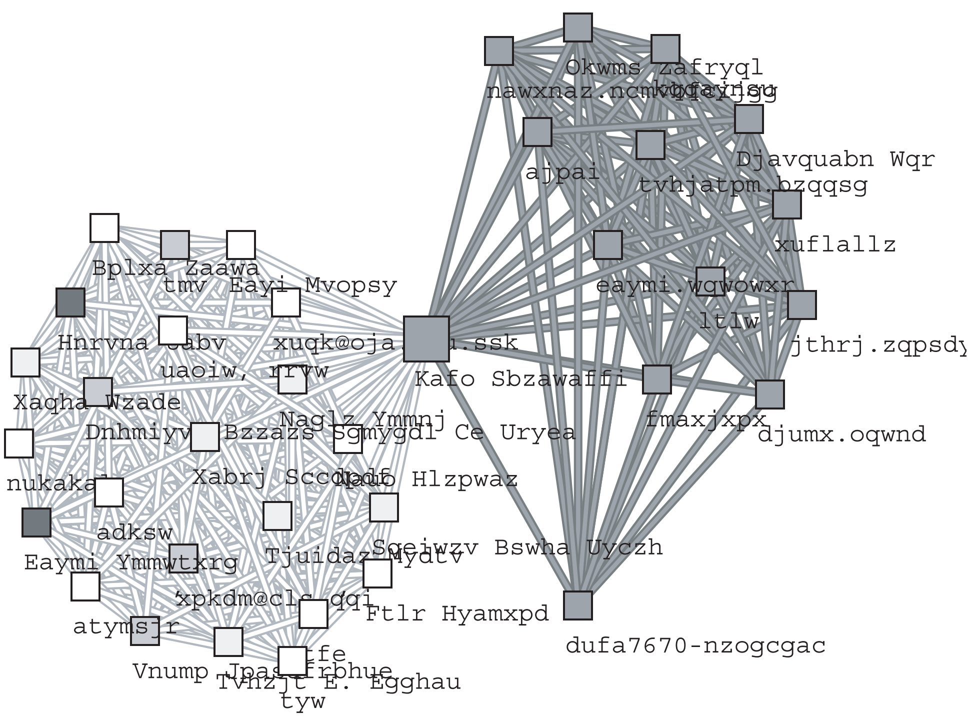

Figure 4-4. The Butterfly pattern.

Example: An example of a butterfly can be found in Figure 4-4. This image comes from the mail of a researcher, and focuses on one of his collaborators, "Kafo." The researcher is a member of both a research group (at right) and a social group (at left). "Kafo", however, is the only person to bridge both groups.

Labeled by their most recent interaction, we see a stark contrast: the two wings are highly separated in time. The interactions on the right wing are all fairly; all the edges have seen activity recently, and so are dark. Members of the left wing, the social group, have more diversity. While the social group as a whole has dissolved (hence the white edges), some members (such as "Eaymi") have had more recent interactions.

While some instances of the butterfly were professional (such as two overlapping work teams), others were the overlap of two very different contexts (such as a religious group and a rideshare group).

Frequency: The butterfly was the least-common pattern. Approximately half of the respondents had at least one situation with a single user bridging several groups.

Interpretation: In a butterfly, the focus is involved in two different populations. Communication with one population may not—indeed, is unlikely to—imply the involvement of the other population. They might be reasonably treated as largely disconnected. (For example, messages to the focus might be profitably separated into two clusters, one for each wing.)

Pattern: Nexus

Type: Network, possibly temporal

Description: A *focus* is connected individually to a series of other individuals and small groups.

Example: An employee, reporting back to his manager on a series of projects, would have the manager tied into each of these different contexts. So too might a pair of teammates working on a series of projects, as might a group administrator.

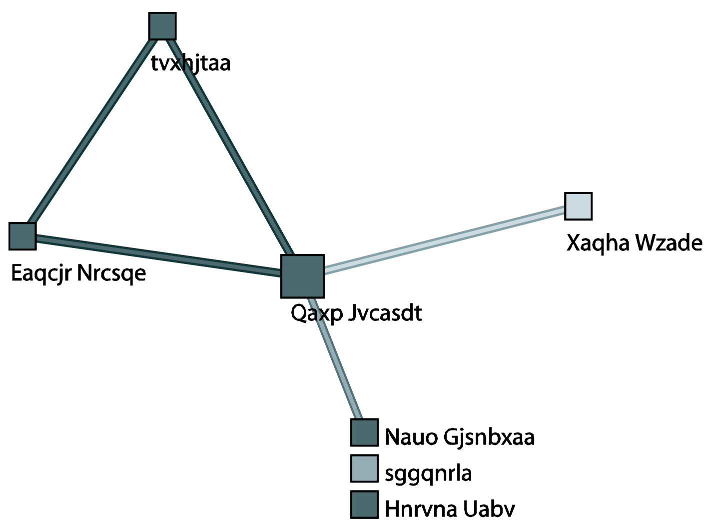

Figure 4-5. The Nexus pattern.
Note the variable times, dark (recent) on the left, and light (old) on the right. Note also the committee on the bottom right: these three people are all "structurally equivalent", meaning they are connected to each other and to everyone else in all the same ways.

Figure 4-5 shows one such nexus. The respondent had a variety of different interactions with the central *Focus*, "Qaxp". The right (oldest) connection shows a connection during Qaxp's initial visit to a group;

“Xaqha” coordinated that visit. The bottom connection shows a job application process; the application committee is the list of names at the bottom right. Most recently, the two individuals at top-left show a continuing involvement with Qaxp after he was hired as a member of the group.

Figure 4-6 is another example of a nexus. “Yitw” worked with the user on an extended book-publishing project. During that time, they brought in a series of other people: a publication team, co-authors, and similar. “Eaqam,” at left, was an invited co-author who decided not to join the project. The three interconnected names at bottom-right are staff associated with the publishing house; they were involved up until publication time. Last, the three separate names on top were co-authors; they each had

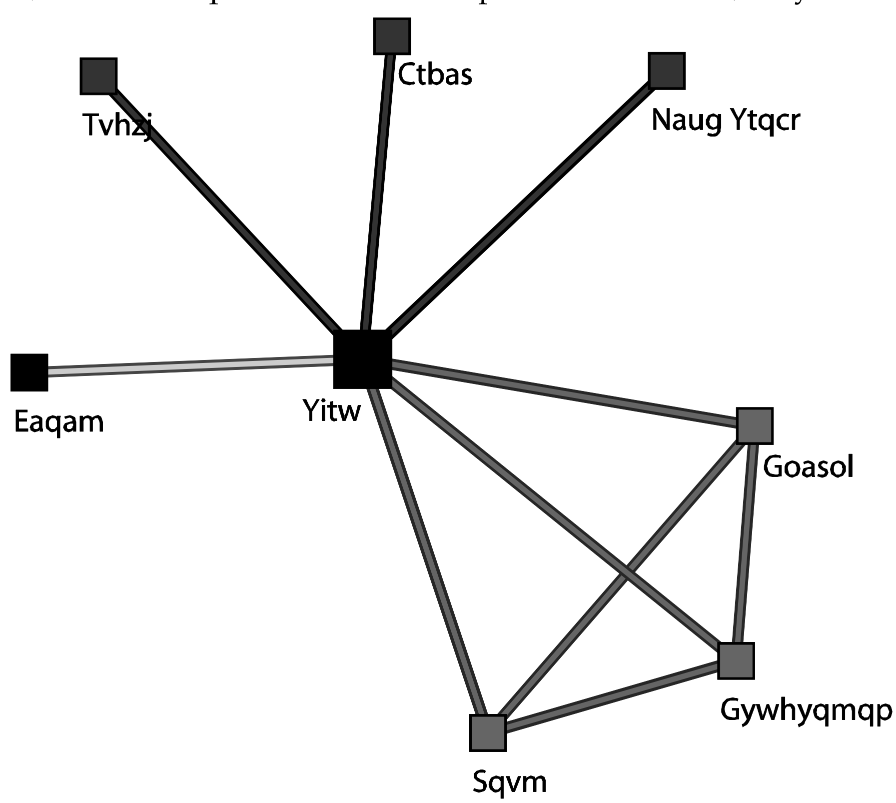

Figure 4-6. A second, different Nexus.

continued contact after the book was finished as they worked to promote and distribute it (and its associated authors' copies).

It is interesting to note that *Ego's* relationship with Eaqam continued on, even after she stopped being involved in the project: the dark color of her icon suggests that they had continued interaction, while the light color of the tie suggests that Eaqam and Yitw fell (from the user's perspective) out of touch. This shows that *temporal contrast* can be an important way of setting a contact off from their setting: the times when they were most involved with *Ego* may be very different from the times when their surrounding neighbors may have been involved.

Frequency: The nexus pattern was universal; between managers, secretaries, and collaborators, every respondent had several people in their visualization who functioned together as a nexus.

Interpretation: There is a different degree of information associated with the people at the periphery of the nexus, compared with the *Focus*. The people at the edges *disambiguate* the role of the *Focus*: that is, their presence in a message makes it clear which of the contexts that message is associated with.

Pattern: Gradual Shifting Positions

Type: Individual, temporal

Description: A single person changes substantially in their relative importance over a period of time. These may include rising or falling slowly, or sudden prominence or demotion in frequency of communication.

Examples: The top-ten list made it clear that, in addition to recurrent sets of correspondents, most participants had a series of changing team members: people would become prominent for a period of time, then slip away. These changes happened with notable events: the starts or ends of collaborations; joining a new team; the arrival and departure of summer interns. The examples here show a slow decrease and increase due to travel and an increase due to a new project.

Figure 4-7 represents mail from one respondent, a researcher. As the year ended (and a number of bureaucratic obligations came due), his contacts with his team members dropped noticeably in exchange for administrative contacts. One team member's name is bolded in order to clarify the pattern. In January, as his commitments eased, the team member's email moved back upwards on the list: that is, became more

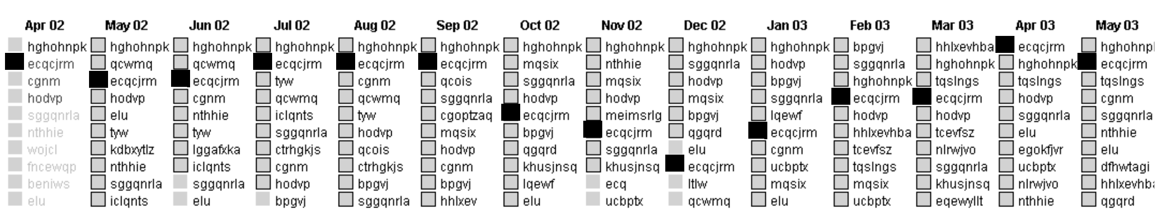

Figure 4-7. Top ten list showing variations from travel.

frequent. The effect was aggravated by the member's physical absence: further away, contacts decreased.

Figure 4-8 shows a new project coming into the foreground. "Kw" (highlighted in green) is a mailing list of a project which the respondent began working on in April of 2003. Messages to the list immediately began to dominate the respondent's other interactions. The top line of this top ten view represents approximately fifty emails sent per month. (The drop in September shows that the project lost some significance compared to a more-urgent paper deadline.)

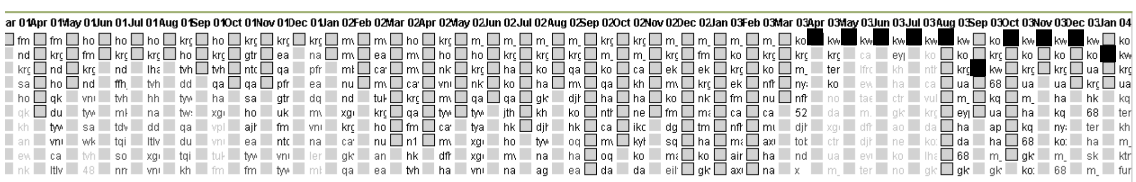

Figure 4-8. Top ten list showing a new project.
The project appears at first position once it is formed.

Figure 4-9 shows the traces of work with a co-author on an extended writing project; the peaks occur around deadlines for drafts and reviews. The later, smaller bumps come from progress in a slow printing process and the final release.

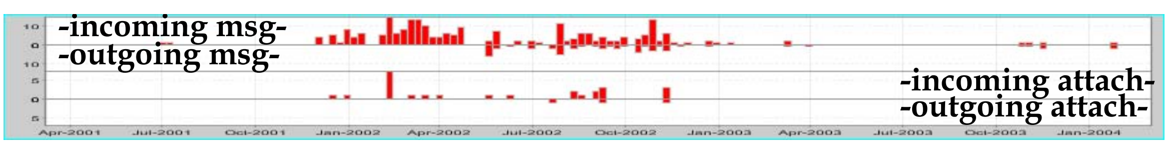

Figure 4-9. Timeline showing interaction on a publication project.
This timeline (ranging from April 2001 to January 2004) shows four rows of histograms: outgoing and incoming messages, and outgoing and incoming attachments.

Figure 4-10 shows the interaction of a romantic relationship: it begins rapidly and rises in the number of messages exchanged; it fades away after some time

.

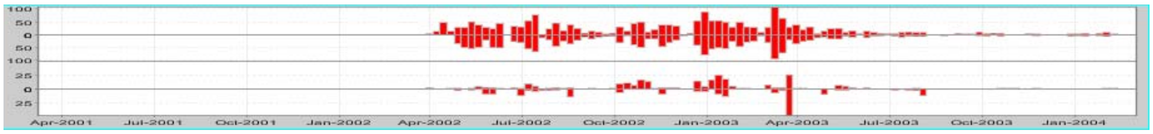

Figure 4-10. Timeline showing interaction in a romantic relationship. Notice the abrupt beginning and diminishing tail.

Frequency: Gradual changes in project teams were a recurrent theme among interviewees. All interviewees identified at least one project that had changed over time.

Interpretation: Gradual changes may be understood as a way of watching how prominence, responsiveness, and roles change; these themes have been important to other researchers (Begole, 2003; Lockerd, 2002), but have been modeled as unchanging constants. As these diagrams suggest, these figures are not constants; instead, the way that people interact with each other changes over time. A view of gradual changes provides broader information to applications that try to provide or use information about responsivenes.

Pattern: Bursts of Activity

Type: Individual, temporal.

Description: Some changes are best tracked by individual, short-term events rather than long-term changes.

Examples: The first chart, figure 4-11, shows a timeline of interactions between a student and a professor. Five separate interactions are visible in those top rows; they are driven by a combination of external social calendars – such as the American school year – and internal deadlines. The first set of messages is a teaching-assistant relationship; the large spike in late May of 2001 represents a final exams period. When summer began, the teaching period ended and a pair of research projects began. The two projects look

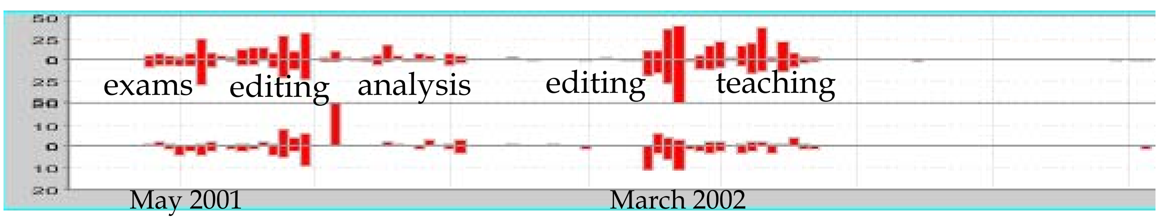

Figure 4-11. Timeline showing interaction with a professor.

rather different from each other: the first involved editing a paper with a July deadline; the second involved a slower data analysis process. The next peak, in March, is a deadline for a prominent conference; the final cluster of interactions occurred around a second quarter of teaching.

