# Designing Tools to Increase Group Awareness in the Work Place

by

Lu Zhang

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved November 2010 by the
Graduate Supervisory Committee:

Hari Sundaram, Co-Chair
Gang Qian, Co-Chair
Aisling Kelliher

ARIZONA STATE UNIVERSITY

December 2010

ABSTRACT

This thesis investigates the role of activity visualization tools in increasing group awareness at the workspace. Today, electronic calendaring tools are widely used in the workplace. The primary function is to enable each person maintain a work schedule. They also are used to schedule meetings and share work details when appropriate. However, a key limitation of current tools is that they do not enable people in the workplace to understand the activity of the group as a whole. A tool that increases group awareness would promote reflection; it would enable thoughtful engagement with one's co-workers.

I have developed two tools: the first tool enables the worker to examine detailed task information of one's own tasks, within the context of his/her peers' anonymized task data. The second tool is a public display to promote group reflection. I have used an iterative design methodology to refine the tools.

I developed ActivityStream desktop tool that enables users to examine the detailed information of their own activities and the aggregate information of other peers' activities. ActivityStream uses a client-server architecture. The server collected activity data from each user by parsing RSS feeds associated with their preferred online calendaring and task management tool, on a daily basis. The client software displays personalized aggregate data and user specific tasks, including task types. The client display visualizes the activity data at multiple time scales. The activity data for each user is represented though discrete blocks; interacting with the block will reveal task details. The activity of the rest of the group is anonymized and aggregated. ActivityStream visualizes the aggregated data via Bezier curves.

I developed ActivityStream public display that shows a group people's activity levels change over time to promote group reflection. In particular, the public display shows the

anonymized task activity data, over the course of one year. The public display visualizes data for each user using a Bezier curve. The display shows data from all users simultaneously. This representation enables users to reflect on the relationships across the group members, over the course of one year.

The survey results revealed that users are more aware of their peers' activities in the workspace.

*To my parents and grandma*

ACKNOWLEDGMENTS

I would like to give my sincere gratitude to Professor Hari Sundaram for his continuous support and exhaustive guidance for my master study and research, for his patience, enthusiasm and immerse knowledge. Those precious advice and suggestions, both in my academic area and in my life, would be priceless and lead to my success in the future.

I would like to express my sincere appreciation to Professor Aisling Kelliher for providing me many suggestions and insights in Reflective Living research group. Her passion and broad knowledge make me feel what a fun thing research is.

I would like to express my thanks to Professor Gang Qian for being my academic advisor, giving illuminating advises on coursework and being my committee member.

I am thankful to Assegid Kidane (Ozzie) for his excellent help in TDL lab and supporting my research work. His experience on physical computing helps me get through many difficulties.

I also would like to thank Professor Todd Ingalls for giving valuable thoughts and advise on media theory.

I am thankful to other Reflective Living group members: Shawn Nikkila, Ryan Spicer, Silvan Linn, Munmun De Choudhury, Heng Chen, Ke Xu, Brandon Mechtley, Li Zhen and Yu-ru Lin. They gave me millions of help on research. Their broad knowledge offered me many inspirations.

TABLE OF CONTENTS

Page

LIST OF TABLES ..... viii

LIST OF FIGURES ..... ix

CHAPTER

1. INTRODUCTION ..... 1

1.1. Background ..... 1

1.2. Challenges and Problems ..... 2

1.3. Summary of Our Approach ..... 4

1.4. Organization ..... 7

2. RELATED WORK ..... 8

2.1. Calendar Display ..... 8

2.2. Public Display ..... 13

2.3. Group Awareness and Collaboration ..... 16

2.4. Reflective Design ..... 20

3. ITERATIVE SYSTEMS DESIGN ..... 22

3.1. Iterative Design Methodology ..... 22

3.2. Earlier Work- Task Visualization Public Display ..... 23

3.2.1. Motivation and Design Principle ..... 23

3.2.2. System Architecture and Data Collection ..... 24

3.2.3. Approach and Design ..... 26

3.2.4. System Weaknesses ..... 33

3.3. Latest Work- ActivityStream Desktop Tool ..... 34

3.3.1. Motivation and Design Principle ..... 34

3.3.2. System Architecture and Data Collection ..... 34

CHAPTER Page

3.3.3. Approach and Design ..... 37
3.3.4. System Weaknesses ..... 49
3.4. Latest Work- ActivityStream Public Display ..... 50
3.4.1. Motivation and Design Principle ..... 50
3.4.2. System Architecture and Data Collection ..... 52
3.4.3. Approach and Design ..... 53
3.4.4. System Weaknesses ..... 56
3.5. Summary ..... 57
4. USER EXPERIMENTS AND RESULT ANALYSIS ..... 58
4.1. User Experiment and Result Analysis for Task Visualization Public Display ..... 58
4.1.1. User Experiment Design ..... 58
4.1.2. Questionnaire Results and Analysis ..... 59
4.1.3. Interview Summary ..... 64
4.1.4. Summary ..... 66
4.2. User Experiment and Result Analysis for ActivityStream Desktop Tool ... 66
4.2.1. User Experiment Design ..... 66
4.2.2. Questionnaire Results and Analysis ..... 67
4.2.3. Interview Summary ..... 71
4.2.4. Summary ..... 72
4.3. User Experiment and Result Analysis for ActivityStream Public display .. 72
4.3.1. User Experiment Design ..... 72
4.3.2. Interview Results and Discussion ..... 73
4.3.3. Summary ..... 76

CHAPTER Page

4.4. Summary of User Experiments ..... 76

5. CONCLUSION AND FUTURE WORK ..... 78

5.1. Conclusions ..... 78

5.1.1. Conclusions of Iterative Design ..... 78

5.1.2. Potential Extension of Current Work ..... 81

5.1.3. Future Direction ..... 82

REFERENCES ..... 83

LIST OF TABLES

Table Page


1. Summary of iterative design, shows how the systems are refined. The symbol "√" means the system has that property. ..... 5
2. Result of usability test ..... 59
3. Result for functionality test ..... 60
4. Result of visual design test ..... 61
5. Vote count for the busiest person depending on amount of tasks. Every user is asked to provide three busies people in RL group ..... 61
6. Count of complete tasks, incomplete tasks, total tasks for each user ..... 62
7. The busiest months for users based on statistical results ..... 63
8. The busiest month for each user ..... 63
9. Survey for calendar tool use, red areas mean that the user uses the calendar tool, green areas mean that this tool is a major tool for that user ..... 64
10. Survey for calendar tool use, green areas represent the calendar tools users use 67
11. Results of pre questionnaire and post questionnaire ..... 68
12. Statistical significance test ..... 69
13. Statistic results of question 9 and question 10 in post questionnaire. Counts represent the number of users who choose that option ..... 70
14. Feedback to ActivityStream interfaces ..... 71
15. Summary of iterative design, shows how the systems are refined. The symbol "√" means the system has that property. ..... 78

LIST OF FIGURES

Figure Page


1. Structure of time: (a) linear time; (b) cyclic time; (c) branching time ..... 8
2. Augur system, viewing a colleague's calendar ..... 9
3. Use a 3D model to show power demand, displayed as a function of hours and days ..... 10
4. A view of an individual patient's data in the KNAVE-II prototype ..... 10
5. The Spiral calendar visualizer ..... 11
6. The Time Lattice visualizer.. ..... 12
7. Online Enlightenment.. ..... 13
8. The Notification Collage ..... 14
9. Panarama represents an "Idle" environment ..... 15
10. The Semi-Public Display prototype. Clockwise from top left: a) Collaboration Space; b) Active Portrait; c) Attendance Panel; d) Reminders ..... 16
11. Newsgroup Crowd visualization of alt.politics.bush. ..... 18
12. AuthorLines visualization of an author that starts about as many thread as the ones they respond to ..... 18
13. Client-server architecture ..... 25
14. RSS feed parser ..... 26
15. Group view of the Task Visualization Public Display ..... 27
16. Individual view of the Task Visualization Public Display. ..... 28
17. Task blocks with different colors with blocks stacked in order of size ..... 28
18. Cartoon thumbnails for all users ..... 31
19. Three facial expressions, from left to right: calm, happy and stressed ..... 31
20. Information bubble shows task content. It supports three lines of text ..... 32

Figure Page

21. Client-server architecture. 35
22. ICS file parser. 36
23. View of the ActivityStream desktop tool when users browse their own calendar data. 37
24. View of ActivityStream desktop tool when users browse others' calendar data. 38
25. Login page. 39
26. Discrete blocks with different colors represent activities. 40
27. Activity blocks are stacked in an order of size. 41
28. Daily view of activity data. 43
29. Weekly view of activity data. 43
30. Monthly view of activity data. 44
31. Interaction in activity visualization-clicking on a brick to view detailed information. 45
32. The system uses four curves to represent aggregate activity information. 46
33. Construction of a quadratic Bezier curve. 47
34. User's activity compared with average curve of group activity. 49
35. Green arrows show the people users collaborated with 49
36. Client-server architecture 52
37. View of ActivityStream Public Display during the animation. 53
38. Curve changes from starting curve to ending curve. 55
39. Recalculate one section of the curve. 56
40. Task counts for each user in 4 months. 62

# CHAPTER 1

# INTRODUCTION

## 1.1. Background

The evolution of the nature of workspaces over the past decades has affected the way people work and collaborate. Before the emergence of cubicles in the workspace, many people used to work in the same room without partitioning boards. Consequently, a quick glance would enable them know what others were doing and whether they were likely to be present or available for a meeting or discussion. This encouraged individuals at a workplace to talk to others directly and serendipitously as and when they would seek their help, suggestions or advice. In other words, such working environment made communication between workers easier in the workspace.