The latter chart, figure 4-12, shows a conference paper submission. The conference in question had a brief response period; the April submission was accepted in July, which triggered a round of revisions. The authors then expanded the paper into a full journal article in November; another comments and revisions period concluded in February.

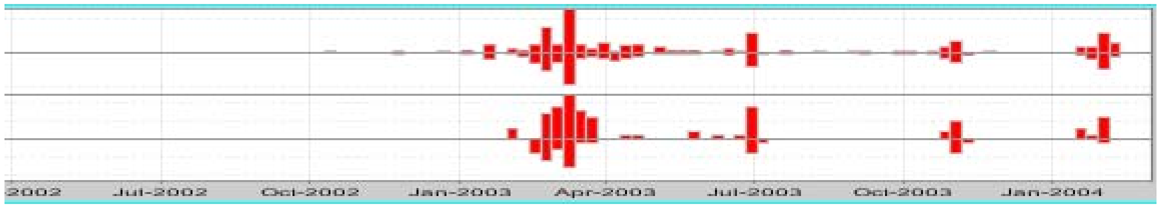

Figure 4-12. Timeline showing conference paper submission.

Pattern: Changing Roles within a Network

Type: Temporal, network

Description: The *focus* of a group changes position within a network over time. This is visible by seeing how the ties around the focus change. (The time-indexed Nexus and Butterfly above, was an example of this.)

Examples: In image 4-13, the different contexts separate themselves out clearly. From interviews, we learn what these different groups represent: the two earliest groups are a social group to which the respondent belonged. Both *Ego* and *Focus* were active members in the group, but at the end of one year, several members left. Those now-departed members are drawn in white, at top. The next cluster at bottom, in a light gray, is another portion

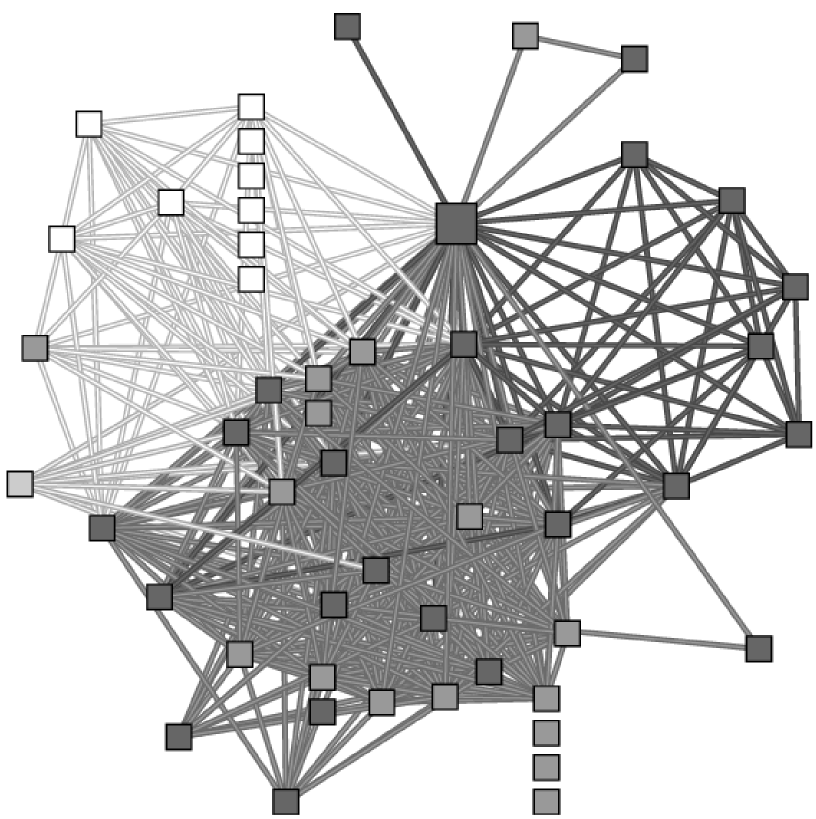

Figure 4-13. A group changing over time

of that group; they, too, graduated and left. The last cluster, at right, shows active members of Focus' social group.

The second image, 4-14, shows the way that a different *Focus* changed between groups. The lower-right (recent) cluster is a research group that *Focus* joined, while the upper-left (older) cluster is a social group that *Focus* had been involved in, but left. *Ego* was involved in both, and thus watched *Focus*' change.

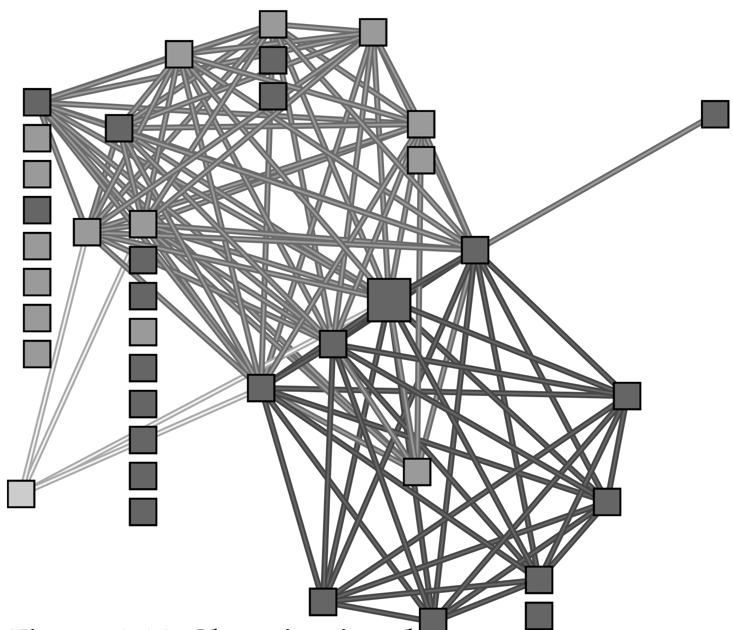

Figure 4-14. Changing involvement.
Focus leaves a social group (top), but stays involved with a research group (bottom).

Interpretation: This pattern helps understand how contexts and interactions change. While some of the network patterns were based only on network information, this pattern helps point out that contexts must be understood as located in time, and that they change.

Pattern: Normalcy

Type: Individual, temporal

Description: While individuals may act differently, each of them settles into patterns of routine activity. Weekly rhythms caused distinct patterns in mail sending. Because much of the working and social world are driven on a seven-day cycle (Zerubavel, 1985), weekly patterns were easily visible features.

Examples: At the field site, the computers were configured by default to run their anti-virus routines on Mondays at noon; the routine was slow, and would render the machine nearly useless for an hour. The coordinated lunch break that most respondents took during that hour was clearly visible in the temporal views. (A similar pattern would be observable in the wok habits of Roy's (1959) factory workers, with their strictly regulated hourly breaks). In another example, an Orthodox Jew proudly noted that Soylent showed he sent email frequently throughout the week—except on Friday nights and Saturday daytimes, his day of rest.

Figure 4-15 shows the weekly interaction between the a graduate student and a professor. With a regularly-scheduled Thursday afternoon meeting, the frequency of communication would gradually increase over the week, as deadlines for teamwork became more prominent. Thursday afternoons—the times after the meeting—would then be a time to exchange additional information and to follow up ideas from the meeting. By early the next week, it would be time to focus on other issues before the next meeting.

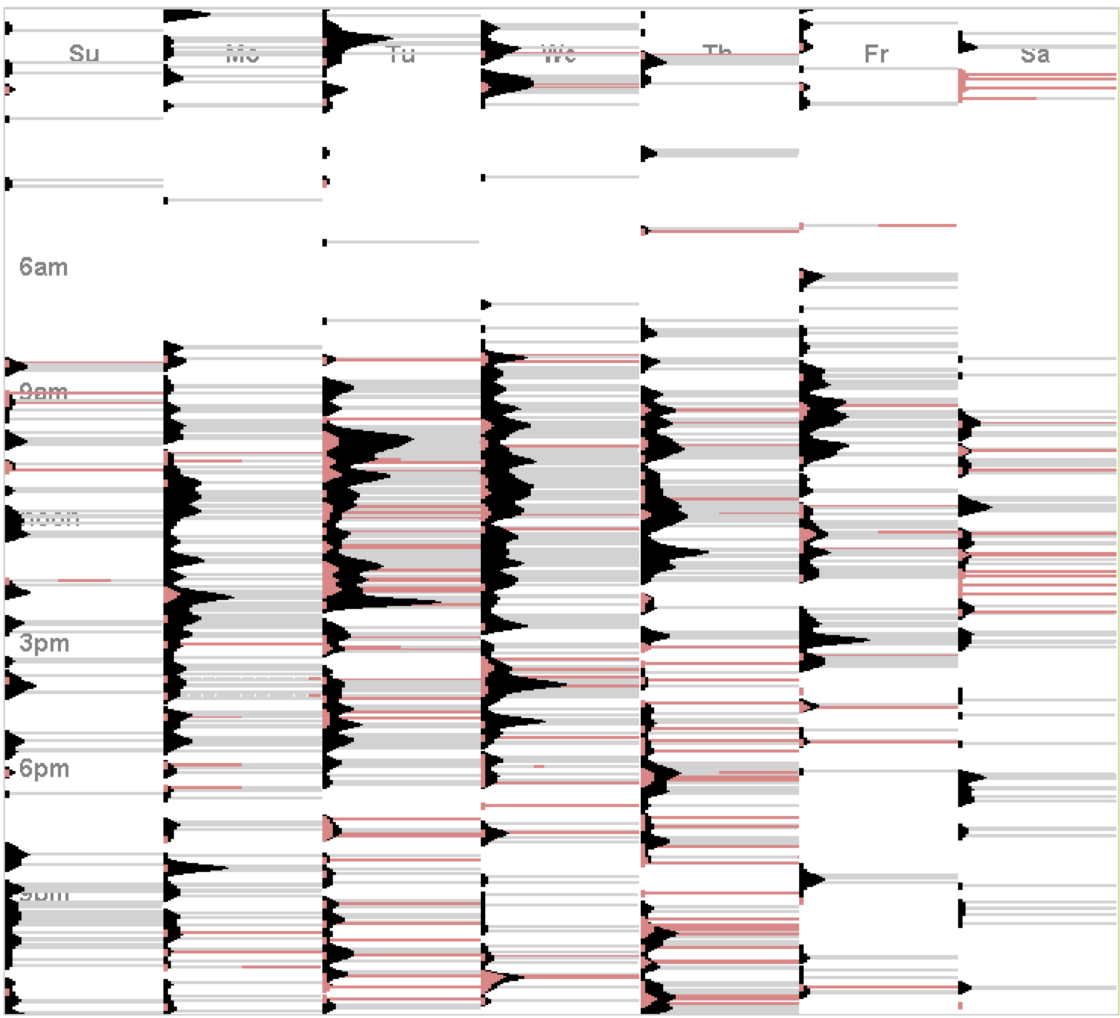

Figure 4-15. Normalcy
This actogram shows communication between a student and advisor. (Reproduction of figure 3-4, page 74).

Frequency: It is possibly to interpret all users as having some sort of normal schedule; however, they radically differ in that regularity. Therefore, all users have this pattern, and most or all find it useful in describing their day to day routine.

Interpretation: Applying various sorts of regularities to predictive systems has been covered in Begole et al. (2003), who suggest algorithms for extracting regularities to predict future availability. These patterns, which can look at both regularities of interaction with others as well as regularities of availability, can be used to predict a variety of future interactions.

Pattern: Unusual Events

Type: Temporal, individual

Description: This pattern describes events that fall outside of expected and normal interaction schedules. The more regular a schedule is, the more visible and clear exceptions to it are. Unusual might be defined both relative to a normal activity schedule, and to a normal interaction pattern with a specific person.

Example: A program office administrator at the field site worked a fairly regular schedule: as Figure 4-16 shows, her workweek started and ended at specific times each day. The few messages that fell outside those normal times, then, took on great significance. During her interviews, she was easily able to identify the digressing occasions: the trip that corresponded

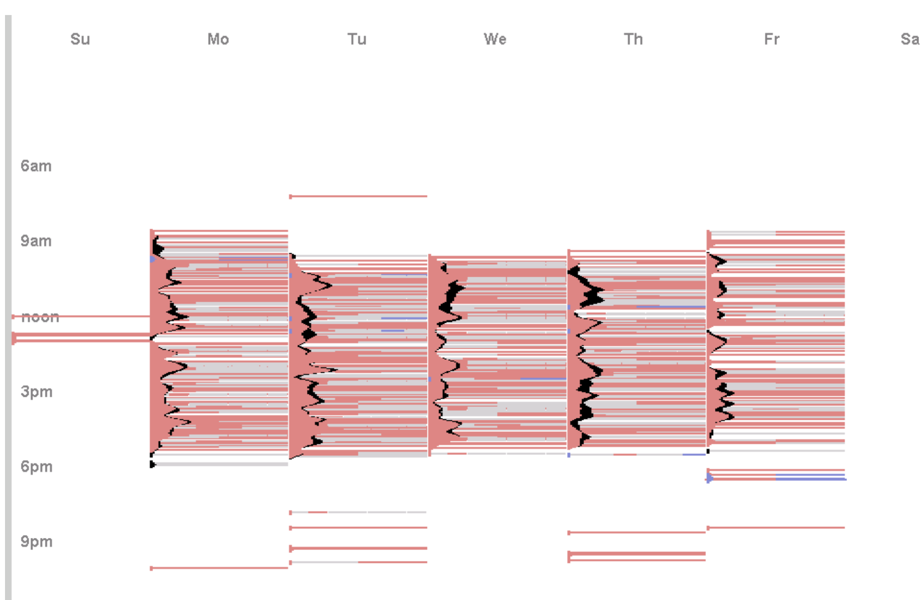

Figure 4-16. Exceptions to normal schedules.

to one message, the early emergency meeting that corresponded to a different late cluster.

Frequency: Finding exceptions to the patterns was not an important aspect during the field studies; in retrospect, few users had specifically unusual temporal events that were easily visible. None the less, as regularities become more visible, so too does the importance of locating differences.

Interpretation: While "unusual" may be hard to specifically operationalize, unusual events are likely to provide insights into important things, as they may be prominent to the user's experience. Exceptions, as the flip side of normalcy, define unusual events: a co-worker missing a usual schedule may indicate that their usual responsiveness may be diminished.