However, as the sizes and work practices in corporations and organizations began to grow from the late eighties, separation of work areas started to evolve as a common practice. This led to the segregation of each individual in a room or in a partitioned area also known as a cubicle. A cubicle is partially enclosed workspace, separated from neighboring workspaces by partitions that are usually 5-6 feet tall [1]. There were several motivations to promote this kind of workspace logistics. Office cubicles seemed to improve working efficiency, at the same time had the potential to isolate office workers from noise and distractions of an open office, protect workers' privacy and help them concentrate on their work.

Note that despite these advantages, segregation of workspaces of individuals made it difficult for workers to communicate with others. Workers were obliged to walk to their peers' cubicle desk to know whether they are present or available to talk. Office cubicles thus became an obstacle to promoting group communication and awareness. Note that awareness is "an understanding of the activities of others which provides a context for

your own activity" [2]. Lack of group awareness thus influences group coordination and collaboration in a negative manner, and decreases group productivity indirectly.

In order to combat this issue, electronic calendars have started to be widely used in contemporary working culture. Before the emergence of electronic calendars, people used paper-based tools such as notebook, stick notes to record their tasks. While paper-based calendars are portable and easily accessible, however they are hard to backup and share with other people. Electronic calendars such as Google calendar, Outlook, iCal and Remember the Milk solve those problems in traditional calendars and have been observed to be used extensively in contemporary work culture in large globally distributed organizations. There are many advantages of using online calendars. The calendar data backed up in the server is secure. By sharing calendar with others, people are able to browse their co-workers' calendar data in their own calendar system. Sharing calendars ease workers' coordination and collaboration. In addition, most electronic calendars provide multi-timescale visualization, which is good for visualizing long-period time series data. Today, people use electronic calendars to schedule their work, organize their daily life, and share it to their co-workers. The collection of calendar data in a group reflects the group activity.

### 1.2. Challenges and Problems

We are interested in designing calendar visualization tools to increase group awareness and promote group reflection, which involves the following research problems:

- *How can we display detailed and aggregate information of calendar data? In what kind of circumstances, does the system provide detailed information or aggregate information about group activity to people?* Detailed calendar information includes every single task's details, and aggregate information only shows summary of tasks during a certain period. The calendar system should visualize detailed information

and aggregate information in different ways. Many existing calendar systems only provide detailed information or aggregate information. For example, Tullio [3] introduced a calendar system called "Augur", which provides details about personal calendar information. However, "Augur" does not provide aggregate information so that the pattern about group activity is not clear in the system. Besides, different groups need different level information about group activity. Some groups of people need detailed, up-to-date information about group activity, and some groups of people only want to keep high-level awareness about group activity. It is difficult to decide the type of information that is adequate to be offered to different classes of users.

- *How do we avoid privacy issues when people share calendar data in the calendar visualization tools?* When people share calendar information in the system, sometimes it is possible that they would not want to share their detailed information to others. Some existing calendar systems such as Time Lattice supported multiple users' calendar visualization [4], but they raise privacy issues. To tackle this, researchers tried to minimize the privacy issues by providing lightweight information [5], but lightweight information is still likely to include detailed information about group activity. Consequently, we need to find a way to solve privacy issues in the system, and make users feel comfortable to share their calendar data to others.
- *How do we solve the scalability issues in the system when group size increases?* Many calendars systems have scalability issues when they display multiple users' calendar data. Huang [5] introduced a semi public display to provide lightweight information about group activity, but the size of the group was considerably limited: the system did not support a group that has more than nine members. Additionally, when Google calendar system displays multiple users' calendar data, the task and

event information overlap if they happen at the same time. To avoid these overlap issues, Outlook displays the multiple users' calendar data by using separate columns. Each column shows one user's calendar data. However, the amount of columns increases with the increase of amount of users. When the amount of columns is large, each column only provides a narrow view for displaying calendar data. It is challenging to find a way to visualization multiple users' calendar data in proper way when the group size increases.

### 1.3. Summary of Our Approach

We designed our calendar tools based on the dimensions below:

- Detailed information versus Aggregate information

  Detailed information shows details about group activity, while aggregate information shows summary of group activity.

- Discrete information versus Continuous information

  Discrete information means the display uses discrete objects to visualize calendar data, while continuous information means the display uses continuous objects to visualize calendar data.

- Animated display versus Static display

  Animated display shows an animation in the display to users to browse different aspects of the calendar information, while static display shows static visualization.

- Single timescale versus Multi-timescale

  Single timescale means the display only uses one timescale to visualize data while multi-timescale means the display uses multiple timescales to visualize data.

- Public display versus Desktop tool

  Public display is a system that is set in public space, all the people are able to watch. Desktop tool is the system that is accessible with PC.

- Small groups versus Large groups

  "Small groups" means the system is designed for small groups, while "large groups" means the system is designed for large groups.

We use iterative design methodology to design and refine our tools. Table 1.1 shows how we refine the calendar visualization systems based on the dimensions discussed above.

Table 1.1: Summary of iterative design, shows how the systems are refined. The symbol "√" means the system has that property.

| **Properties** | **Task Visualization Public Display** | **ActivityStream Desktop Tool** | **ActivityStream Public Display** |
|---|---|---|---|
| Public display | √ | | √ |
| Desktop tool | | √ | |
| Detailed information | √ | √ | |
| Aggregate information | | √ | √ |
| Privacy concerns | | √ | √ |
| Discrete objects | √ | √ | |
| Continuous objects | | √ | √ |
| Static | | √ | |
| Animation | √ | | √ |
| Single time scale | √ | | |
| Multi-timescale | | √ | |
| Scalability concerns | | | √ |

The first segment of our work is the design a public display to provide detailed task information to a small group of people. The purpose of the system is to provide up-to-date, detailed task information in the group. We design the system as a public display to make it easily accessible to everyone in the group. The calendar system collects data from Remember the Milk by parsing RSS feeds. To show detailed information, we use discrete objects to visualize every single task data. To show both all task data in the group and task data for each individual, we run an animation in the display, which goes through group's task information and each individual's task information. This system

shows four months data with one time scale, where the time unit is one week. At the early stage, this calendar system only supports a small group of people. Based on the user feedback, the system is geared towards collecting more calendar data from other calendar tools, add interactions, support multi-timescale visualization, provide aggregate information about group activity and solve privacy issues. This comprises the next segment of research presented in this thesis.

The first piece of our latest work is ActivityStream desktop tool, which enables users check detailed information of their own task data, in context of their peers' anonymized task information. To handle the privacy issues in our earlier work, the desktop tool has a login page to protect users' privacy. In contrast to our earlier work, ActivityStream collects data from more calendar tools: Google calendar, Outlook, iCal and Remember the Milk. The display is static because users access information by interacting with the tool. Besides, the system offers three timescales, and visualizes the daily, weekly and monthly calendar data. To show aggregate information about group activity, ActivityStream uses continuous object –curves to represent workload in the group. The user feedback reveals users are more aware of their own activities and their peers' activity. However, the system needs to add hourly data visualization, solve scalability issues when the group size increases and support multi users' calendar data visualization for comparison.

The second piece of our latest work is ActivityStream public display, which provides high-level, aggregate information about group activity to a large group of people. The system shows an animation to explain how group's workload changes over time. Our key motivation in this research segment is that public display is a better way than desktop tool to provide ambient information for a large group of people. Since the system is designed for large groups, it only provides aggregate information to minimize privacy issues.

Compared to ActivityStream desktop tool, ActivityStream public display supports multiple users' calendar information, and shows everyone's workload curve at the same time for comparison. To solve the scalability issues, the system draws one curve to represent a person's calendar data. The computational complexity of the system is linear to the increase of group size. The curves' density increases with the increase of group size. The user feedback reveals that users become more aware of the stressful group activity when curves' density increases.

## 1.4. Organization

This thesis has five chapters. The current chapter introduces the background, the challenges and problems, the summary of our approach and organization of this thesis. Chapter 2 discusses prior work on calendar display, public display, group awareness and reflective design. Chapter 3 discusses our earlier work - task visualization public display first, and addresses the problems that need to solved to accomplish our research goals. Thereafter, based on iterative design methodology, chapter 3 indicates latest work: ActivityStream desktop tool and ActivityStream public display. This chapter explains these two systems with discussion of their approach and design. Chapter 4 discusses user experiments and analysis of our systems. Finally, chapter 5 presents conclusion of our earlier work and latest work, the improvement of current work and future research directions.

# CHAPTER 2

# RELATED WORK

## 2.1. Calendar Display

Electronic calendars such as Google calendar, iCal, Outlook and Remember the Milk are widely used in the workplace. People use calendars to organize their work in daily life and share their work with other co-workers. A good calendar display supports multiple users' calendar data and provides multi-timescale visualization of a summary of calendar data. The patterns of Calendar data reveal the summary of calendar information during a certain period, and indicate to people how the workload changes over time.

Some prior works focus on how to visualize time series data. In [6] the author categorizes the techniques for visualizing time-oriented data. He indicates three time structures: linear, cyclic and branching.