4.6. The Efficacy of Patterns

Unlike manual systems such as ContactMap (Whittaker et al., 2002), Soylent uncovers patterns automatically from activity records such as electronic mail logs or stores. Some CSCW researchers have been skeptical about the efficacy of automatically mining social networks through this sort of analysis. Indeed, such skepticism is clearly warranted; email records, for example, provide only a partial, incomplete view of anyone's activity, and no automatic techniques can assess the relevance or meaningfulness of any connection. However, the goal of building these applications is not to derive formal social network analyses, with their attendant mathematical formulations of social structure and interaction. To some extent, this mitigates the problems: users are capable of recognizing the patterns that are important to them.

A related observation is that the meaning and the value of the patterns that this section discusses is highly variable. Similar patterns mean different things to different people, and the same pattern might hold different implications over time. Again, this argues against abstract interpretation of the information, such as its use to create a categorization of working relationships. Instead it suggests that it is important to present concrete results which are amenable to examination and interpretation. If the meaningfulness of the structures and patterns that we have observed lies in their interpretation by the people whose work activities they describe, then this process of interpretation should remain a user activity.

A final observation is important to note here. Although this work has identified a number of characteristic patterns, it is clearly not the case that every person falls into such a pattern, or that every interaction with a person follows the pattern of previous interactions. However, patterns can be used to explain aspects of specific communicative events whether or not those patterns are followed in each specific instance. A pattern may be salient to an interaction because the interaction follows the pattern; but by the same token, a pattern may be salient to an interaction precisely because the interaction deviates from it. Unusual situations are often even more interesting than usual ones, but their unusual nature is, of course, defined with respect to the same patterns that characterize normal interaction.

4.7. Discussion

The views that we present here are simply visual depictions of an email mailbox, showing incoming and outgoing messages; they are, in some sense, entirely personal information. However, they are also the records of interactions; they place individual actions within a broader context that relates to them to other actions and other people. The complexity of the patterns – bursts of activity, rising and falling patterns of involvement, periodicity and intensity – are not modeled but presented for reflection and interpretation.

In these patterns, a single person may be caught up in multiple structures concurrently. Soylent provides different ways to explore a data space, depending on the views selected and the ways in which parameters are adjusted. There is no single account to be offered of a specific individual, a specific time period, or a particular interaction; there are multiple accounts reflecting different aspects of the relationships between users and work. This allows users to explore their social spaces and to discover what is meaningful to them.

The patterns presented are not solely meaningful to the users who encounter them, but that are meaningful *in terms of accounts of working patterns and activities*. The patterns are not just patterns of electronic message exchange; they are patterns of organizational relationships, styles of work and interaction, events and settings, roles and responsibilities. In other words, the patterns provide the means to give an organizational account of individual and collective behavior. This is a critical property for patterns and structures that we wish to use in order not just to analyze working habits but to support them, and to situate specific working activities within broader patterns of collaboration. If the patterns that we were able to find were purely patterns of information exchange, then this would be much harder to achieve; however, in our user engagement, subjects volunteered consistent explanations of organizational context, which supports the earlier research questions.

This work is not the first, of course, to combine a temporal perspective into the design and consideration of social networks. Historically, however, networks have been presented as timeless. A typical network survey, for example, will ask "who do you associate with" to establish ties. Phrased timelessly, the question has no internal notion of when the association might have happened, or might yet happen. Network surveys infrequently attempted to establish temporal bounds or scopes.

Recently, however, a growing literature on dynamic networks has begun to emerge (Carley, 2003), which does attempt to rectify these issues. The analytic perspective of dynamic networks models the network as a series of fixed snap-shots, captured in time. Each of these networks is then contrasted the network at different times from a structural perspective, examining the changes in various network statistics. For example, they might follow the centrality of a given person at different times. In those studies, there is in fact a serious attempt to collect information at different times, and to compare those times to each other. Within this growing discipline, surveys *do* attempt to compare relations to each other at different times.

This perspective is noticeably different than the TellMeAbout visualization. TellMeAbout is trying to show a broad overview of a series of events, and to display the development of this network. TellMeAbout makes an unusual contribution in presenting a static, readable visualization of (some of the) changes in a network over time. Rather than comparing whole graphs to each other, the TellMeAbout visualizations of this section attach temporal information to the edges of the graph. This allows a fine-grained perspective on the changes that occur over time: it allows a visualization of *how* a relationship has changed over time, rather than a mathematical statement of how *much* it has changed.

4.8. Conclusion

In this chapter, I discussed the Soylent user engagement. Users at two field sites, an industry research and an academic office, ran Soylent on their own mail, and were interviewed about what visual arrangements they observed in their email. The major finding, of *patterns*, showed that there were recurrences in both the *social network* and several *temporal* views. Social patterns such as the "Onion," the "Nexus," and the Butterfly showed a variety of relationships between people; temporal patterns such as "Changing Roles" and "Normalcy" displayed the ways that these relationships can change over time.

The next chapter discusses the use of these patterns for both *awareness* tools, and explores the *application* of the patterns.

Chapter 5. Applying Social Information

The user engagement that we carried out through the Soylent prototype supports the original design intent: structural and temporal analysis of interactions can reveal aspects of how individual work is embedded within broader social contexts. In addition, it provided us with a set of recurrent, meaningful patterns that occur across a variety of working settings, and provide a starting point for more structured design explorations. In order to explain the implications of these patterns, this chapter will discuss two specific applications: a generalized awareness tool and an extension of an existing communications technology. In the following chapter, a richer vision of a broader system that uses Soylent as an underlying concept will be developed by comparing Soylent to ContactMap.

This chapter outlines an approach to developing applications based on the Soylent infrastructure. It identifies important characteristics of the patterns discussed in the previous chapter, and uses those as a guide toward developing new types of software. This guide is used first to describe a novel activity overview interface, known as the "workspace social proxy," which is used to describe some of the desirable features of a communication system. It then is used to apply social information to both awareness applications and to ways of extending existing applications.

5.1. Design implications from Patterns

What does it take to build a system based on these patterns? The primary role for the social and temporal patterns in awareness systems is to tie specific people or moments of time to broader activities. That is, when engaged in activity connected to one individual, social patterns can be used to place that person in a context of collaborators and peers, while temporal patterns can provide greater specificity by placing current activities within a temporal context. These patterns give information about how the social context around a person is textured in both their network and across time.

Of course, while a software system might be able to note connections and links between specific activities and broader patterns, the meaningfulness of those relationships is a matter purely for users to determine. The systems that this project proposes do not take any action on behalf of a user, but rather *suggest* to the user how activities are related to recurrent structures of contact and collaboration.

Simply reprinting the full Soylent social network is not a satisfactory solution. Awareness tools, by their very nature, sit adjunct to other tasks, and so must be easily visible without detracting from the primary task. The user tests of Soylent showed that the network view was complex, and required substantial examination and manipulation to fully understand. It is clearly inappropriate for a lightweight awareness tool or for a system that acts as supplementary information in a larger workflow. Indeed, the idea of a screen-dominating network view changes this from an annotation to a task in itself, perhaps one of reflection (Thimbleby, 1990).

Interpreting the network through patterns provides a way to reduce this complexity. In the interviews, the participants used the network as a way to evoke stories and events, and used patterns as a way to simplify these. So too, an awareness tool can use results from the patterns to summarize and simplify user interactions. My goal, then, is to examine the patterns for key insights. An analysis based around the patterns helps find important events and group dynamics; we collect the information we gain from that analysis into a simplified interface that shows minimal information but allows exploration. This refined network information is then exposed through an API, which allows various awareness applications to be developed.

The prototypes are built to exploit automatically-derived information discovered using the Soylent tool. These patterns can be used in different ways. Sometimes, it makes sense to present the patterns themselves as part of the user experience, by indicating the type of pattern that connects individuals with respect to a particular working task. In other cases, the patterns can be used to select and filter information: for example, a system might be trained to be able to distinguish a manager from a coworker according to the patterns in which they typically occur, and filter accordingly.

Such a system would be able to place artifacts such as email and files within their social and temporal context, and would be able to expose that information to the user.

5.2. Aspects of Patterns

First, let us abstract away from the pattern catalog of the previous chapter to extract three important, general characteristics from the patterns.

- *Structures may bridge between different activities*. Single tasks often spread out between multiple applications. By uncovering relationships between individuals and artifacts, collaboration structures may be able to show how the separate activities located in different applications are, in fact, part of the same task.
- *Structures may help disambiguate different activities*. By placing individual activities in a broader context, the collaboration structures may help not only relate but also separate activities by linking them to different higher-level patterns.
- *Structures may throw current activities into relief*. Frequently, the interesting relationship between current activities and patterns of past action are that current activities do not fit the pattern of the past. People do not always follow uniform paths. Activity might be meaningful not because it fits the pattern, but because it does not.

These characteristics will be applied to the design of the tools described later in this chapter. These insights can be combined with the "interpretations" section of the pattern catalog in order to get at some of the underlying ideas within the patterns.

5.3. Prototype Sketch: a Social Proxy

A hypothetical destination would be a social awareness tool. Such a social awareness tool should be able to provide information that puts a recent work into a social context.

The first application, then, is a prototype sketch, the "workscape social proxy" (WSP) This section is meant to discuss possibilities of how future systems might act; it does not describe an existing or working system. In particular, the proxy counts on information that is not currently available to the Soylent system, including associations between files and people. While issues regarding the construction and implementation of these associations are discussed in the "Future Work" section (section 7.3.3, page 171), this conceptual sketch presupposes these ideas.

The design is partially inspired by the social proxy design that Erickson and colleagues developed as part of the Babble system (Bradner et al., 1999). Babble's social proxy is a visual indicator of activity and attention in an online conversation tool. The proxy allows conversational participants to see the activity of others at a glance, and visually draws attention to transitions such as the arrival of new participants. Babble's social proxy is, however, restricted to conveying information about patterns of activity within the Babble system itself. In combination with our techniques for analyzing interaction, it suggests a design that provides an electronic proxy not simply for the users of a single application, but for the activities—the clusters of people and projects—around which a user's activity is directed.

The WSP is a tool that conveys a visual representation of the workscape. It provides the user with an overview of *recent activity*, and provides visual representations of the connections between *people* and *artifacts*. In particular, it highlights very recent and current activity, and also shows past highly-active work events that are no longer prominent.

The WSP is designed for peripheral monitoring (Heath and Luff, 1992). It shows people, relationships, and artifacts relevant to the users' current actions. It responds dynamically to the users' activities (e.g. the set of applications being used, and the documents loaded in them), to the availability of others (e.g. through instant messaging tools), to the arrival of information (e.g. new electronic mail or IM messages), and to the passage of time (building on previously established patterns). The goal of this design is two-fold. First, it should situate particular working activities within their social context, helping to show how they are connected to other activities and to other people; and second, it should provide a resource for easily moving between activities and getting in contact with people relevant to the work of the moment.

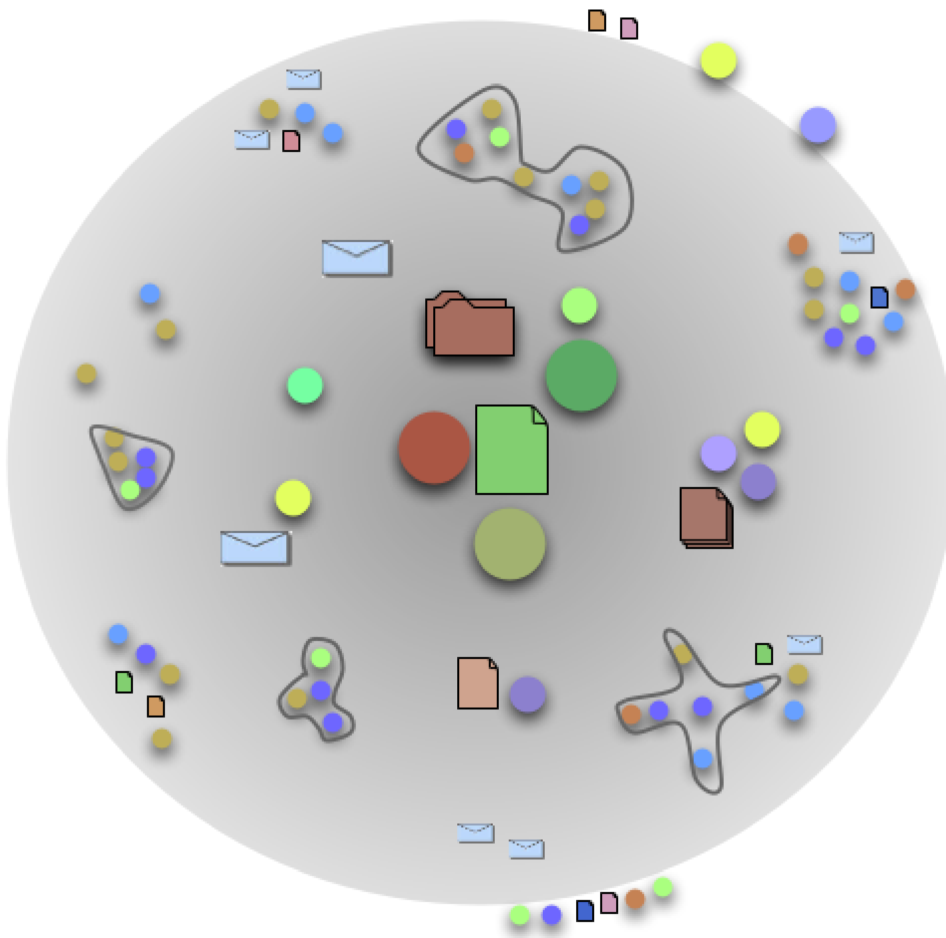

Figure 5-1. The Workspace Social Proxy.

Figure 5-1 shows an interface sketch for the WSP. Using the derived information about social and temporal patterns, it highlights the set of objects and people that are connected, at various degrees, to the documents that the user is currently working on. The use of proximity, layout, size, and color indicates groupings and degrees of connection; dynamic responsiveness reinforces these. Note that while distance outward reflects activity and currency, location around the circle is arbitrary.

The WSP takes network information and integrates it with information about activities and the artifacts associated with it. Within the display, a number of different patterns can be seen. Parts of the screen show disconnected groups, butterflies, and a nexus are visible; in addition, the WSP uses patterns to structure and filter information within the chart for presentation. The central section, for example, shows a group of people surrounding a current document and a set of files associated with the document. The smaller circular dot above the cluster suggests an outer member of an onion. Other constructs—such as the large, bow-tie shaped group at top—suggest two clusters of people sharing both an interest and a single mutual contact person.

This display is similar to that of the Scope (van Dantzich, 2002); however, Scope limits itself to a display to "incoming" events. Scope breaks current activities into four sectors ("Task", "Alert", "Inbox", and "Calendar"), which all show incoming events and activities. Objects toward the center are more urgent. In contrast, the WSP shows how clusters of people and activities are chronologically and socially interconnected: because calendar events, messages in the inbox, and activities are all displayed together, there is no need to divide into sectors.