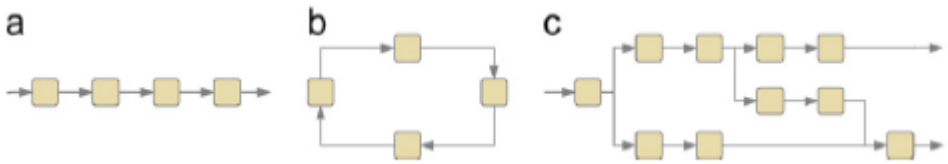



Figure 2.1: Structure of time: (a) linear time; (b) cyclic time; (c) branching time

Linear time corresponds to natural perception of time as being a (totally or partially) ordered collection of temporal primitives, i.e., time proceeds from the past to the future. A cyclic time axis is composed of a finite set of recurring temporal primitives (e.g., the season of the year). Branching time axes are modeled as graphs. Temporal primitives are the vertices of the graph. Directed edges describe temporal order. Vertices with more than one outgoing edge indicate a split of the time axis into alternative scenarios. Linear time and cyclic time structure are special cases of branching time structure. Branching time structure is linear time structure in the case that every vertex has no more than one outgoing edge. Branching time structure is cyclic time structure when the graph is a circle.

Our earlier work, task visualization public display and one aspect of our latest work, ActivityStream public display use cyclic time structure because the system runs an animation from the start time point to end time point. The other aspect of our latest work, ActivityStream desktop tool uses linear time structure because the system provides an unlimited timeline for users to browse their calendar data.

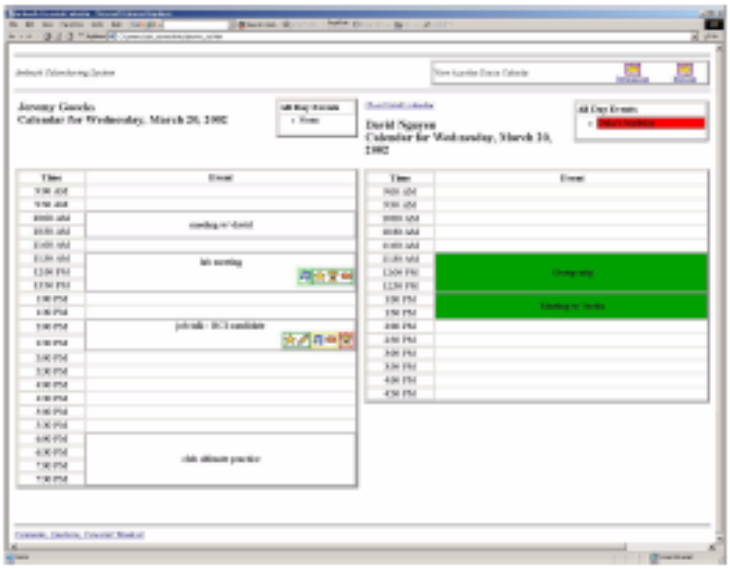


Figure 2.2: Augur system, viewing a colleague's calendar

In [3], the author indicates a groupware calendar system called *Augur*, which not only shows personal calendaring practice, but also shows co-scheduled event that may benefit group awareness. However, the augmented daily calendar only provides details about calendar activity, and does not show aggregate or summary information about calendar data.

It is important to visualize time-oriented data on multiple timescales to provide aggregate information, and show calendar data patterns clearly. In [7], the author presents a new solution to identify patterns and trends on multiple time scales. The solution is to cluster similar daily data patterns and visualize these average patterns as graphs. The author uses a 3D model to visualize electrical power consumption. The 3D model has two

time axes: one axis is day based, and the other is hour based. The 3D model offers rich information, and provides comparison of data patterns on different timescales (the patterns is the summary of the data's properties). However, the 3D model is only able to show two timescales. A high-dimensional model is needed for visualizing multi-timescale data. The computational complexity of this model increases dramatically with the increase of time scales.

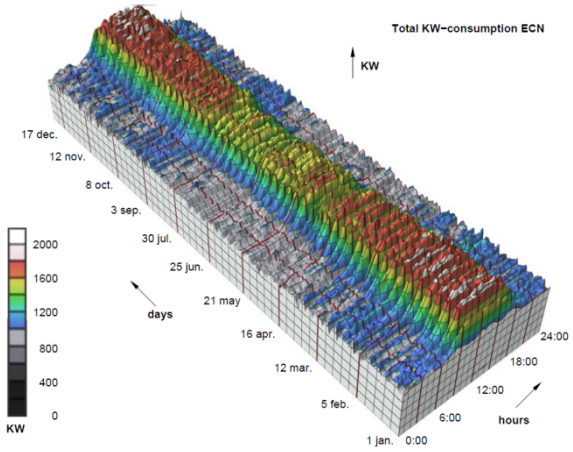



Figure 2.3: Use a 3D model to show power demand, displayed as a function of hours and days

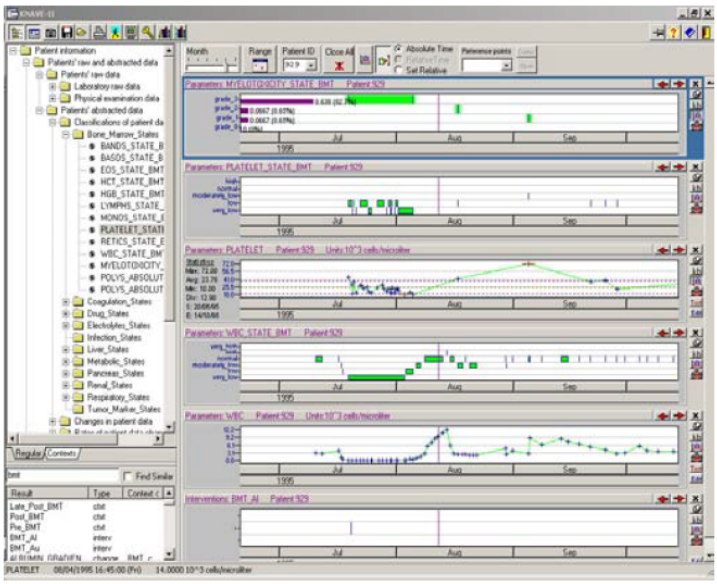


Figure 2.4: A view of an individual patient's data in the KNAVE-II prototype

In [8] the author creates an interface called “KNAVE-II” to enable user to query, visualize and explore clinical time-oriented databases. “KNAVE-II” shows clinical data at different levels of temporal granularity along absolute (calendar-based) time-lines. Compared to 3D models, “KNAVE-II” is a better method of displaying data with more time scales. Our ActivityStream desktop tool system shows data at three levels of temporal granularity along timelines: daily, weekly and monthly. Users are able to switch between different time scales with one click.

In [4], the author presents two different calendar visualizations: “Spiral Calendar” and “Time Lattice”. “Spiral Calendar” makes efficient use of screen space since small context can give large details. The animation is smooth and attractive, and increases the real-time interaction between a user and an electronic calendar. However, it does not provide category or priority information that people may need to organize their activities. The visualization shows detailed information like a traditional calendar does, it does not show aggregate information or any pattern based on large calendar data. Furthermore, “Spiral Calendar” does not support sharing and visualizing a group of calendars.

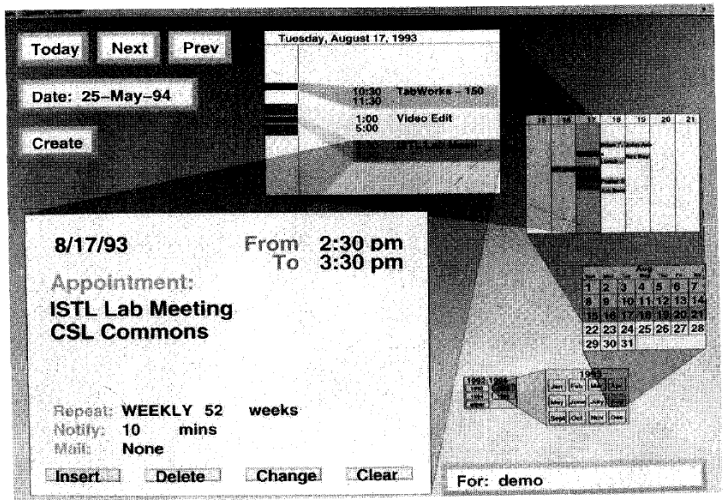



Figure 2.5: The Spiral calendar visualizer, which is designed for rapid access to an individual's daily schedule. A spiral layout combines detail and context in an intuitive 3D layout that allows the connection among calendars to be visible.

Another system is “Time Lattice”, which is useful for comparing multiple calendars, and contributes to scheduling meetings or planning group activities. “Time Lattice” aligns multiple calendars in 3D for comparison, the vertical axis represents hours, the horizontal axis represents days and the depth axis represents individuals. The system uses a shadow metaphor to provide aggregate views of information. The shadow on the left wall is a view of the information that aggregates the individual calendars so that the user can see which time slots are relatively available. The other two shadows, one on the floor and one on the right wall, show how busy individual people are. However, “Time Lattice” requires a large space to project information on to the wall and ground. It does not handle privacy issues well, because the 3D model shows too many details.

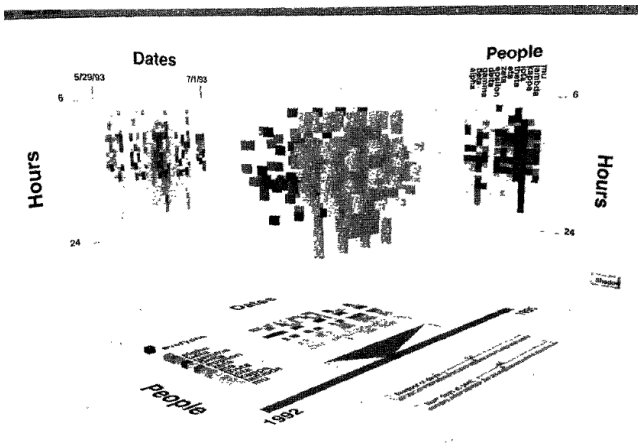



Figure 2.6: The Time Lattice visualizer, which is designed for analyzing the time relationships among the schedules of groups of people. A 3D object compares the weekly calendars of a group of people. Translucent shadows are used to support understanding and interaction with this complex 3D object.

In our latest work, the ActivityStream desktop tool provides an interactive visual calendar like “Spiral Calendar”, but also supports multiple users’ calendar data. Additionally, the ActivityStream desktop tool provides aggregate information with simple curves, and has a login system to protect users’ privacy.

## 2.2. Public Display

As an effective way to broadcast information, the public display has been widely adopted in schools, offices, streets, airports and many other places. People use blackboards to broadcast notices to students at school, use billboards to attract customers in the street, and hang bulletin boards at the top of doors to leave messages at home. With the explosive growth of digital displays and computer technology, many public displays now use large LCDs to display rich information. For example, a large LCD in the street might show local temperature, a road notice, or a football match, the content depending on the program. The large display works as an ambient display in a public place, and provides an opportunity to exchange information within individuals as they watch it [9].




Figure 2.7: Online Enlightenment. The appearance reflects the layout of the lab. Caricatures spaces on position provide three ways to identify users: image, name and location.

Early studies try to use physical objects in a public display to provide specific information for group members. In [10], the author designs a physical public display to show workers' presence. Figure 2.7 shows the physical display. The device uses a map

metaphor to represent a set of connected labs, and shows instant message (IM) online states for a community of users. It is intended to strengthen connections within the community, to act as a bridge to other technologies, and to encourage the use of IM. This display provides ambient presence information to users so that login to computers is not necessary. However, a change in real-world layout will require modifications to the physical map in the device. This device also only works for small groups with limited users, and abstract presentation of physical design may cause confusion to visitors.




Figure 2.8: The Notification Collage. Media elements are annotated. New items are positioned only left of the vertical bar.

Some prior works focus on using digital displays to provide detailed information about group activity. In [11] the author creates a system called *Notification Collage* (NC), where people post media elements, and other people become aware of them and selectively reacted to them. NC becomes a rich resource for awareness and collaboration by spanning several types of tool: awareness notifies, instant messengers, media spaces and asynchronous bulletin boards. However, the media elements posted on the college raise privacy issues; not everyone represented in the media elements may feel

comfortable. Researchers have used different ways to handle privacy issues in public displays.



Figure 2.9: Panarama represents an "Idle" environment

In [12], the author introduces an intelligent, large screen display called *Panorama* that is aimed at supporting and enhancing social awareness within an academic work environment. *Panorama* provides rich interpersonal information related to co-workers and their everyday interactions in the department. Privacy issues may arise when monitoring people in real-time. Panorama avoids these issues by presenting videos in abstract forms using shades and particle overlay effects.

In [5], the author designs a semi-public display for a small co-located group, providing lightweight information about group activity to minimize privacy concerns. The semi-public display uses weekly reports and big events to identify relevant types of information, which is collected through email, calendar and word-of-mouth. The lightweight information maintains low-level group awareness. These prior works mentioned above, though, do not solve privacy issues completely. In our work, ActivityStream desktop tool solves the problem by using a logon system and ActivityStream public display solves the problem by providing only high-level, aggregate information about group activity.

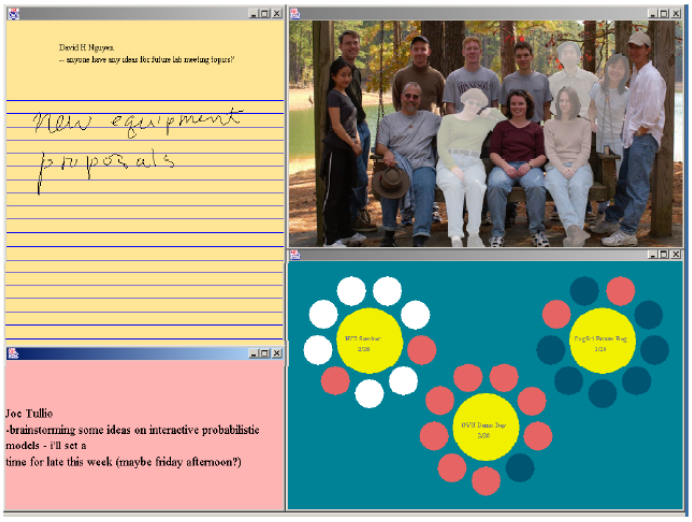



Figure 2.10: The Semi-Public Display prototype. Clockwise from top left: a) Collaboration Space; b) Active Portrait; c) Attendance Panel; d) Reminders

In [13], the author defines four interaction phases in the public display: ambient display, implicit interaction, subtle interaction and personal interaction. He demonstrates that fluid movement between different interaction phases is possible. He uses technologies like touch screens, and sensors to create different interactions. The way people interacting with the display will generate different information and awareness levels. Both task visualization public display and ActivityStream public display have two interaction phases: ambient display phase and implicit interaction phase. Users are able to get a sense of overall information from both displays with a quick glance, and receive deeper information when they are closer to the displays.

## 2.3. Group Awareness and Collaboration

The concept of awareness has come to play a significant role in CSCW (Computer Supported Cooperative Work) at a very early stage. With explosive growth of digital media, people have attempted to explore the potential of “media space” technologies to facilitate awareness in a group [14]. The definition of awareness by the author of [2] is an “understanding of the activities of others, which provides a context for your own activity”.

The author thinks that the shared feedback system conveys a continuous sense of individual actions and group progress with less information-production costs to individuals. Group awareness is the understanding of who is working with you, what they are doing, and how your own actions interact with theirs [2]. Group awareness reduces the effort needed to coordinate tasks and resources by providing a context in which to interpret statements and to anticipate others' actions [15].

In [16], the author suggests three mechanisms to maintain awareness in a group: explicit communication, where people talk about their activities directly or through email and chat tools; consequential communication, in which watching another person's work provides information about their activities and plans; and feedthrough, where observation of changes to a project indicates who is doing what. These mechanisms maintain low-level group awareness by collecting details about other people's activities. There are two ways to increase the level of group awareness. One way is as the author suggests in [17], the group awareness starts low and increases over time. The other way is to provide aggregate information of group activity. In our work, AcitivityStream desktop tool and ActivityStream public display provide aggregate information of group activity to maintain high-level awareness.

Prior works focus on using physical display and information visualization to improve group awareness. In [10] the author designs a physical device to show people's presence in a lab by mapping labs' position and showing instant message (IM) online states for a community of users. However, the physical system is not reliable for larger groups because changes to the user group require reconstruction of the physical device.




Figure 2.11: Newsgroup Crowd visualization of alt.politics.bush; authors are distributed all over the scatter plot attesting to the diversity of contributors and the conversational character of the newsgroup.

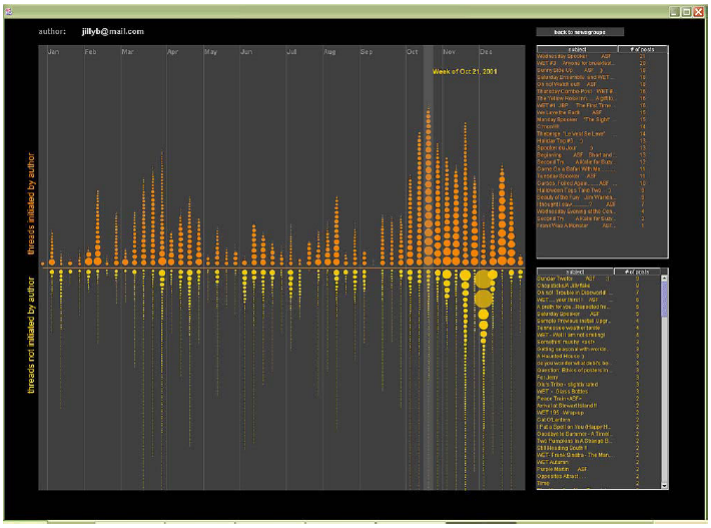



Figure 2.12: AuthorLines visualization of an author that starts about as many thread as the ones they respond to. This author has been active for partically every single week of the year with the most amount of activity happening towards the end of the year.

In [18], the author creates two visualizations: *Newsgroup Crowds* and *AuthorLines*, to visualize activities in conversational cyberspaces. *Newsgroup Crowds* graphically represents the population of authors in a particular newsgroup. *AuthorLines* visualizes a particular author's posting activity across all newsgroups over a period of one year.

*Newsgroup Crowds* reveals that authors in the same group concentrate on the same topic or are all very active in a certain time, making authors more aware of other authors in the same group. However, neither visualization allows users to get to the actual content of the messages.

In [12], the author presents an intelligent, large screen display called *Panorama* that is aimed at supporting and enhancing social awareness within an academic work environment. Panorama establishes two levels of communication amongst the co-workers: information and impression. Information means concrete information such as individual achievements. Impression means presenting different sets of information in different ways to indicate activity level. From this perspective, information increases low-level group awareness while impression increases high-level group awareness. Panorama collects information from people with sensors, which may cause privacy concerns. Compared to these prior works, ActivityStream desktop tool avoids privacy issues by implementing logon system, and provides detailed task information and aggregate information with different graphic objects. ActivityStream public display avoids privacy issues by providing high-level information only, and uses the same idea in Panorama: representing information in certain ways to indicate activity level and then conveying impression.