5.4. TMA as Awareness

In order to implement tools like the WSP, we have developed an API to the underlying Soylent system (see *Technical Appendix* part 3). The first instance of this interface is an end-user tool called TellMeAbout. TellMeAbout is a simple client using the Soylent infrastructure; it describes how particular individuals are situated socially, relative to other people, to time, and to the documents they exchanged. Thus, it must be able to address questions such as "in what context do I know this person," "when did I last interact with this person," and "what were the last documents I exchanged with this person?" These initial goals take into account the major categories above: the tool, by making visible the structures about a person, might helpfully locate them within a particular activity. This network, in turn, might disambiguate the multiple activities that the person might be involved in. Last, unusual contacts—messages that come after a long period of silence, for example—would be highly visible.

TellMeAbout is a straightforward implementation that explores some of the most important aspects of the problem space. To be able to supply TellMeAbout with information, the Soylent infrastructure must provide an API that can analyze and find connected groups of people, can show the temporal duration of individuals and groups, and can understand the connections between messages, people, and data files.

It is this API that is important. While the end-user interface may be simple, it illustrates the major features of the analysis, and provides a outline over the scope of the problem space. TellMeAbout can be straightforwardly expanded into a variety of rich graphical tools that provide broad awareness and expanded interface information

---

```
> TellMeAbout -person bmarkham
79 messages since Mar 23 '01,
 most recently May 12 '02
 especially
    Mar 26 '01-May 7 '01,
```

```
    May 28 '01-Jun 11 '01
closest connections include (gayle)
Attachments (in and out) include:
   Quiz21c.doc
```

As in the sample output printed above, TellMeAbout gives temporal information about the user, including the first time the correspondent was ever emailed; the most recent time they were emailed, and particular periods during which they sent the most messages. This provides a brief temporal overview of the interaction. Since this sample code was executed in late 2003, it says that the correspondent has been out of touch for over a year; the dominant interaction dates back a year beyond that.

The display also shows other correspondents who are closely connected, in order to give a brief overview of their social location. The algorithm extracts only connections above a threshold (generally, one percent of messages), and so gives a very abbreviated view of the social context. Only one file has been exchanged by email: "Quiz21c.doc," which was sent as an email attachment.

Looking at temporal structures provides characterizations of the rhythms of contact; looking at social structures helps tie this person to others. This quickly solves the dilemma faced by Joe, the software salesman in our opening scenario (section 1.6): his client's context would be visible, the range of their past interaction highlighted, and person who introduced them would be visible.

Note that patterns are not directly used in TellMeAbout: *bmarkham's* presence in an onion is not shown, even though her relationship to *gayle* is made visible.

While TellMeAbout is an extremely simple tool, it is the basis for more elaborate tools that provide a richer experience. For example, TMA accepts input from the command line's "standard input" pipe, and displays profiles based on names that come in from that pipe. This can be easily connected to an email client, and can then display information about the people from whom the user has most recently received email. Connecting this to a Sideshow-style display (Cadiz, 2002) allows a continually-updated display of information relevant to the current situation, updated continually as new email arrives or is sent. In both "standalone" mode and "tickertape" mode, TellMeAbout stands separate from other applications, although it is part of the conventional single-user desktop.

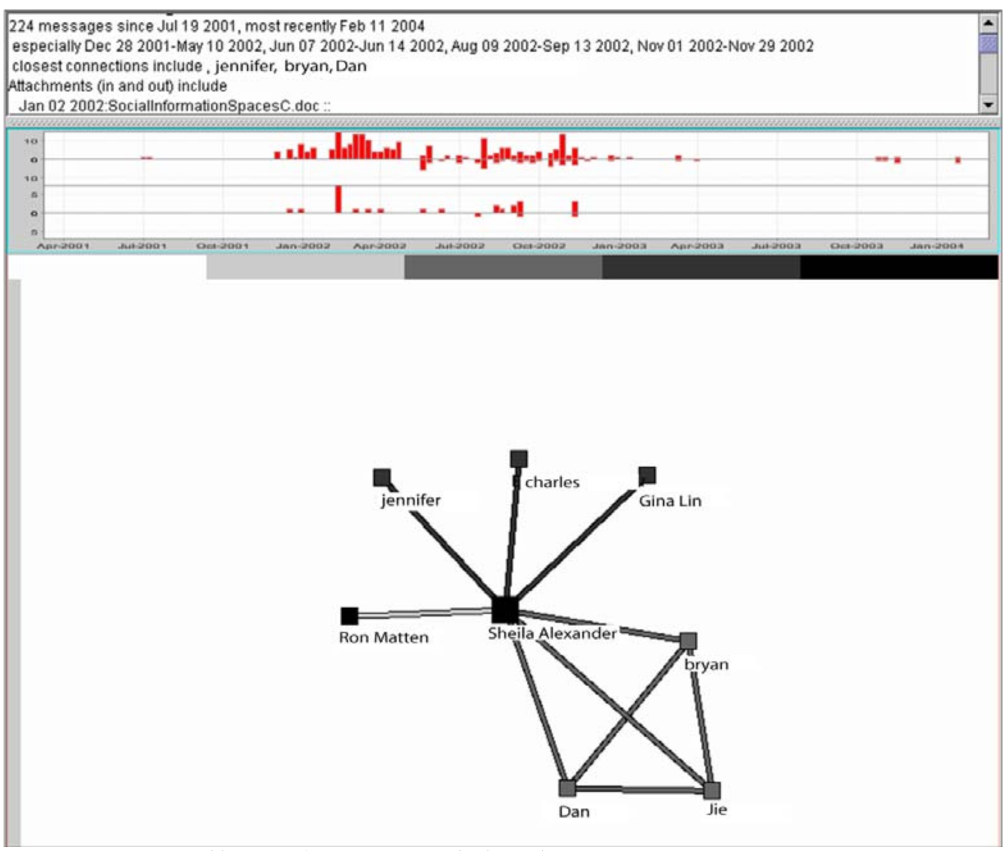

Figure 5-2. TellMeAbout visual display.

A graphic version of TellMeAbout can also provide a rich visual display, shown in Figure 5-2. This visual display provides all the information available in the command line version (in the top box), as well as two additional sections: a temporal chart of past interaction, and a visual display of the social network (which is also annotated with temporal information). Like the textual form, it can also act as ticker tape, changing display to match user input.

In Figure 5-2, for example, we see interactions with "Sheila." The top box gives the same basic statistics as discussed before. (Note that the scrollbar suggests that there are many more attachments not shown.) The display is not currently optimized for size; it is clear that it could be reduced substantially to take up less space.

The timeline section is divided into four rows: outgoing email messages from the user to Sheila, incoming messages from Sheila, outgoing attachments to Sheila, and incoming attachments from Sheila. We may read the chart as suggesting that Sheila most heavily interacted with the user over a year ago, in late 2002, although there has been sporadic contact since. The user sent many more messages to Sheila than he received, suggesting that some of his older mail was lost, or Sheila's mail came in from a different email address; Sheila and the user have not exchanged any attachments at all since early 2003.

Below the timeline section, the social network diagram is a simplified view that shows only Sheila and her immediate neighbors. These connections are shown in the same way as the classic Soylent edges (see section 3.2.4, page 81); however, the diagram is limited to the *strongest* connections around Sheila:

approximately 5% of the email messages directed to Sheila must be carbon-copied to one or more of the neighbors in order for them to be shown. In this case, Sheila acts as a nexus among a series of connections, including a cluster of three names at bottom right.

The edges are labeled by age, as described in section 4.5.1, page 104. The features to note are that both edges and nodes are labeled by their age. The uniform gray of Jie, Dan, and Bryan's messages and connections suggests that the three of them were contacted simultaneously—or at least within the same time period—and that there was no independent connection with them afterward. In contrast, the dark spot on Ron's name next to the light gray connection suggests that the user kept in touch with Ron, even after he no longer connected Ron to Sheila.

The visual TellMeAbout therefore provides a brief, visual overview of both a combined social context for a given correspondent. With TellMeAbout, it is possible to derive basic social context information about a particular person. It places a person in time and in their network.

5.5. Enhancing Email with EE4P

While TellMeAbout provides peripheral information about current activities, it is disconnected from the working context. In particular, the display provides feedback and further information about current activities, but cannot be in turn manipulated to interact with the people or information in question. For that, the tool must have more domain-specific data. The second application discussed in this section combines knowledge of work tasks with Soylent data sources to enhance an existing application with social information.

The prototype tool is known as EE4P, an abbreviation for "Enhancing Email for People." EE4P is an extension to a traditional, three-pane email client. Its code is based on ICEMail (Nourie, 2001), an open-source project. EE4P uses the Soylent databases and API to provide the user with annotated information about both incoming and outgoing email. Every message and every person is tied to a series of other messages, people, and groups; as the user reads or writes a message, EE4P provides auxiliary information about current interactions with them. Thus, for example, in writing an email to a current group, the system makes available past emails both received and sent to members of the group, and a selection of past people who were associated with it, in a sidebar.

5.5.1. Recipient prediction

One important feature of this tool is recipient prediction, which allows the system to suggest which set of people may be connected to a current message recipient. Recipient predication is triggered when the user types in a name to the "to" line of a message and presses the comma key, which suggests that more names are to come. The system then searches an immediate network for other names that have frequently co-occurred with that name, and suggests them in a pop-up menu. It is therefore easy to invoke an entire repeated carbon-copy list at

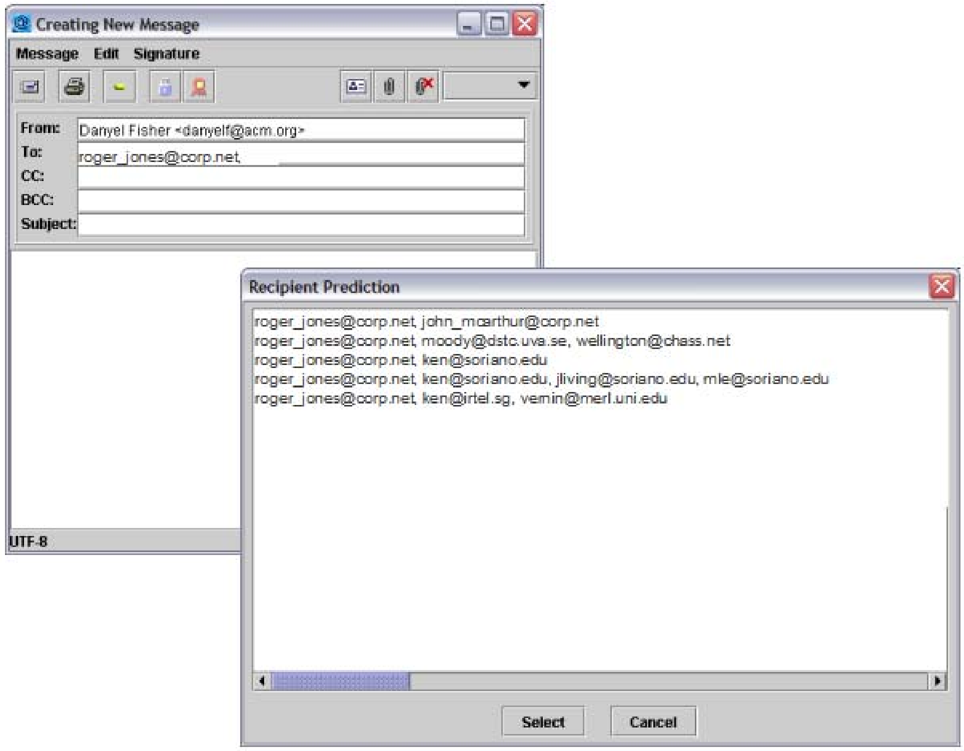

Figure 5-3. Enhanced Email for People (EE4P) Recipient Prediction

once. The system is not constrained to historical combinations; it is able to suggest groupings that are logical expansions of the previous names.

This is a place where patterns *are* directly used, but not shown. By using clustering algorithms, EE4P is able to suggest various granularities of groups. For example, when a message is sent to a team lead, it might suggest four distinct lists, based on different sections of the network:

- the core members of a team
- the core members plus the developers
- the core members plus the designers

- everyone involved in the team, including core members, developers, and designers.

Of course, these are only recommendations; the user is free to select names that the system does not suggest at all.

5.5.2. Message awareness

When reading or writing a message, the system uses the network around the name to allow easy access to other messages (based on similar audience and time) and other groups that may be relevant to the user. Each of these names, messages, and groups is selectable, and can reveal broad information about the user's history. In particular, the user has access to:

- address book entries for every person involved in the message,
- other people who may be closely linked to the people mentioned in the message,
- message histories for every person involved in the message, and other people who are closely involved,
- and a group "message history" that covers the participation of the whole group.

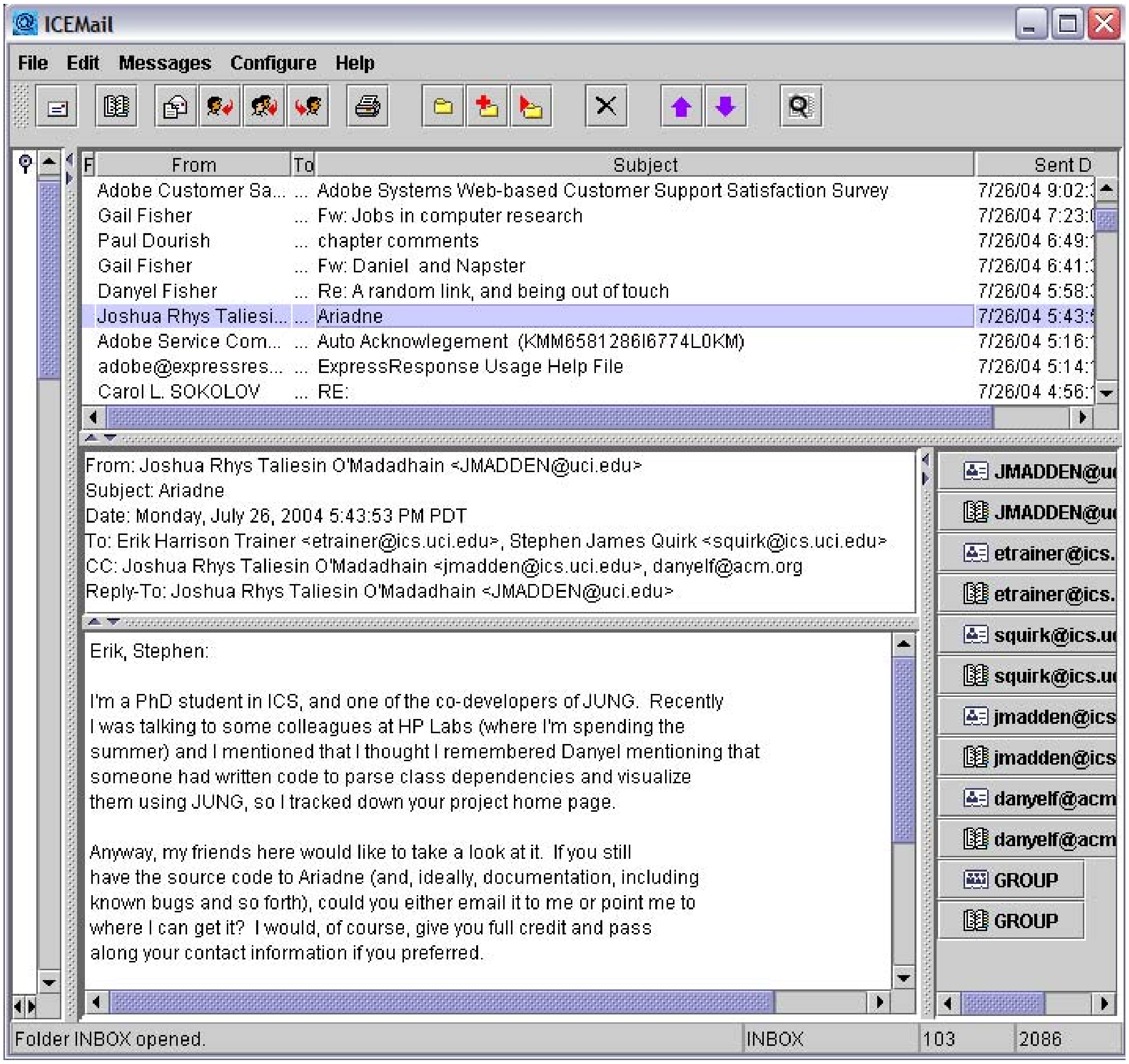

Figure 5-4. EE4P main screen.
Main display for Enhanced Email for People (EE4P), built over the ICEMail client. Note "message awareness" panel on the right.