An increase of group awareness will promote collaboration in some situations. In [19], the author raises the point that information visualization improves group collaboration, but it depends on whether visualizations promoted communication. The visualization does not improve collaboration without communication or discussion. Aggregate information about activity – a change of workload in ActivityStream public display will encourage users to talk and probably start collaboration. In [20] and [21], the authors mention that activity awareness, situational awareness, and being aware of what

happened in the environment are all critical to increasing group collaboration. ActivityStream desktop tool provides detailed information and aggregate information of users' activity, which increases situation awareness and gives every member a sense of what is occurring in the group. In addition, the interaction design in the system eases group collaboration, allows users to email their co-workers by one click.

## 2.4. Reflective Design

HCI (Human Computer Interaction) opens a new space to improve technologies. Reflection, defined in [22], is one technology design goal in HCI. Reflection can bring unconscious aspects of experience to conscious awareness, and thereby make them available for conscious choice. Unlike an HCI system, which always conveys clear, specific information to users, reflective design conveys an impression or emotion, which has multiple interpretations. In [23] the author presents three interpretations for reflective design. At the lowest level of interpretation, users need to interpret a system's interface and actions; at middle level of interpretation, users intend to connect the system to their everyday lives; at the highest levels of interpretation, user interpret the embodied value and social and cultural meaning of systems.

Even though HCI systems and reflective design systems have different goals, and HCI systems are functional while reflective design are usually like an artwork, many researchers have tried to bridge the worldviews of art and HCI. In [24] the author presents a system called Birdscape with the goal of hybridizing art and tool, for an art museum. As a HCI tool, it provides patrons information about the museum and their interactions with it. As an artwork, it creates new experiences in an open-ended way. The evaluation system proves that Birdscape program is more like art than tool.

In [25] the author suggests using different evaluation systems since HCI and reflective design had different design purposes. The design purpose for HCI system should focus on

how to interpret the machine and how to communicate a message clearly. The goal for artwork should focus on how an emotion is expressed from the machine, how artists use it to convey their ideas, and this emotion leads to the desired effect on the interaction. This evaluation system will check whether this system is able to provoke such emotion so that users can better reflect on their experience [26][27]. In [28], the author demonstrates that the interpretations differ depending on users' background and real life experience. ActivityStream public display examines whether the system conveys a stressful emotion about group activity to users, and whether this emotion leads to correct interpretation.

# CHAPTER 3

# ITERATIVE SYSTEMS DESIGN

This chapter introduces the approach and design methodology that were used in order to design the calendar visualization systems. Chapter 4 will discuss the user studies and analysis for these systems thoroughly.

## 3.1. Iterative Design Methodology

Our research goal is to design a calendar visualization tool that increases group awareness and promotes reflection. In order to achieve this goal, we used an iterative design approach with each iteration of the design improving upon the previous one. Specifically, iterative design is a design methodology based on a cyclic process of prototyping, testing, analyzing, and refining a product or process [29]. Changes and refinements are made based on the results of testing the most recent iteration of a design. This process is intended to improve the quality and functionality of a design. The typical steps of the iterative design process are as follows:

1) Complete an initial interface design
2) Present the design to several test users
3) Note any problems which were noted by the test users
4) Refine interface to account for and fix the addressed problems
5) Repeat steps 2-4 until all interface problems are satisfactorily resolved

In earlier work, we present a task visualization public display to provide detailed task information for a small group of people. The motivation in doing so is to provide up-to-date information about group activity to every other member of the group. User feedback reveals that users wanted more interaction and aggregate information about group activity. Additionally, users want privacy issues to be addressed as well as being able view data from other sources such as Google Calendar. Our latest work that we named

ActivityStream attempts to solve these problems. The motivation behind ActivityStream is to design an interactive system to provide detailed, aggregate information about group activity while maintaining sufficient privacy. The ActivityStream desktop tool has a login system that helps protect privacy, uses curves to provide aggregate information about group activity, and enlarges the number of data sources by collecting data from Google calendar, iCal and Outlook. We design a new system with many interactive interfaces. User feedback to this iteration is more positive than in the previous iteration. The feedback that we receive shows that users want an easier way to compare their calendar data with the data from others, and they want the system to be reliable for large groups. To further refine our system, we design a third system that is known as the ActivityStream public display. The main change made for this system is to provide high-level information about group activity to large groups. This system further protects privacy by providing high-level information only and shows multiple curves that are a visual comparison of the workload of every user. Users receive high-level information about group activity by being able to feel a stressful emotion due to the density of the workload curves. We have proven that the system is reliable when group size increases.

## 3.2. Earlier Work- Task Visualization Public Display

### 3.2.1. *Motivation and Design Principle*

The motivation behind our work is to provide up-to-date task information for a small group of people, show patterns of group activity, and give lightweight, high-level information about group activity. The pattern represents summary information of group activity during a period. The pattern implies the group workload. We design the system as a public display in order to make it easily accessible to everyone in the group. In order to show detailed information, the system uses discrete objects to visualize every single task that we collected from the data. To show both the task data at the group level and

task data at the individual level, we run an animation in the display that goes through a group's task information as well as each individual's task information. This system shows four months of data with one time scale (weekly) to make the visualization clear to users.

### 3.2.2. *System Architecture and Data Collection*

We define a task is a piece of work assigned or done as part of one's duties. The task visualization public display collects task data from Remember the Milk (RTM), which is a web-based to-do list manager with an emphasis on simplicity and integration with popular third-party applications [30][31]. It has a comprehensive set of features which include a nice Ajax interface), text-message and e-mail reminders of due dates and mobile access for any web-enabled cell phone. People are able to get access to RTM through Gmail, Twitter, iPhone applications, or Android applications. The system we developed collects data from RTM because RTM includes many task based features such as tag, priority and completion information. Tag specifies the task category information, priority specifies the task priority, and completion information shows whether tasks were completed.

We build a client-server architecture for the system, which is shown in Figure 3.1. The Java program running on the server collects RTM data by parsing RSS feeds. In RTM, every to-do list has a published RSS feed. After parsing the RSS feed, the system stores task objects into a MySQL database. We define a task object as an object with the following task properties: title, id, time, username, category, priority and completion information. We setup a cron job on the server side to run the parser every 10 minutes. Cron [32] is a time-based job scheduler in Unix-like computer operation system. On the client side, the public display is running an animation loop. It retrieves new data from the database each time a new animation loop starts.

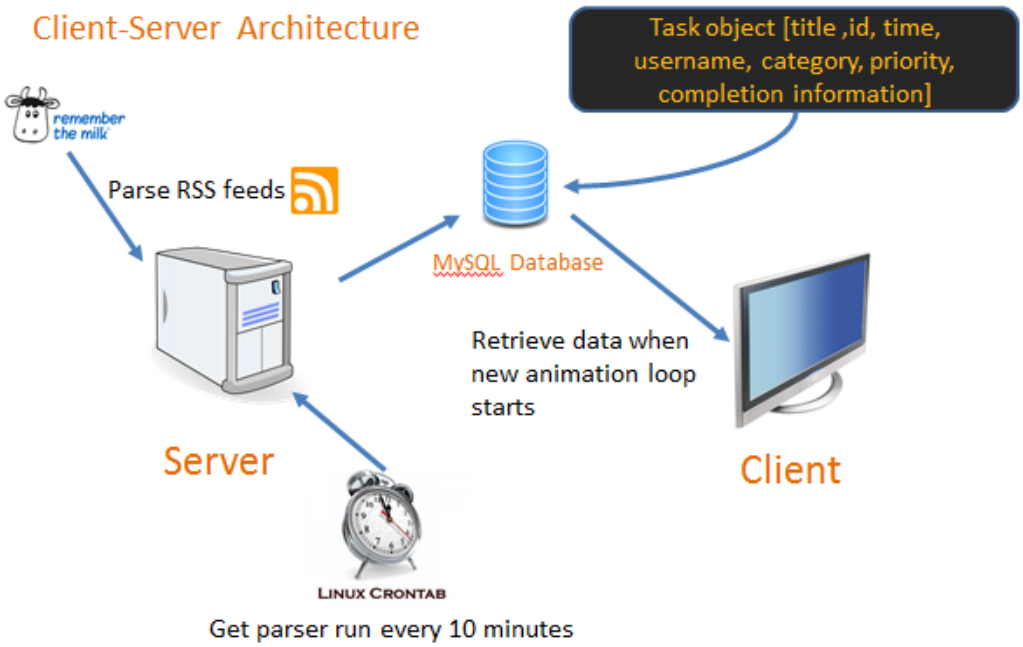



Figure 3.1: Client-server architecture. The server collects data from RTM by parsing RSS feeds every 10 minutes and stores the data into the database. The client retrieves data every time a new animation loop starts.

Figure 3.2 demonstrates how the RSS parser constructs a task object from an RSS feed node. The left block is a standard RSS feed node. It includes several tags: <title>, <link>, <guid>, <dc:publisher>, <dc:creator> and <description>. <title> has the task title information. <link> has specific task id information. <dc:creator> has information about the username of the task. <description> has further information including task due time, task category, task priority and completion information. The right block is a standard task object defined in a Java program. The program generates a new task object by parsing a RSS feed node. Unfortunately, many RTM users do not set the category and priority information when they create a task in RTM. Our system gives a default value for them that is “personal category” for the category field and “medium” for the priority field.