While the implementation does not currently take into account some potentially useful data, such as links to attachments or directly-related messages, the mechanism is designed to be extensible enough to add those features.

5.5.3. Enhanced Address Book

EE4P provides a standard address-book that stores manually-entered information about individuals. Entries, however, are annotated with additional

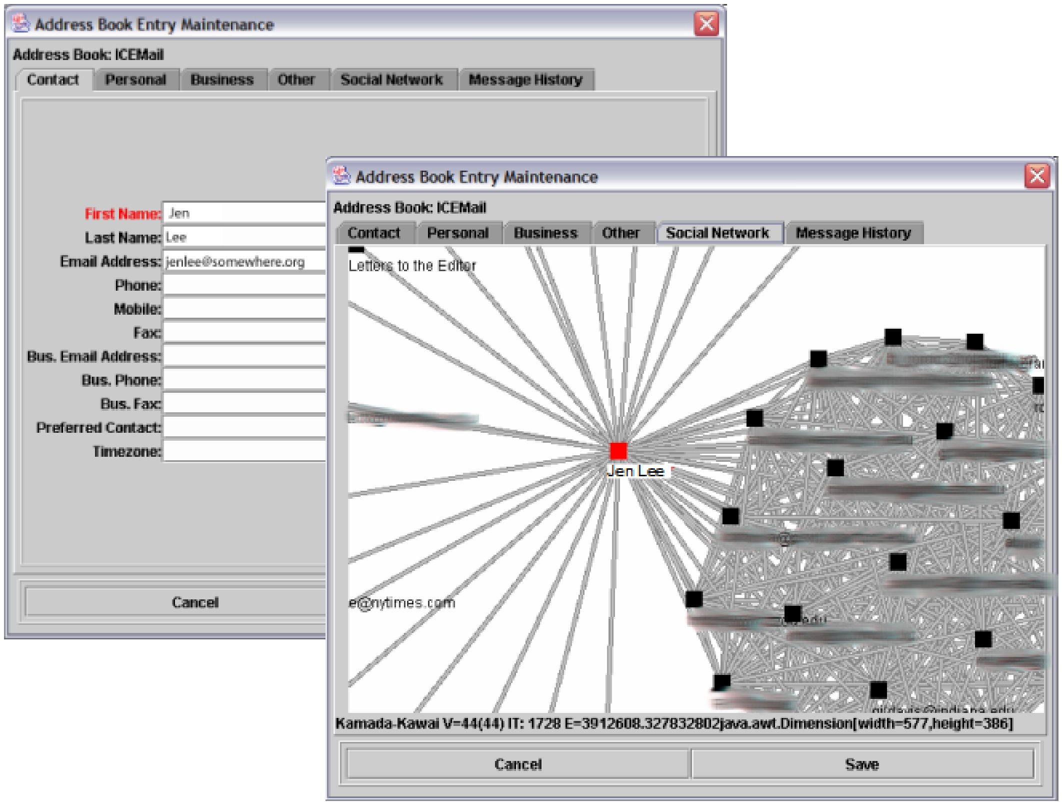

Figure 5-5. EE4P Address Book Entry Maintenance

information: one pane shows a social network view, while another pane gives a history of past messages to and from that person. EE4P can also generate on-the-fly address book entries for groups of people.

5.6. Discussion

These three systems illustrate three different possible views of the workspace. The Workspace Social Proxy could operate as a background tool, providing introspection about current activities and easy access to data, files, and people. TellMeAbout displays a way of using networks to provide information that resolves contact management problems, while EE4P situates the results of a network computation within the context of an email client.

Chapter 6. The Future of the Workspace

In previous chapters, the Soylent system has been described in isolation, as one approach to addressing the issues of everyday collaboration. Soylent suggests a more ubiquitous use of connections between people, artifacts, and times: that connections between people and resources could be available within an operating system, providing social information as a service to system components. Other systems, however, have attempted to take on some of these issues, and have cut across the domain with different dimensions. This chapter attempts to better explore the Soylent place in the world by comparing it to a related project, ContactMap (Nardi 2002b).

The comparison both strengthens the picture of how Soylent works, and provides an opportunity to look into ways the social interfaces can be organized.

6.1. ContactMap

ContactMap began with the insight that personal social networks are critical resources in today's economy (Nardi 2002a). ContactMap organizes the computer desktop according to people in the user's personal social network. It does this by displaying the contacts in the user's social network and providing functionality relevant to those contacts. A contact can be an individual or group who the user is familiar with, and wishes to make available to themselves. Each contact has an icon: a photo of the contact, or

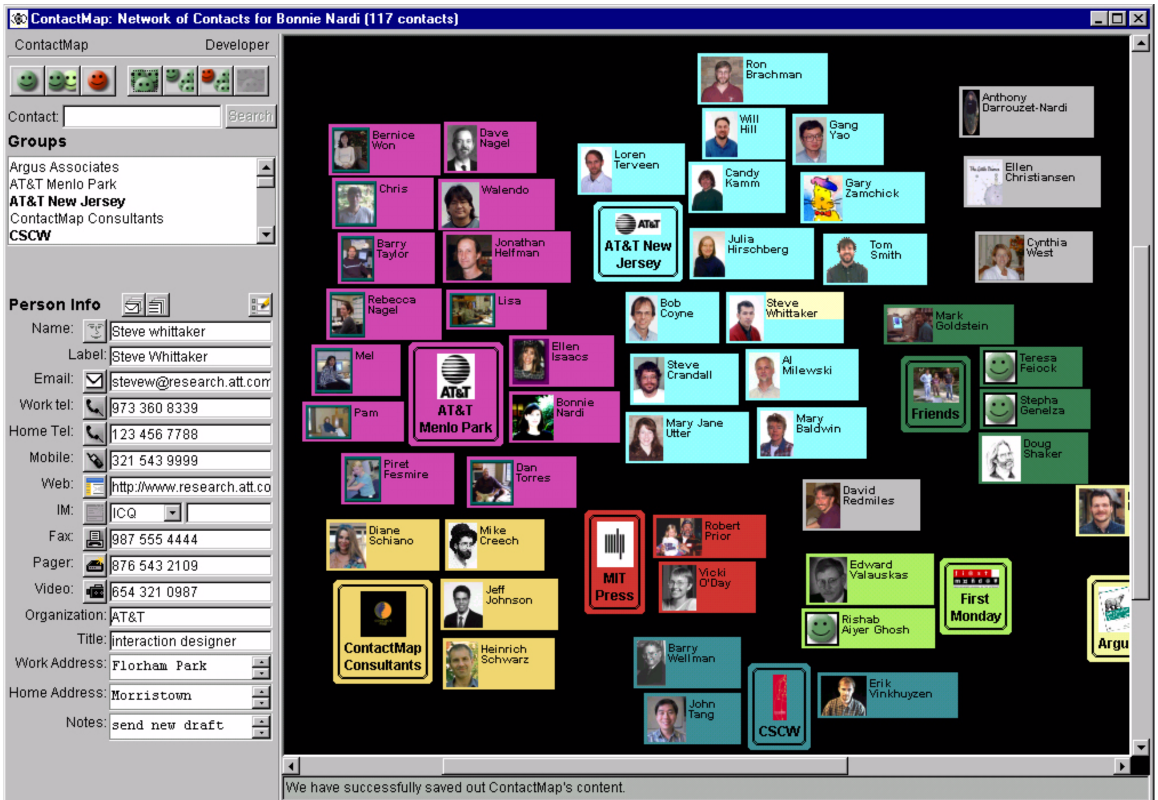

Figure 6-1. Contact Map.

another mnemonic image. Contacts may be clustered together into groups; each contact may belong to none, one, or more groups, as shown in Figure 1.

ContactMap integrates communication and information management in a single user interface. Each contact can be clicked to access information associated with the contact or to communicate with the contact. Let's say Sally is our user and Sam one of the contacts in Sally's ContactMap. In a typical scenario, Sally clicks on Sam to get a list of the email messages he has sent her. She reads the last couple messages from him, and then wants to call him. She clicks on his icon, and uses ContactMap's click-to-dial feature to make the call. After the call, Sally remembers something she forgot, and she clicks on Sam to send an email message. ContactMap opens a new message addressed to Sam. Sally's work has taken place with a minimum of fuss—no looking up phone numbers or email addresses, no launching of additional applications. Sally sees only the email from Sam and does not have to search through folders.

ContactMap helps people manage the multi-tasking with different individuals and groups characteristic of work in the modern economy (Nardi 2002a). Within the tool, Sally could have found any documents Sam had sent her, as they are indexed by contact. Sally could have started a video conference with Sam, or an IM exchange. She could have sent him a fax or gone to his website. Sam's icon also shows reminders and notifications of unread email messages associated with him.

Any ContactMap functions can be performed on a group instead of an individual. A conference call could be initiated, a group email sent, a website linked to, and so forth. Individual contacts can be placed in multiple groups in ContactMap, as individual in a social network often occupy more than one role in a user's life. Sam might be Sally's coworker, and also a member of her gardening club.

ContactMap is not hierarchically organized. Empirical research showed that most users had small maps, with an average of 95 people (Whittaker, 2004) While maps will certainly grow over time, it was found that people had a small set of active contacts. Elaborate means of organizing contacts hierarchically are not needed (and would be confusing for many users). As active contacts come in and out of a user's life, contacts can be shrunk down to small icons, and placed off-screen when not needed. The information about contacts is preserved, but does not need to be visible, cluttering the display.

ContactMap is a social workscape in which the most common actions of a user's daily worked are reorganized to reflect the people with whom the user interacts. It does not replace operating system functionality but provides a different user interface to that functionality. Instead of privileging files and folders, ContactMap privileges people in the user's personal social network.

Setting up ContactMap begins with a numerical analysis of the user's email folders. Based on domain names, frequency of contact, and frequency of replies to messages, ContactMap presents a list of contacts to the user (Nardi 2002b). Users then select the contacts to include in their map, and group them as they wish. An individual contact can appear in multiple groups. Groups can be color coded. ContactMap supplies a default color scheme, or the user can choose any colors desired.

While not currently implemented, future versions could connect contact lists to address books, phone logs and other digital sources. Web-based updates could handle the chore of keeping up with changes in contact information. Contacts could be shared selectively among work groups or "buddies" as in instant messaging.

ContactMap was tested with 15 users including researchers, managers, administrative assistants, and marketing staff (Whittaker 2004). The tests showed that the mean number of contacts chosen was 95, with a range of 15-184. Even 184 contacts is an easily manageable number to display iconically.

Users grouped their contacts, with a mean of 11 groups and a range of 2-23. Constructing automatic groups seemed like a good idea during the initial design of ContactMap, but after several failed experiments, it was decided to allow users to form their own groups. With the small number of contacts, the grouping task was easy and users even seemed to enjoy reflecting on their social networks as they grouped contacts. The average size of groups was 8, and nearly all contacts appeared in groups. Only 7% were "singletons." The nature of the groups was surprisingly uniform across the test population: workgroups, work projects, friends, family, and special interests such as PTA, rock band, or stock club.

Research on the importance of face to face interaction in everyday communication (Nardi and Whittaker, 2002) suggested that making it easy to use a photo of a contact would be pleasing. Users simply need to locate a digital photo or image and ContactMap will size and place it properly in the map. This feature was popular with those in the user test.

Further testing would be needed to learn more about this issue and other aspects of the user of ContactMap. At this time, ContactMap exists as a prototype but is no longer under development.

6.2. Comparing Soylent and ContactMap

ContactMap and Soylent, then, have similar perspectives on portraying how a user is interconnected with their contacts; both aim to display a social perspective on files and communication. In this section, I would like to discuss some of the similarities and differences between ContactMap, both in approach and implementation.

Soylent and ContactMap are both platforms for handling social network information. While ContactMap is an end-user tool, designed to model groups of people, Soylent is an infrastructure for developing and constructing social workspaces. ContactMap could be built as an end-user application with Soylent.

ContactMap visualizes networks explicitly for users to help them integrate communication and information tasks in a single user interface. In contrast, Soylent uses the networks as a form of background information on how users interact with each other.

Soylent gathers information from an email network, while ContactMap uses an automatically generated list of contacts from which the user manually assembles the network. Both, however, start from the user's communication history as a basis for understanding the set of contacts that should be modeled. There is a shared logic to how both Soylent and ContactMap view social networks. While traditional social network analysis tends to view the broad span of a network, and while tools like Friendster allow users to explore their networks at a distance, Soylent and ContactMap emphasize the user's personal social network. They look only at the people with whom the user has interacted. This information is a reflection of the user's perspective on the world.

Because both ContactMap and Soylent scan only the user's personal email folders (and the user can specify which particular folders to scan), conventional privacy issues do not arise. However, both sets of user tests observed a different privacy problem, which we might call "social privacy." ContactMap, with its photos and color-coded groups, makes the user's personal social network so instantly visible that it reveals quite a lot about how the user thinks about her social world to anyone glancing at the user's desktop. Users would sometimes be embarrassed if they did not have their manager, for example, centrally located on the map.

Similarly, the Soylent display also clearly marks how sets of people are connected. Users were sometimes concerned that connections that they considered important not be too visible to passers-by—or that people not appear to have status on their maps that they do not deserve. Farnham (2002) refers to a parallel case in her user studies, when network visualizations fail to show socially-important people who are poorly connected to their email network in the users' life, such as offline parents.

6.2.1. Distinguishing Soylent from ContactMap

Soylent and ContactMap have taken a number of different choices in their implementation. In this section, I will contrast some of those different choices, with the intent of understanding something more about the space in which both are embedded.

Soylent automatically collects personal information from archives and assembles networks. These automatically-generated networks are incomplete: they undercount face-to-face interaction in favor of email-based communication. ContactMap uses a hybrid approach: while it seeds the network with automatically-collected names, it allows the user to manually create new contacts and to organize the contacts into groups.