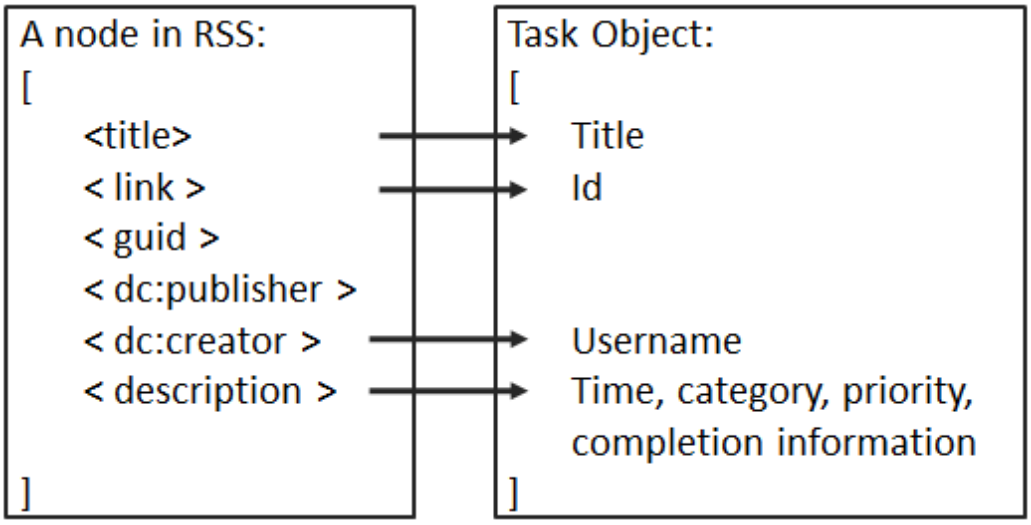



Figure 3.2: RSS feed parser. Left block shows a standard RSS feed node structure. Right block shows a standard task object structure. Arrows mean the task information is parsed from the tag information in RSS feed.

### 3.2.3. *Approach and Design*

This task visualization public display shows weekly task information of the Reflective Living research group in ASU. The timeframe of the visualization is from January 15 to May 15, 2010. The public display has two modes: a group view mode and an individual view mode. Figure 3.3 shows the group view of the task visualization public display, and Figure 3.4 shows the individual view of the task visualization public display. The group view mode shows all tasks in the group, and the individual view mode only shows a single user's task information. The display continuously cycles through the group view and each individual view. A moving bar in the visualization controls the animation time and shows what date it is currently iterating through. Discrete blocks represent a task, and the horizontal timeline and vertical category define the position of every task block. The task blocks are stacked in order of task priority. The size of the block reflects task priority, and the color identifies the who the task belongs to. Unfilled blocks represent incomplete tasks and filled blocks represent complete tasks. Information bubbles randomly appear for some task blocks that are touched by the moving bar. When the moving bar reaches

the far right side of the display, it transfers to either the group view mode or another individual view mode. The goal of the system is to provide detailed task information to the users which includes content, task priority, task category, the user of task, and temporal task data. Therefore, we design the block color, block size, layout, user thumbnails, information bubbles and animation in our system in such a way that enables the user to quickly see this information. We will address these design decisions in detail below.

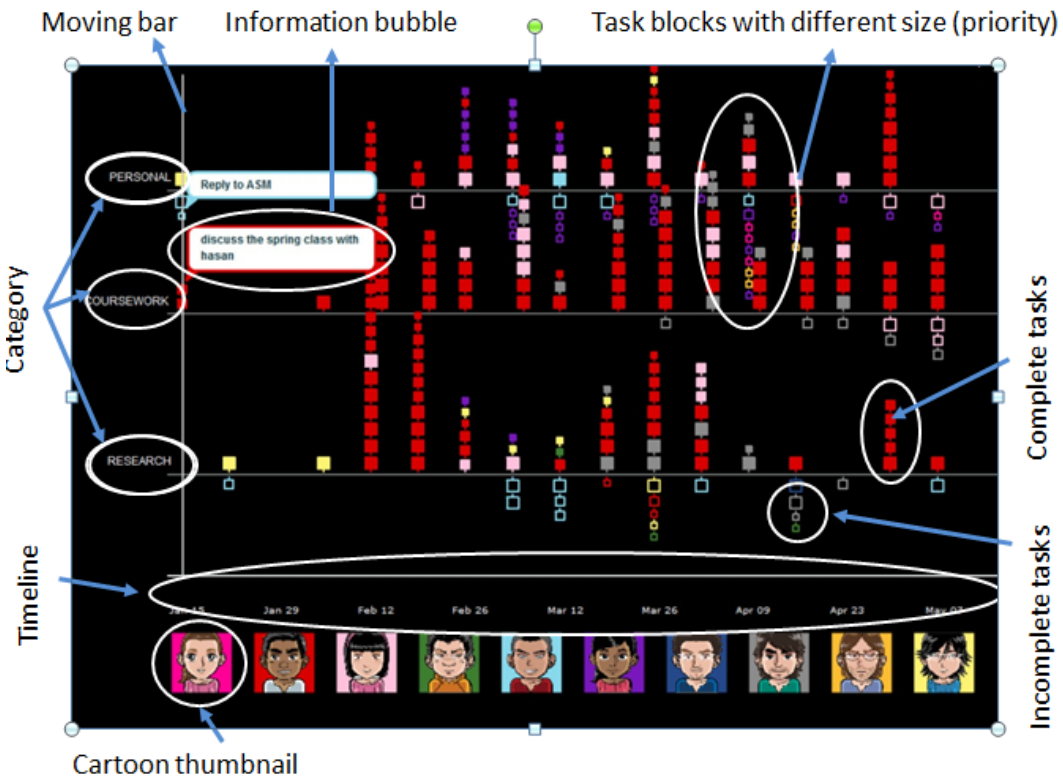



Figure 3.3: Group view of the Task Visualization Public Display. This screenshot shows all tasks in the group. Horizontal axis is the timeline. Vertical axis shows category information. Every single block represents a task. The group view transfers to the individual view when the moving bar moves to the right side.

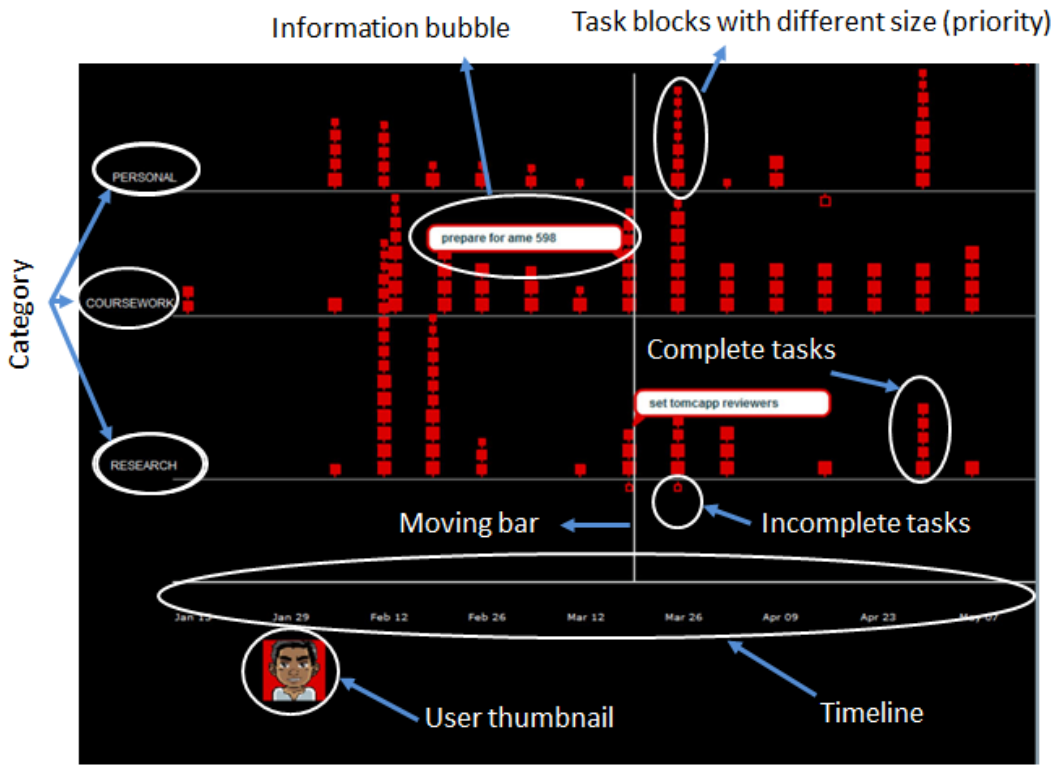



Figure 3.4: Individual view of the Task Visualization Public Display, this screenshot shows individual's tasks. Horizontal axis is the timeline. Vertical axis shows category information. Every single block represents a task. The individual view transfers to another user's individual view when the moving bar moves to the right side.

**Color Design.** Figure 3.5 shows task blocks with different colors. The system assigns a specific color to each user that is used as an identifier in the visualization. This is our color palette: {0xFE0096, 0xDA0000, 0xFFC1DC, 0x3E7E30, 0x88D9EC. 0x7918BD, 0x26498B, 0x8C8C8C, 0xFFBE34, 0xFFF373}.

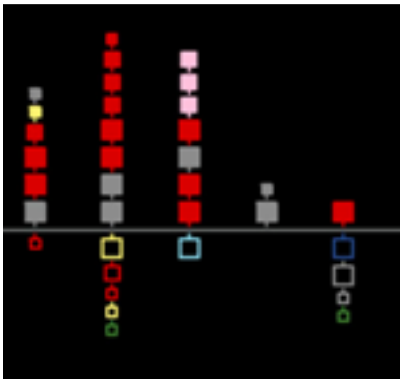


Figure 3.5: Task blocks with different colors with blocks stacked in order of size

**Size Design.** Priority defines the importance of a task. The purpose of having task priorities is to help the user organize their large number of tasks and enables them to pay attention to important tasks in order to improve productivity. Figure 3.5 shows blocks of different sizes with the size of the block representing the priority of the task. We do this because it is intuitive for the user to see that the larger task blocks are more important than the smaller ones. The big, medium, and small blocks correspond to three levels of task priorities: high, medium, and low. The side length of a block is defined by the formula shown in 3.1, where $p_i$ is task priority and $p_i$=1 indicates high priority while $p_i$=3 indicates low priority. The block side length is larger when the priority is higher.

$$Length_i = 6 \times (2.5 - p_i \times 0.5) \quad (3.1)$$

The system uses three priority levels because Remember the Milk supports three levels of priority. When the user does not set a priority value for a task, our system sets it as medium by default. Figure 3.5 also shows stacked blocks that are ordered by their size. For blocks above the central line, bigger blocks are on the bottom and smaller blocks are on top. For blocks below the central line, bigger blocks are on top and smaller blocks are on the bottom. This particular ordering is used for aesthetic purposes.

**Layout Design.** In Figure 3.3, the horizontal axis is a timeline, which covers one semester's worth of data from January 15 to May 15, 2010. The timeline only has single time scale that shows weekly task data. The timeline makes it easy to observe how the group changes over time. We only show four months data because we wished to see the task information of this group over one semester first, and afterwards, future work will improve upon the timeline. Since it is a non-interactive public display, a timeline with only a single time scale is simple and easy to understand.

The vertical axis shows category information. Category information is meaningful because users are composed of faculty and students who mainly use three categories for

their tasks: personal, education and research. Figure 3.3 shows the three categories arranged vertically. Categorizing tasks makes it easier for users to browse tasks and compare them across different categories. The horizontal axis shows how tasks change over time, and the vertical axis shows how users divide their time among different types of activities. For example, by comparing tasks horizontally, people understand that others have more research tasks in April than in other months, because there are more paper deadlines in April. By comparing tasks vertically, people can easily see that a user has about 60% research tasks, 30% coursework tasks and 10% personal tasks in a week, which implies the user spends more energy on research than anything else. When the user does not set the category information for a task, we assume that the user does not want to share the category information. Therefore, the system sets “personal” as the default category for that task. The category name could be changed easily to fit the requirements of different groups. For example, a research group typically uses the categories: personal, coursework and research while a software development company may use programming, design and documentation.

**User Thumbnail Design.** Figure 3.6 shows cartoon thumbnails for all users that are shown at the bottom of the display. Figure 3.3 shows that every thumbnail has a background color, which corresponds to the color of his or her task blocks. That helps users locate their tasks easily and compare tasks between different users. We designe these cartoon thumbnails with an online tool called “Face Your Manga” [33]. We use cartoon faces rather than real photos because it injects a fun and uniform aesthetic into the visualization. It also protects privacy as it is difficult for outsiders to identify which RL member corresponds to which cartoon face.



Figure 3.6: Cartoon thumbnails for all users



Figure 3.7: Three facial expressions, from left to right: calm, happy and stressed

For each cartoon thumbnail, we designed three faces that represent workload level. The workload level is aggregated from an individual's tasks in one week. The workload depends on the quantity and priority of tasks which is defined in formula 3.2, is the number of tasks in one week, is priority of task , and is the workload in one week.

(3.2)

Figure 3.7 shows three facial expressions that correspond to three workload levels. A happy face indicates a small workload, a calm face indicates a medium workload, and a stressed face indicates a large workload. Based on our observation to calendar data, the workload value is distributed in three intervals: [0, 2], [3, 5], [6, ), which corresponds to a low, medium and high level workload. If the workload is in this interval [0, 2], the face is happy; if the workload is in interval [3, 5], the face is calm; and if the workload is in interval [6, ], the face is stressed. These faces provide lightweight, aggregate information about the activity for every user.

**Information Bubble Design.** Information bubbles show tasks details and provide up-to-date information. Usually when a group of people works on the same project, they need working progress information from other peers. Asking everyone for that information consumes a large amount of time. These information bubbles provide that information. Even outside of shared projects, users in a group care about their peers'

work. The information about the work of other peers creates possible collaboration and a complement of skills and knowledge in a group. The color of the information bubble's frame is consistent with the user's color and the height of bubble depends on the length of the content. The largest bubble displays three lines of text with 84 letters.

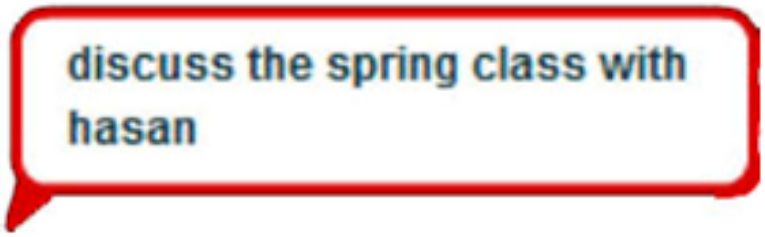



Figure 3.8: Information bubble shows task content. It supports three lines of text.

Two information bubbles pop up randomly when the moving bar touches some task blocks. The moving bar has been explained in section 3.3.3. To avoid the possible overlap of the two information bubbles, we display one bubble on the left side of the moving bar and display the other bubble on the right side of the moving bar (see Figure 3.4).

**Animation Design.** The public display has two modes: group view mode and individual view mode. Figure 3.3 is group view of task visualization public display, and Figure 3.4 shows individual view of task visualization public display. Group view mode shows all tasks in the group and individual view mode only show single user's task information. The display is running a cycle through group view and each individual view. A moving bar moving from left to right in the visualization controls the animation time. When the moving bar moves to the far right side, the display switches to a group view mode or another user's individual view mode. The speed of the moving bar is adaptable, depending on the quantity of tasks in that view. The system sets a timer for the moving bar which moves it one pixel every five milliseconds. The timer is defined in formula 3.3.

where $n$ is the number of tasks during the entire timeframe of four months. The adaptable speed of the moving bar saves people's time when they watch someone's data that is very small, and they want to skip to next view quickly.

$$Timer = \begin{cases} 5\,, & n < 5 \\ 10, & n \geq 5, n < 10 \\ 60, & n \geq 10, n < 100 \\ 100, & n \geq 100 \end{cases} \tag{3.3}$$

3.2.4. *System Weaknesses*

- The task data that is collected from RTM is not complete. A survey that we conduct shows that many users use Google calendar, iCal and Outlook to organize their tasks. The system should collect calendar data from more online calendar tools.
- The Task visualization public display is a non-interactive display. Users want more interaction with the system in order to access information more easily. Users want to click the blocks to examine details, drag the timeline and click the thumbnails to switch to the individual view mode. Therefore, the system should have more interactive features.
- The system does not handle privacy issues well. The only solution is to use thumbnails to anonymize the user's information from people outside of the RL group. However, the information bubbles show too many details that makes users uncomfortable.
- This system only displays four months of data over a single time scale. The system should allow users to go over their past data and future data over longer periods. A visualization showing multiple time scales may be solution to visualize data over long periods.
- The Task visualization public display only provides low-level, detailed information of group activity, which is the same information provided by other calendar tools like

Google calendar. This system should provide high-level, aggregate information of group activity that is missing in other calendar tools.

- This visualization is slightly cluttered because we separate the task data into three categories. The task blocks in different categories may overlap. The new system should address this issue.

## 3.3. Latest Work- ActivityStream Desktop Tool

### 3.3.1. *Motivation and Design Principle*

Based on the weaknesses described in our earlier work, the goal of our latest tool was to create an interactive system to provide both detailed and aggregate information of group activity to a small group of people while addressing privacy concerns. The new system should collect data from more online calendars, protect users' privacy when they share calendar data with the system, and provide clearer multi-timescale visualization. As a system to raise group awareness, the system should show relations between individual activity and group activity. Additionally, we want to add some additional functionality in order to encourage group collaboration.

Regarding the design dimensions in 3.2, this system will use both discrete information and continuous information to display detailed and aggregate information to a small group. This system will be an interactive desktop tool with a static display that visualizes multi-timescale data.

### 3.3.2. *System Architecture and Data Collection*

Based on our experience with the previous work, we collect more calendar data in order to reflect group activity. This system collects data from Google calendar, iCal, Outlook, and Remember the Milk, which are the calendar tools that our users use most frequently according to a survey that we conduct in our earlier work. We also notice that some data in Google calendar are events rather than tasks that are semantically different.

As was stated earlier, we define a task as a piece of work assigned or done as part of one's duties. Tasks have consequences when people are unable to finish them. Tasks are not necessary to have a specific due time. People may schedule to finish a task in a week without a specific time. However, events are different and take place on specific dates. To account for these differences, we simply get both task and event data. Since both tasks and events reflect group activity, the system defines them as activity data. The system deals with them differently in the process, which will be discussed in the data collection section.