ContactMap is structured in a non-hierarchical manner; it allows selection of a group or of its constituent members individually. The use of groups in ContactMap can be less flexible than in Soylent. For example, it would not be possible in ContactMap to specify “all the members of a group, less a few.” This can be mitigated, to some extent, by creating several groups with overlapping membership. In contrast, while Soylent allows generalized access to groups as the clusters of people around a correspondent, these groups do not have a consistent identity within the system. As such, it is more difficult to index information to a specific group.

ContactMap has a strong notion of visualizing the personal social network. The ContactMap designers argue that viewing faces is something like bringing the spirit of face to face contact to computer-mediated communication. The presence of the contact nodes is also important for easy access to contacts and for a place to attach reminders and notifications, as well as a means of displaying groups.

Neither ContactMap, nor the Soylent EE4P interface, show the explicit box-and-line visualization traditionally associated with social networks. Instead, both store network information in the back end. ContactMap stores sets of names; Soylent’s views repeatedly process parts of a large network in order to generate displays and recommendations. While the networks are a useful way to handle social information, they are not necessary—and, indeed, are likely to be confusing as a primary interface.

6.3. The Integrated Workscape

ContactMap shares an important theme with the Soylent infrastructure and tools. Both are steps toward a socially-integrated computer system: one in which people *can be anywhere, and should be everywhere*. If the notion of 'people' becomes a fundamentally available service within the computer, then applications can be adapted to use that information.

The file system, for example, can be extended to consider the people who are involved in it; calendar entries can be annotated with personal information. Files have several groups of people associated with them: those who created the file, who sent it, who edited it—as well as the future steps, those to whom it has been sent, or those who are the ultimate audience. Some of this information might be associated automatically, while other parts might have to be connected manually.

Similarly, word processors and other end-user applications might follow the cues of both Soylent and ContactMap: a document would be automatically connected with the resources and people that helped generate it. While Sally writes the next draft of her paper, for example, Sam's contact—as her teammate in writing the paper—is immediately available *within the word processor,* as both a history correspondence and as a live contact with an instant messaging status.

This calls for a consistent notion of personal identity throughout the operating system. The name for an editor of a file must be connected to their instant messaging identity and their email identity: all collected in one place.

6.4. Building an Infrastructure

Both ContactMap and Soylent act as partial repositories of social and temporal information. Neither, however, keeps a full or disciplined infrastructure for accessing files and messages easily. In this section, then, I discuss how the insights from ContactMap and Soylent could be combined with a search system like Stuff I've Seen (Dumais et al. 2003) and with document meta-information systems like Placeless Documents (Dourish et al. 1999b).

Stuff I've Seen (SIS) acts as a full-text database of files, web pages, and email pages. The database is searchable both by time and by keyword. The interface supports "implicit queries," in which headers from incoming email messages are used to generate queries against the database, which provides associations not unlike the message associations feature of EE4P. SIS also provides temporally-ordered search results that can be keyed against both calendar appointments and historical events.

However, SIS disregards personal information. While names are usable as keywords, the system does not have an internal notion of identity or personal information. Therefore, many of the contact management challenges (discussed, for example, in Chapter 1) cannot be easily resolved. In particular, without a consistent notion of identity, there can be no way to resolve aliases (see Technical

Appendix, section 2) and so users must search under all possible names that a person might appear under.

SIS also does not track personal information as metadata. A file, saved from a message attachment, should be identifiable as connected to that email message, and should be connected to the persons involved. With this information, file attachments saved into the file system gain a social presence, and are then connected to the person. This feature, supported by ContactMap, is not available in a purely text-based search engine. SIS also does not keep information about close associations. Soylent suggests that it could be possible to recommend information based on the people to whom they are connected.

Placeless documents (Dourish, et al. 1999b) is a mechanism for attaching arbitrary metadata to documents. Documents are stored in an infrastructure that gives access to arbitrary labs; the labels come with no semantics at all, but applications can interpret them as needed. Applications and operating system services can then be instrumented to access this information, to tag it, and to dynamically query it. This infrastructure has been used to enable differing views of a shared categorization system (Dourish et al., 1999a), in which local variants of the system could be individually used and manipulated. It has also been used, perhaps more relevantly, to track documents through a workflow system (Dourish et al., 1999c). By tagging a document with its degree of progress, the system could place a document with its work context. Because Placeless adds metadata without altering the current document, the documents could still be edited, worked on, or used with traditional tools.

While Placeless assigns no particular meaning to the labels, the user application certainly may. The Soylent infrastructure could be used to assign Placeless labels, by providing labels that are semantically meaningful about relationships. In such a system, email messages, documents, and communication histories would all be interconnected by labels that connected them to each other and to people.

It is not sufficient, however, to simply annotate each file, or even each piece of data, with a single name. Soylent's field research has reminded us that people and projects are closely associated with *temporal extents* and *social clusters.* Thus, the interconnections between people provide us with valuable information as to how to index their messages and information. In order to fully flesh out this notion of the social workscape, three layers of information are needed.

- First, there must be a layer of personal annotation associated with files and messages. Those annotations connect one or more names with computer resources, and can be associated at a variety of times: at creation time, when emailed, transferred, or received, and similar.
- Second, this information must be able to tie people to *each other*. Some form of data storage should be able to track interconnections between people as they are revealed in shared editing of files, sending and receiving communications, and so on.
- Third, there must a way to specify and learn groups. The results from ContactMap make it clear that allowing *both* manual interconnections between

people is an important and useful task. Group information can, and should, come from a variety of sources: social network information, corporate hierarchy information, and manual choices.

With these three layers of labeling, a social system can then implement the ideas discussed in chapter 1.

6.5. Summary

This chapter discussed the notion of a social workscape which ties computer objects into their social context. Soylent, articulated earlier in the dissertation, was contrasted with another system: ContactMap, an end-user application which displays groups of people on screen. While the development of Soylent has concentrated on the infrastructure aspects, ContactMap has considered a user-interface perspective. The two approaches are largely complimentary: a tool based on ContactMap could be used as an end-user application over an infrastructure based on Soylent.

Last, by contrasting the approaches of ContactMap and Soylent, a better understanding of the possible solution space can be articulated. Understanding ContactMap suggests that Soylent could expand with a cross-cutting concept of "groups" to accommodate user knowledge of interconnections and associations.

Chapter 7. Conclusions

7.1. Overview

In this dissertation, I have shown some ways to use social networks and temporality can to provide meaningful, useful descriptions of the interconnections within groups of people. I used these descriptions to show how software can be developed that supports collaboration within individual, by placing the single-user experience more explicitly in the wider social frame within which works takes place.

Chapter one presented the idea of everyday collaboration, the often individual work that people do when they work on a task that involves other people. It discussed ways that the conceptual divide between CSCW and HCI, and the parallel divide between individual and collaborative work, restricts the abilities of users to accomplish their tasks. Users, of course, find ways around these challenges: they rethink tools and work around problems by making apparent the ways that individual work connects with group work. The chapter suggested that system designers find ways to accommodate those needs by better understanding the structure of online, interactive activity.

Chapter two suggested two ways of accommodate those needs: the tools of social networks and temporality. It presented a variety of online and electronic social network systems that have been oriented toward end-users, and so tries to categorize the breadth of uses for social networks. In doing so, it highlighted the importance of looking at the information immediately around a user, and discusses several different egocentric approaches to data. It also examined several approaches to the notion of "social time", looking at both the important short-term temporal rhythms that shape much of our experience, as well as the long-term changes that occur in group composition and development. It contrasted this textured perspective on time with the usual user interface for temporal interaction, the ordered list, and examined alternative interfaces that bring out an idea of temporal landmarks.

Chapter three presented "Soylent," a system intended to derive contextual information about social interaction. Soylent collects records from mail archives, and stores it in a relational database. Soylent then contains a variety of tools to read and display network views and temporal perspectives on interaction histories. Together, these components give a multi-faceted view of a person's email interactions and network; they present a perspective on how the user interacts with their correspondents. Privacy issues are largely resolved by presenting only a view of information that the user already has: the system is not collecting others' information, but rather aggregating the users' own data.

Chapter four discussed the user tests of Soylent. The tests were oriented toward identifying the recurrent social interaction patterns in email records and interaction histories. They found a series of social patterns, temporal patterns, and combined social and temporal patterns. The social patterns included the "Onion" pattern, which emphasizes a small group that is part of a larger group, and the "Nexus", emphasizing multiple interactions with one person. The temporal patterns look more generally at changes over time, both gradually and quickly.

Chapter five abstracted the patterns to a short list of important aspects: that structures bridge activities, that social structures can disambiguate activities, and that structures may throw current activities into relief. These abstractions, along with the patterns, were used as a way of directing the design of end-user applications that reflect social interaction. The chapter sketched out the idea of a "workspace social proxy," which loosely reflects changes in a worker's schedule, and then illustrated two prototype applications. "TellMeAbout," an awareness tool, shows how social information can be provided in to systems in general; "Enhancing Email for People" adds social information to an email system.

Chapter six looked back at the implications of the designs. It compared Soylent to ContactMap to discuss ways that the "social workscape" might be developed and enhanced. Soylent is an infrastructure tool, while ContactMap has concentrated on a user-interface perspective. By contrasting the approaches of ContactMap and Soylent, a better understanding of the design space became clearer: ContactMap traces a different path to address some similar problems.

7.2. Research Conclusions

In the first chapter, four questions were presented that the dissertation was intended to answer.

- **Are there recurrent social interaction patterns in electronic communication and activities?**

Recurrent social patterns were visible in the data, and users easily identified and recognized their characteristics**.** The patterns made sense to the users, who were able to interpret them as traces of their activities. Chapter 5 presents a pattern catalog of social interaction patterns. These patterns occurred in different contexts, and among different groups of people.

- **How can these patterns be extracted and analyzed? Is it possible to derive recognizable and salient social patterns from these electronic traces?**

A variety of different ways were illustrated of extracting and analyzing these patterns. Electronic traces were extracted from email with the Soylent tool, and could be displayed in several different ways. In particular, the Soylent view (Chapter 4) shows views of the broader network, while TellMeAbout shows a subset of the network centered around a given person.

While these patterns are not *automatically* extracted, the utility of tools like EE4P suggests that the patterns are still extractable.

- **Can these patterns be used to design and develop a software system that is attentive to social and personal roles within the workspace?**

Both TellMeAbout and EE4P use information based on the characteristics from the patterns to present additional information about contacts and communication. While the patterns aren't strictly necessary to understand the networks, the patterns led to a series of important characteristics of personal networks. These characteristics were directly used in implementing the systems that underlie these tools.

- **Can a social software system based around these patterns bridge the gap alluded to earlier between personal and collaborative technologies?**

The discussion at the end of chapter 6 suggests that this gap *can* be bridged. While Soylent itself is too limited in scope to do this bridging itself, a system that has the three-part infrastructure discussed—of annotated data objects, of interpersonal connections, and of group objects – could make many of the tasks that were discussed in the first chapter far easier.

7.3. Limitations and Future Work

As suggested in the previous chapter, Soylent points the way toward a more complete system with better system integration. This section highlights a series of improvements to Soylent that would improve both the interaction of the system and would make progress toward that more complete system.

7.3.1. Speed as a Limiting Factor

While Soylent builds its network based on email messages, neither high-speed access nor efficient data storage was an important development goal. As a result, it was a slow process to both store initial information in the system and to retrieve or view it. This kept users from updating their mail frequently; the inconvenience of the system prevented sustained adoption. As a result, many of the results are more anecdotal than they might be. For example, there are no records of sustained use of any of the applications, and so the changes in patterns have not been tracked over time.

Indeed, there are likely to be far more regularities and patterns to the data than this dissertation observes; the selection presented are results from a careful analysis of the user engagement, rather than being an exhaustive list. Longer user experience might make some of patterns more visible.

7.3.2. Generalized Groups

While Soylent currently has a fixed notion of groups, imposed by the social network that it collects, the system can support other sorts of interconnection information. In the section comparing ContactMap to Soylent-as-infrastructure (section 6.4, page 162), I called for generalized access to groups that can be selected or generated by the user. In this way, Soylent could support hierarchy charts and organizational group membership sets—and could also support the explicit groupings that ContactMap supports.

7.3.3. File System Integration

Soylent does not currently have a notion of file activity. In order to build a tool like the Workscape Social Proxy, the file system would need to be instrumented to monitor changes and edits, and a mechanism would need to link social information to file system objects. As the previous chapter suggests (section 6.4, page 162), one way to address this would be to link Soylent to an infrastructure like Placeless documents.

7.3.4. Text Processing

While this work has so far focused on the formal properties of communication—the message headers and envelopes—then the content of the messages, recent work has suggested that text can be useful as a way of providing additional contextual information. During the user engagement, some subjects had difficulties identifying the history of a particular connection. They were reassured when they learned the subject of the messages: in conjunction with the date and participants, most messages were fully disambiguated.

Recent work (McArthur and Bruza, 2003) has suggested that finding dominant keywords with email messages can be a useful way of organizing and linking between parts of a corpus of messages.

These cues suggest that the "headers-only" approach adopted by Soylent may be too stringent for long-term use, that the additional information available from textual sources may help provide additional cues to users. A version of Soylent that could both *display* cues based on words, and could *find associations* based on text, might show other valuable connections and patterns.

7.4. Conclusion

Much of the activity carried out through standard office applications is, in essence, collaborative; conventional tools are used to coordinated and conduct a wide range of interactions and everyday collaborations. Everyday collaboration is poorly supported by conventional collaborative tool development.

In this work, I have proposed an approach to supporting everyday collaboration. It has discussed prototype applications to help people understand, coordinate and manage the collaboration that they achieve through conventional "single-user" applications. The approach is to make people's "social workscapes" visible – to let them see the structure of their collaborative interactions. We have focused in particular on two sorts of structure – social structure (that is, the patterns of contact and collaboration between people) and temporal structure (that is, how those contacts and collaborations are distributed through days, weeks, and years.)

Bibliography

Ackerman, M. and Starr, B. (1995). "Social Activity Indicators: Interface Components for CSCW Systems". Proceedings of the 1995 ACM Symposium on User Interface Software and Technology (UIST '95).

Adam, B. (1988) "Social Versus Natural Time". In M. Young and T. Schuller, eds. The Rhythms of Society. New York: Routledge.

Alexander, C., Ishikawa, S., and Silverstein, M. (1977) A Pattern Language. New York: Oxford University Press.

Ahuja, M. and Carley, K. M. (1999). "Network Structure in Virtual Organizations", joint issue of Organization Science, Nov./Dec and Journal of Computer-Mediated Communication, 3/4 1998.

Amento, B., L. Terveen, and Hill, W. (2003a). From PHOAKS to TopicShop: Experiments in Social Data Mining. From Usenet to CoWebs: Interacting with Social Information Spaces. C. Lueg and D. Fisher. London, Springer.

Amento, B., L. Terveen, Hill, W., Hix, D., Schulman, R. (2003b). "Experiments in Social Data Mining: The TopicShop System." ACM Transactions on Computer-Human Interaction **10**(1): 54-85.

Begole, J., Tang, J., Smith, R., and Yankelovich, N. (2002). "Work Rhythms: Analyzing Visualizations of Awareness Histories of Distributed Groups." Proceedings of the ACM Conference on Computer-Supported Cooperative Work CSCW 2002 (New Orleans, LA), 334-343. New York: ACM.

Begole, J., Tang, J., and Hill, R. (2003). "Rhythm Modeling, Visualizations and

Applications". Proceedings of the ACM Symp. User Interface Software and Technology UIST 2003. New York: ACM.

Berg, S., A. S. Taylor, and R. Harper. (2003). Mobile phones for the next generation: device designs for teenagers. Proceedings of the 2003 ACM Conference on Human Computer Interaction (CHI '03), Ft. Lauderdale, FL.

Bowker, G.C. and Star, S.L. 1999. Sorting things out: classification and its consequences. Cambridge, Mass.: MIT Press.

boyd, d. (2002). Faceted Id/entity: Managing representation in a digital world. Media Lab, School of Architecture and Planning. Masters thesis from Massachusetts Institute of Technology**:** 118.

boyd, d. (2004). Friendster and Publicly Articulated Social Networking. Proceedings of the 2004 ACM Conference on Human Factors in Computing Systems (CHI '04).

Bradner, E., W. A. Kellog, et al. (1999). The Adoption and Use of 'BABBLE': A Field Study of Chat in the Workplace. Proceedings of the European Conference on Computer-Supported Collaborative Work (ECSCW '99), Copenhagen.

Burt, R. S. (1992). Structural Holes: The social structure of competition. Cambridge, MA, Harvard University Press.

Busch, P., Richards, D., and Dampney, C. (2001). Visual Mapping of Articulable Tacit Knowledge. Australian Symposium on Information Visualisation, Sydney, Australian Computer Society.

Cadiz, JJ, G. Venolia, G. Jancke, and A. Gupta. Designing and deploying an information awareness interface. Proceedings of the 2002 Conference on Computer-Supported Collaborative Work (CSCW 2002)

Carlis, J. and Konstan, J. "Interactive Visualization of Serial Periodic Data." Proceedings of the 1998 ACM Symposium on User Interface Software and Technology (UIST '98).

Carley, K. and Pattison, O. (2004), eds. Dynamic Social Network Modeling and Analysis. Washington: The National Academies Press.

Chiu, P. and Trong, K. (2002). "Interactive Space-Time Maps for Document Visualization." IEEE InfoVis 2002, Interactive Poster and Demo.

Churchill, E., Trevor, J., Bly, S., Nelson, L., Cubranic, D. (2000). Anchored Conversations: Chatting in the Context of a Document. Proceedings of the 2000 ACM Conference on Human Factors in Computing Systems (CHI '00).

Contractor, N., Zink, D., and Chan, M. (1998). "IKNOW: A tool to assist and study the creation, maintenance, and dissolution of knowledge networks." Community Computing and Support Systems: 201-217.

Dodge, M. and R. Kitchin. (2000). Mapping Cyberspace. New York: Routledge.

Domingo, P. and M. Richardson (2001). Mining the Network Value of Customers. Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, ACM Press.

Donath, J. (1995). Visual Who: animating the affinities and activities of an electronic community. Proceedings of the third ACM international conference on Multimedia, San Francisco, California, ACM Press.

Dourish, P. and S. Bly (1992). Portholes: supporting awareness in a distributed work group. Proceedings of the 1992 ACM Conference on Human Factors in Computing Systems (CHI '92), Monterey, California, New York: ACM Press.

Dourish, P., Bellotti, V., Mackay, W. E., & Ma, C.-Y. (1993). Information and Context: Lessons from a Study of Two Shared Information Systems. 1993 ACM Conference on Organizational Computing Systems (COCS '93).

Dourish, P., Lamping, J., Rodden, T. (1999a). "Building bridges: customisation and mutual intelligibility in shared category management." In Proceedings of the ACM Conference on Supporting Group Work GROUP'99. New York: ACM

Dourish, P., Edwards, K., LaMarca, A., Salisbury, M. (1999b). "Presto: An Experimental Architecture for Fluid Interactive Document Spaces." ACM Transactions on Computer-Human Interaction, 6(2). New York: ACM

Dourish, P., Bentley, R., Jones, R. and MacLean, A. (1999c). "Getting Some Perspective: Using Process Descriptions to Index Document History". In Proceedings of the ACM Conference on Supporting Group Work GROUP'99 (Phoenix, AZ), 375-384. New York: ACM.

Ducheneaut, N. and V. Bellotti (2001). E-Mail as Habitat. Interactions. **8:** 30-38.

Ducheneaut, N. (2003). "The reproduction of Open Source programming communities." Unpublished PhD Thesis, UC Berkeley School of Information Management and Systems.

Dumais, S., Cutrell, E., Cadiz, J.J., Jancke, G., Sarin, R., and Robbins, D. 2003. Stuff I've Seen: A System for Personal Information Retrieval and Reuse. Proceedings of the ACM Conference on Information Retrieval SIGIR 2003 (Toronto, Canada). New York: ACM.

Eades, P. (1984). A heuristic for graph drawing. Congressus Numerantium, 42, 149-160.

Erickson, T. Smith, D. N., Kellogg, W. A., Laff, M. R., Richards, J. T., and Bradner, E. (1999) "Socially Translucent Systems**:** Social Proxies, Persistent Conversation, and the Design of 'Babble.'" In Human Factors in Computing Systems: The Proceedings of CHI '99. ACM Press, 1999.

Erickson T. (2000) "Supporting interdisciplinary design: towards pattern languages for workplaces." In Luff, P., Hindmarsh, J and Heath, C. (eds) Workplace Studies: Recovering Work Practice and Informing System Design. Cambridge, Cambridge University Press.

Erickson, T. (2003). Designing Visualizations of Social Activity: Six Claims. Proceedings of the 2003 ACM Conference on Human Factors in Computing Systems (CHI '03), Ft. Lauderdale, Florida, ACM Press.

Eveland, J. D. and T. K. Bikson (1986). "Evolving electronic communication networks: an empirical assessment." Proceedings of the 1986 ACM Conference on Computer Supported Cooperative Work **3**: 91-101.

Farnham, S., Turski, A., Portnoy, W., and Davis, J. (2002). MSR Connections: Exploring Who Knows Whom through Social Networks. Unpublished internal paper.

Fine G. (1990) "Organizational Time: Temporal Demands and the Experience of Work in Restaurant Kitchens." Social Forces 69.1: 95-114.

Fisher, D. and Dourish, P. (2004) "Social and Temporal Structures in Everyday Collaboration." Procedings of the 2004 Conference on Human Factors in Computing Systems (CHI). ACM Press.

Fisher, D. and Moody, P. (2002) "Studies from Automated Collections of Email". University of California, Irvine Institute for Software Research Technical Report. ISR-TR-02-02.

Freeman, E. and Fertig, S. (1995) "Lifestreams: Organizing your Electronic Life." AAAI Fall Symposium: AI Applications in Knowledge Navigation and Retrieval, November, Cambridge, MA.

Freeman, L (2004). "Finding Groups: A Meta-Analysis of the Southern Women Data." In Breiger, R., Carley, K. and Pattison, O., eds. Dynamic Social Network Modeling and Analysis. Washington: The National Academies Press.

Froehlich, J. and Dourish, P. 2004. Unifying Artifacts and Activities in a Visual Tool for Distributed Software Development Teams. Proceedings of the International Conference on Software Engineering ICSE 2004 (Edinburgh, UK), 387-396.

Getoor, L., Friedman, N., Koller, D., & Taskar, B. (2002). Learning Probablistic Models of Link Structure. Journal of Machine Learning Research, 3, 679-707.

Girvan, M. and Newman, M. (2002). "Community Sturcture in Social and Biological Networks." Proceedings of the National Academy of Science, 99, 8271-8276.

Grinter, R. and L. Palen (2002). "Instant messaging in teen life." Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work, ACM Press: 21--30.

Grudin, J. (1989). "Why groupware applications fail: Problems in design and evaluation." Office: Technology and People, 4(3), 245-264.

Gummadi, K., Dunn, R., Saroiu, S., Gribble, S., Levy, H., Zahorjan, J. (2004) "Overlay & peer-to-peer networks: Measurement, modeling, and analysis of a peer-to-peer file-sharing workload". Proceedings of the nineteenth ACM symposium on Operating systems principles (SOSP-19).

Guardiola, X., R. Guimera, A. Arenas, A. Díaz-Guilera, L. Amaral. (2002). "Macro- and Micro-structure of Trust Networks." Preprint (ArXiv.org), cond-mat/ 0206240.

Gutwin, C. and S. Greenberg (1998). Effects of Awareness Support on Groupware Usability. Proceedings of the 1998 ACM Conference on Human Factors in Computing Systems, New York: ACM Press.

Gutwin, C. and S. Greenberg (1999). "The effects of workspace awareness support on the usability of real-time distributed groupware." ACM Transactions on Computer-Human Interaction **6**(2): 243-281.

Havre, S., Hetzler, E., Whitney, P., and Nowell, L. "ThemeRiver: Visualizing Thematic Changes in Large Document Collections." IEEE Transactions on Visualization and Computer Graphics. 8(1): 9-20

Heath, C. and P. Luff (1992). "Collaboration and Control: Crisis Management and Multimedia Technology in London Underground Line Control Rooms." Computer Supported Cooperative Work: The Journal of Collaborative Computing. **1**(1-2): 69-94.

Herbsleb, J., Atkins, D., Boyer, D., Handel, M. Finholt, T. (2002). Introducing instant messaging and chat in the workplace. Proceedings of the 2002 ACM Conference on Human Factors in Computing Systems.

Hill, W., Hollan, J., Wrobleski, D., McCandless, T. (1992). Edit Wear and Read Wear: Text and Hypertext. Proceedings of the 1992 ACM Conference on Human Factors in Computing Systems (CHI '92), New York: ACM Press.

Hook, K., Benyon, D., Munro, A., Eds. (2003). Designing Social Information Spaces: The Social Navigation Approach. Readings in CSCW. London, Springer.

Hudson, S., Fogarty, J., Atkeson, C., Avrahami, D., Forlizzi, J., Kiesler, S., Lee, J. and Yang, J. (2003) Predicting Human Interruptibility with Sensors: A Wizard of Oz Feasibility Study. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2003), 257-264.

Hudson, J.M., Christensen, J., Kellogg, W.A. and Erickson, T. (2002) "I'd be overwhelmed, but it's just one more thing to do": Availability and Interruption in Research Management. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2002), 97-104.

Isaacs, E., W. Alan, et al. (2002). The character, functions, and styles of instant messaging in the workplace. Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work, ACM Press**:** 11--20.

Isbell, C., Kearns, M., Kormann, D., Singh, S., Stone, P.. (2000). "Cobot in LambdaMOO: A Social Statistics Member." Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAAI '00).

Jeong, H., Tombor, B., Albert, R., Otlvai, Z, Barabasi, A. (2000). "The large-scale organization of metabolic networks." Nature. **407.** 5 October, 2000. 651-654.

Jensen, C., Davis, J., and Farnham, S.. (2002). Finding Others Online: Reputation Systems for Social Cyberspaces. Proceedings of the 2002 ACM Conference on Human factors in Computing Systems, Minneapolis, Minnesota, ACM.

Kautz, H., Selman, B., Shah, M. (1997). "Referral Web: combining social networks and collaborative filtering." Communications of the ACM **40**(3).

Kerr, B. (2003). "THREAD ARCS: An Email Thread Visualization," Proceedings of the IEEE Symposium on Information Visualization, Seattle, WA, October 19-21, 2003.

Kleinberg, J. M. (1998). "Authoritative Sources in a Hyperlinked Environment". Proceedings of the 1998 ACM-SIAM Symposium on Discrete Algorithms, San Francisco, CA, ACM Press.

Kullberg, R. (1991) "Dynamic Timelines: Visualizing the History of Photography." Video, Proceedings of the 1996 Conference on Human-Computer Interaction.. ACM.

Lee, A., Girgensohn, A., Schlueter, K. (1997). NYNEX Portholes: Initial User Reactions and Redesign Implications. Proceedings of the 1997 ACM Conference on Supporting Group Work, Phoenix, AZ.

Lockerd, A. (2002). "Understanding Implicit Social Context in Electronic Communication." Masters Thesis from Massachusetts Institute of Technology.

Mackinlay, J. , Robertson, G. and Card, S. (1991). "The Perspective Wall: Detail and Context Smoothly Integrated". ACM Conference on Human Factors in Computing Systems (CHI '91) 173-179.

MacDonald, D. (2003). Recommending collaboration with social networks: a comparative evaluation. Proceedings of the 2003 ACM Conference on Human factors in Computing Systems.

Mallis, M. (2003). "Working a Mars Sol (24.65-hr) Schedule on Earth." NASA Ames' Human Factors: Research and Technology Group news item. December 4, 2003. Retrieved from http://human-factors.arc.nasa.gov/ web/accomplishments/news/news_item.php?id=15

Marshall, C. C. Annotation: from paper books to the digital library. Proceedings of the 1997 ACM Conference on Digital Libraries (DL '97).

Martin, D., Sommerville, I., Rodden, T., Viller, S., and Rouncefield, M. (2001). "Finding Patterns in the Fieldwork." Proceedings of the European Conference on Computer-Supported Collaborative Work (ECSCW '01)

McArthur, R. and Bruza, P. (2003). "Discovery of implicit and explicit connections between people using email utterance." Proceedings of the 8th European Conference on CSCW (ECSCW'03) Helsinki, Finland. Kluewer.

McCarty, C. (2002) "Structure in Personal Networks." Carnegie Mellon Journal of Social Structure. Volume 3, number 1.

McGrath J. E. and J. R. Kelly (1986). Time and human interaction : toward a social psychology of time. New York, Guilford Press (1986).

Moody, J. and White, D., (2003) Social Cohesion and Embeddedness: A Hierarchical Concept of Social Groups. American Sociological Review 68(1):1-25.

Moreland, R. and Levine, J. (1988) "Group Dynamics Over Time: Development and Socialization in Small Groups." In J. McGrath, ed. The Social Psychology of Time: New Perspectives. Newbury Park: Sage.

Morris, M. and Kretzschmar, M. (1997) "Concurrent partnerships and the spread of HIV." AIDS. **11**(6):641-648.

Muller, M., Geyer, W., Brownholtz, B., Wilcox, W., and Millen, D. (2004) "One-hundred days in an activity-centric collaboration environment based on shared objects." Proceedings of the 2004 conference on Human factors in computing systems (CHI 2004)

Nardi, B., Whittaker, S, Schwarz, H. (2002a). NetWORKers and their activity in intensional networks. The Journal of Computer-supported Cooperative Work 11, 205-242.

Nardi, B., Whittaker, S., Isaacs, E., Creech, M., Johnson, J., Hainsworth, J. (2002b). ContactMap: Integrating communication and information through visualizing personal social networks. Communications of the Association for Computing Machinery. April. Pp. 89-95

Nardi, B. and Whittaker, S. (2002). The role of face-to-face communication in distributed work. In Hinds, P. and Kiesler, S, (Eds.), Distributed Work, 83-112. Cambridge, MA, MIT Press.

Newman, M. (2003), Ego-centered networks and the ripple effect, Social Networks 25, 83–95.

Nourie, D. (2001). The Java Technologies Behind ICEMail. developers.sun.com: The Source for Developers. June 2001. http://java.sun.com/developer/technicalArticles/javaopensource/icemail/

Olson, G. M., & Olson, J. S. (2000). Distance Matters. Human-Computer Interaction, 15(2/3), 139--178.

Orlikowski, W. and Yates, J. 2002. It's About Time: Temporal Structuring in Organizations. *Organization Science*, 13(6), 684-700.

Page, L., Brin, S., Motwani, R., Winograd, T. (1998). The PageRank Citation Ranking: Bringing Order to the Web, Stanford University.

Paley, B. (2002). TextArc: Showing Word Frequency and Distribution in Text.

IEEE Symposium on Information Visualization (InfoVis) 2002.

Pipek, V and Wulf, V. (1999) A Groupware's Life. Proceedings of the 6th European Conference on CSCW (ECSCW'99), Copenhagen, September 14-16, 1999, Kluwer, Netherlands, 1999, pp. 199-218

Reddy, M. and Dourish, P. 2002. A Finger on the Pulse: Temporal Rhythms and Information Seeking in Medical Work. Proceedings of the ACM Conference on Computer-Supported Cooperative Work CSCW 2002 (New Orleans, LA). New York: ACM.

Rekimoto, J, "Time-Machine Computing: A Time-centric Approach for the Information Environment", Proceedings of the 1999 ACM Symposium on User Interface Software and Technology (UIST '99).

Resnick, P. and H. R. Varian (1997). "Recommender Systems." Communications of the ACM **40**(3): 56-58.

Resnick, P. (ed). (2001) "RFC 2822: Internet Message Format". The Internet Society: Network Working Group. Available at http://www.faqs.org/rfcs/rfc2822.html

Ringel, M., Cutrell, E., Dumais, S., Horvitz, (2003) "E. Milestones in Time: The Value of Landmarks in Retrieving Information from Personal Stores", INTERACT 2003, September.

Roy, D. "Banana time": Job satisfaction and informal interaction. Human Organization. 18 (1959-1960), 158-168.

Sack, W. (2003). Conversation Map: A Content-based Usenet Newsgroup

Browser. From Usenet to CoWebs: Interacing with Social Information Spaces. C. Lueg and D. Fisher, ed. London: Springer.

Sahami, M., Dumais, S., Heckerman, D., and Horvitz, E. (1998) "A Bayesian Approach to Filtering Junk E-Mail," in Proceedings of the American Association for Artificial Intelligence (AAAI-1998) Workshop on Learning for Text Categorization.

Schmidt, K. (2002). "The Problem with 'Awareness': Introductory Remarks on 'Awareness in CSCW'." Computer Supported Cooperative Work: The Journal of Collaborative Computing. **11**: 285-298.

Shardanand, U. and P. Maes (1995). "Social Information Filtering: Algorithms for Automating "Word of Mouth''. Proceedings of the 1995 ACM Conference on Human Factors in Computer Systems (CHI '95).

Smith, M. (2003). Measures and Maps of Usenet. From Usenet to CoWebs: Interacing with Social Information Spaces. C. Lueg and D. Fisher, ed. London: Springer.

Sorokin, P and Merton, R. (1937) Social time: a methodological and functional analysis. The American Journal of Sociology, 42, 615-629

Suchman, L. (1987) Plans and Situated Actions. Cambridge: Cambridge University Press.l

Tang, J. , de Bonte, A., Raven, M., Isaacs, I. (2000). Instant messaging: products meet workplace users. Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work, ACM Press**:** 363.

Terveen, L., Hill, W., Amento, B. (1999). "Constructing, Organizing, and Visualizing Collections of Topically Related Web Resources." ACM Transactions on Computer-Human Interaction **6**(1): 67-94.

Thimbleby, H., Anderson, S. and Witten, I. (1990) "Reflexive CSCW: Supporting long-term personal work," Interacting With Computers, 2(3), p330-336

Thrift, N. "Vivis Voco: Ringing the Changes in the Historical Georgraphy of Time Consciousness." In M. Young and T. Schuller, eds. The Rhythms of Society. New York: Routledge

Tuckman, B. and Jensen, M. (1977) "Stages of Small-Group Development Revisited." Group and Organization Studies. 2:419-427.

Tyler, J. and Tang, J. (2003). "When Can I Expect an Email Response? A Study of Rhythms in Email Usage". Proceedings of the European Conf. Computer-Supported Cooperative Work ECSCW 2003 (Helsinki, Finland). Dordrecht: Kluwer.

Tyler, J., Wilkinson, D., and Huberman, B. (2003). "Email as Spectroscopy: Automated Discovery of Community Structure within Organization." In Proceedings of the First International Conference on Communities and Technologies.Amsterdam: Kluwer.

van Dantzich, M., Robbins, D., Horvitz, E. & Czerwinski, M. (2002) "Scope: Providing awareness of multiple notifications at a glance." In Proceedings of the 2004 ACM Conference on Advanced Visual Interfaces, Trento, Italy

Venolia, G., Neustaedter, C. (2003) Understanding sequence and reply relationships within email conversations: a mixed-model visualization, Proceedings of the conference on Human factors in computing systems, April 5-10, 2003, Ft. Lauderdale, Florida, USA

Viégas, F., Wattenberg, M., Dave, K. "Studying cooperation and conflict between authors with history flow visualizations." Proceedings of the 2004 conference on Human factors in computing systems.

Viegas, F., boyd, d, Nguyen, D., Potter, J., and Donath, J.. (2004) Digital Artifacts for Remembering and Storytelling: PostHistory and Social Network Fragments. HICSS-37, Persistent Conversation Track.

Wasserman, S. and K. Faust (1994). Social Network Analysis. Cambridge, Cambridge University Press.

Wellman, B. (1993). An Egocentric Network Tale. Social Networks **17**(2): 423-436.

Wellman, B., Salaff, J., Dimitrova, D., Garton, G., Gulia, M., Haythornthwaite, C. (1996). Computer Networks as Social Networks: Collaborative Work, Telework, and Virtual Community, Annual Review of Sociology, 22, 213-238.

Wexelblat, A. (2003). Results from the Footprints Project. Designing information spaces: the social navigation approach. K. Hook, D. Benyon and A. J. Munro. London, Springer.

Whittaker, S. and Sidner, C (1996) Email Overload: Exploring Personal Information Management of Email. Proceedings of Conference on Human Factors in Computing Systems (CHI 1996): 276-283

Whittaker, S., Jones, Q., Nardi, B., Creech, M., Terveen, L., Isaacs, E., Hainsworth, J. (2004). ContactMap: using personal social networks to organize communication in a social desktop. Submitted to Transactions on Computer Human Interface.

Whittaker, S., Davis, R., Hirschberg, J., & Muller, U. (2000). Jotmail: a voicemail interface that enables you to see what was said. In Proceedings of Conference on Human Factors in Computing Systems (CHI 2000), 89-96, ACM.

Whittaker, S., Jones, Q. Terveen, L. (2002). Contact management: identifying contacts to support long term communication. Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work.

Xiong, R. and J. Donath (1999). PeopleGarden: Creating Data Portraits for Users. Proceedings of the 12th annual ACM symposium on User interface software and technology, Asheville, NC., ACM.

Zerubavel, E. (1981). Hidden Rhythms: Schedules and Calendars in Social Life. Berkeley: University of California Press.

Zerubavel, E. (1985). The Seven-Day Circle: The History and Meaning of the Week. Detroit: Free Press

Zerubavel, E. (2003) Time Maps: Collective Memory and the Social Shape of the Past. Chicago: University of Chicago Press.

Web References

Amazon.com: http://www.amazon.com

Atlas: http://www.cybergeography.org/atlas/atlas.html

BuddyZoo: http://www.buddyzoo.com

Blogosphere Ecosystem: http://www.myelin.co.nz/ecosystem/ and http://www.truthlaidbear.com/EcoFAQ.php

Popularity: http://www.daypop.com; http://www.popdex.com; http://blogdex.media.mit.edu

Slashdot: http://slashdot.org/moderation.shtml

Advogato: http://www.advogato.org/trust-metric.html

PGP: http://bcn.boulder.co.us/~neal/pgpstat/ http://dtype.org/keyanalyze/explanation.php

Friendster: http://www.friendster.com/

SixDegrees: http://www.sixdegress.com

Appendix 1. Database fields in Soylent

The Soylent database was initially designed to store email records. However, the mechanism is fairly general; it may be expanded to record other conversation types. In general, the format consists of six tables: they are, in turn, mappings that store information about the message *envelope*, the *recipients, file attachment* information, *metainformation* about the mail collection, a table of *aliases* (see Appendix 2) , a table of *real names*.

In this appendix, I will briefly describe each of the tables.

- the *unique IDs and dates* of conversations

This first table shows the format of the "SENDER" table. The roman-face text gives the general column type, while the italicized text shows the application of that column to email. The general column type is applied should the table be used to store other sorts of conversational or collaborative information, such as file exchange or instant messaging. Primary keys are marked with (*); external with (2).

Table A-1. Envelope information.

Unique ID	Sender's name (2)	Date	Object label	Type of interaction
Message ID	*Mail from*	*Message send date*	*Mail subject*	*"EMAIL"*

- the *recipients and participants* in interactions

In order to simplify graph analysis, the sender's name is also stored in the receiver table. Therfore, any email message generates two or more "RECIEVER" entries. These two entries are stored in a table that connects the message to the receivers. The table also stores information on whether that name is in the "To", "From", or "CC" field

Table A-2. Recipients and Participants

UID(2)	Recipients's name (2)	Type of sender
Message ID	*Mail To/From*	*"CC" / "To" / "From"*

- *file attachment* information

This table stores the names of files that were attached to email messages, and, more generally, external resources. This table is used to store information about attachments: content header data that may be associated with the unique message IDs. In general, while there are two fields (one for an attachment name, the other for the resource type, this table can generally hold a link to external information source.

Table A-3. Attachment table

UID (2)	Resource identity	Resource type
Message ID	Attachment name	Attachment type

- *metainformation* about the data collection (e.g. the date that mail file was scraped)

This table stores information about the data collection. While in this implementation it was used as a record of each time that data was added to the system, it more generally allows the system to know more about what sorts of data it can expect to find in the various tables above.

- *aliases,* storing alternate names for people

While the alias table will be discussed in more detail in the next appendix, it generally was used to associate a single canonical email name with the list of alternate email addreses that they might have.

- *real names*, storing a single human-readable name for people

The canonical naming systme in Soylent is storing an email address. However, reader-friendly names are often available, and so this table stores a mapping from a canonical name ("danyelf@acm.org") to a reader-friendly name ("Danyel Fisher").

Appendix 2. Handling Aliases and Cleaning Email

Aliases are alternate names for individuals. Within a single company, people might have several similar names: for example, **first_lastname@local.corp.com**, **lastname@corp.com**, and even **nickname@previouscorp.com**. This leads to several difficulties: if messages are indexed by name, for example, these three strings might show as distinct persons. A search engine that naively searched under names might therefore miss the fact that all are the same, and thus might force the user to repeatedly search. This is a common problem with some search-oriented interfaces.

Soylent therefore maintains three different conceptions of user identity, which it reconciles to a single name. It keeps a *single* canonical email address for a person; a *list* of non-canonical email addresses that map to that person; and a single canonical *real name*. The remainder of the task is to maintain a database that links the non-canonical email addresses to the canonical address, and the real name to the canonical email address.

Soylent takes a hybrid manual and automatic approach to resolving these alias problems. While it stores all information within a database under the name that appeared in the email message, it also maintained an alias table to store alternate names for users. This table is generated automatically, but can be filled in by the user. When constructing the network diagram, Soylent cross-references all names that had been matched together.

It turns out that assigning aliases is both one of the more important tasks during the Soylent initialization phase, and one of the more irritating for users. Soylent therefore breaks the process into three stages.

- First, it reconciles sub-domain matches. Thus, **jpd@ics.uci.edu** and **jpd@uci.edu** would be merged together. This stage is followed by a user-confirmation step offering the user an opportunity to uncheck mistaken pairings.
- Second, it automatically collects obvious real name matches. If two different email addresses share the same real-name field, it merges the names. Thus, "Paul Dourish" would have **jpd@ics.uci.edu** linked to **paul@dourish.com**. Again, this step is followed by an opportunity for the user to prevent mistaken pairings.
- Last, it offers the user a manual entry field. A user can select names from the current alias table, can remove aliases from the list, and can merge sets of aliases together.

With this system, the user is able to tell the system about aliases. Unfortuantely, automated solutions to this problem are difficult: these different names often follow social contexts, and become institutionalized in address books. A group of people who know someone as **paul@dourish.com** may not quickly change their references to **jpd@ics.uci.edu**, even after several years.

Appendix 3. The Soylent-TellMeAbout API

The Soylent class "sna.api.ApiFirstClass" provides the Soylent API for external calls. In this section, I briefly outline the calls available. This gives some background information on the capabilities of the Soylent tool.

- Initialization

`void` **`prep`**`(MiniMailEntryCollection collection)`

A "MiniMailEntryCollection" is a data structure that points to a database and the query functions that access it. (In particular, the interface can read the Soylent database as a bipartite graph from messages to people.)

- Graph Access

`BipartiteGraph` **`getBipartiteGraph`**`()`
`Graph` **`getFoldedGraph`**`()`

These functions access the underlying graphs in JUNG format. The bipartite graph maps `NameVertexes` to `MessageVertexes`; the folded graph maps `NameVertexes` to each other. A `MessageVertex` stores a unique identifier for a message and a date; a `NameVertex` stores a name.

`NameVertex` **`getBipartiteNameVertex`**`( Name name )`

This function gets the NameVertex that stores the name.

- Queries

`DatePair` **`getMeaningfulDateRange`**`()`

This function provides information on the date range over which the graph extends.

`NameVertex[] getTopFew()`

This function returns all of the names, in order of the number of messages they have exchanged with the user. With it, a client can find the most popular few contacts.

```
Set findIncomingMessages(Name name)
```

This function gets all incoming messages assocatiated with the input name.

- Person-Specific Queries

```
InquiryClassAttachment getAttachmentInfo(Attachment attach)
InquiryClassMessage getMessageInfo(Message message)
InquiryClassName getNameInfo(Name name)
```

These three functions all return InquiryClass objects. These contain type-specific queries that discuss a specific message, person, or attachment. The InquiryClassName, for example, contains functions to get the number of messages sent per month, the set of attachments exchanged between the user and the person, and the neighbors with whom this person shares more than a number of ties.

The InquiryClassMessage finds other messages with overlapping populations of users.

With the methods outlined here, a system can retrieve stored associations betwen people, messages, and attachments, indexed both graphiclaly and temporally.