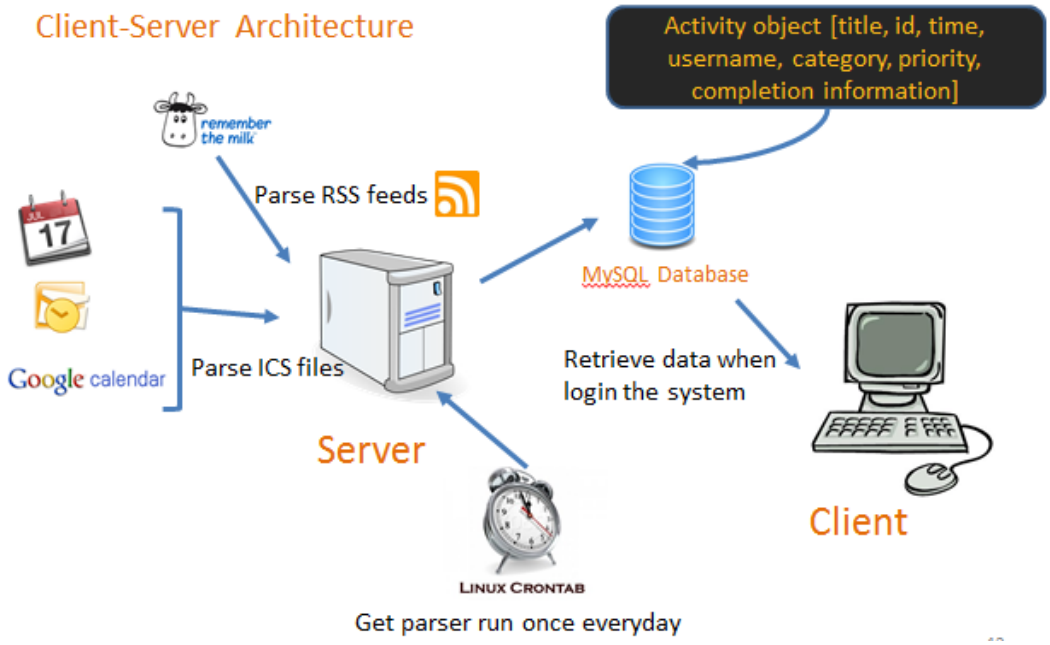



Figure 3.9: Client-server architecture. The server collects data from RTM by parsing RSS feeds, collects data from iCal, Outlook and Google calendar by parsing ICS files every day, then stores the data into a database. The client retrieves the data when a user logs into the system.

For this new iteration, we build a client-server architecture for the system, which is shown in Figure 3.9. The architecture and data collection is very similar to the one described in our earlier work. One major difference is that the Java program in the server not only collects RTM data by parsing RSS feeds but also collects calendar data from

Google calendar, Outlook and iCal by parsing their corresponding ICS files. ICS is a standard format for calendar files. After parsing the RSS feed and ICS files, the system stores activity objects into a MySQL database. An activity object is an object with the following activity properties: title, id, time, username, category, priority and completion information. We setup a cron job on the server side to have the parser run once every day. On the client side, the desktop tool retrieves data from the database when users log into the system.

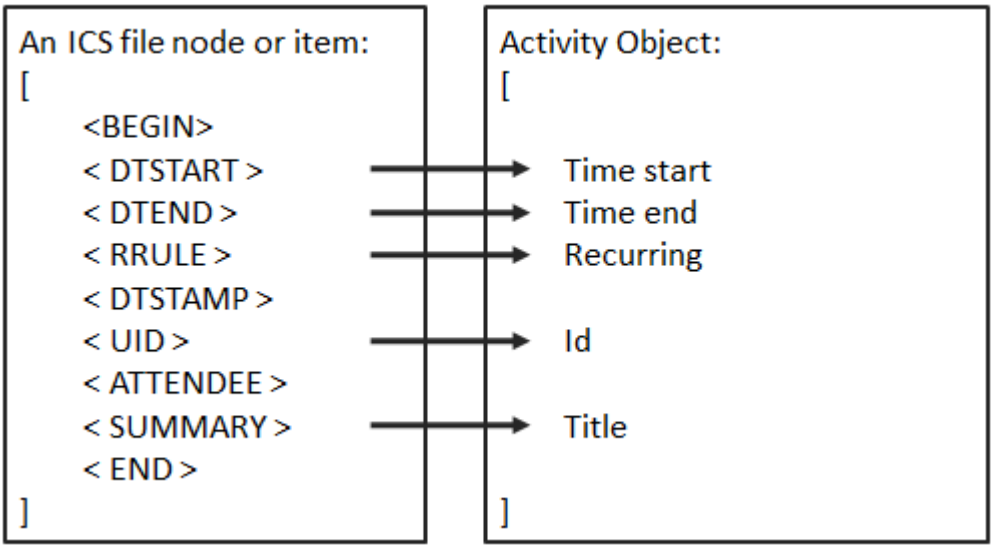



Figure 3.10: ICS file parser. Left block shows a standard RSS feed node structure. Right block shows a standard task object structure. Arrows mean the task information is parsed from the tag information in RSS feed.Approach and Design

Section 3.3.2 has already described the implementation of the RSS parser. Now we will discuss the ICS file parser. Figure 3.10 shows the ICS file parser constructs an activity object from an ICS file node. The left block is a standard ICS file node. It includes several tags: <BEGIN>, <DTSTART>, <DTEND>, <RRULE>, <DTSTAMP>, <UID>, <ATTENDEE>, <SUMMARY>, <END>. <DTSTART> and <DTEND> show the start and end time information. <RRULE> shows frequency information for recurring events such as daily, weekly and monthly. <UID> has information about the activity id. <SUMMARY> includes the content of the activity. The right block is a standard activity

object defined in a Java program. The program generates a new activity object by parsing a node in the ICS file. If the node shows this is to be a recurring event, the program will generate many repeated activity objects depending on the frequency of occurrence of the event and the start and end times of the event. Compared to the RSS feed, the ICS data loses much information that includes the category, priority and completion information of activity. When the system does not collect that information, the default category is "uncategorized", the default priority level is "medium", the default completion information is "true" when the time of occurrence is in the past, and the default completion information is "false" when the occurrence time is today or in the future.

### 3.3.3. *Approach and Design*

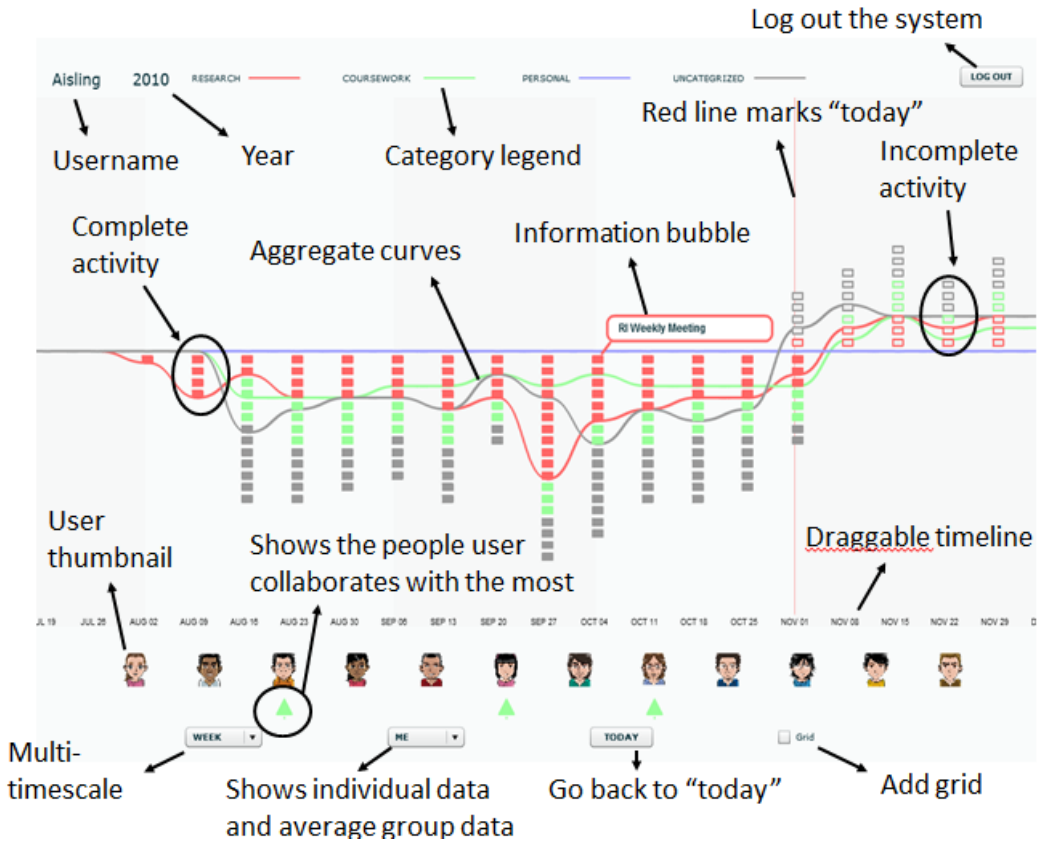



Figure 3.11: View of the ActivityStream desktop tool when users browse their own calendar data. Horizontal axis is a draggable and multi-timescale timeline. This view provides both detailed activity information (brick with information bubble) and aggregate information (aggregate curves).

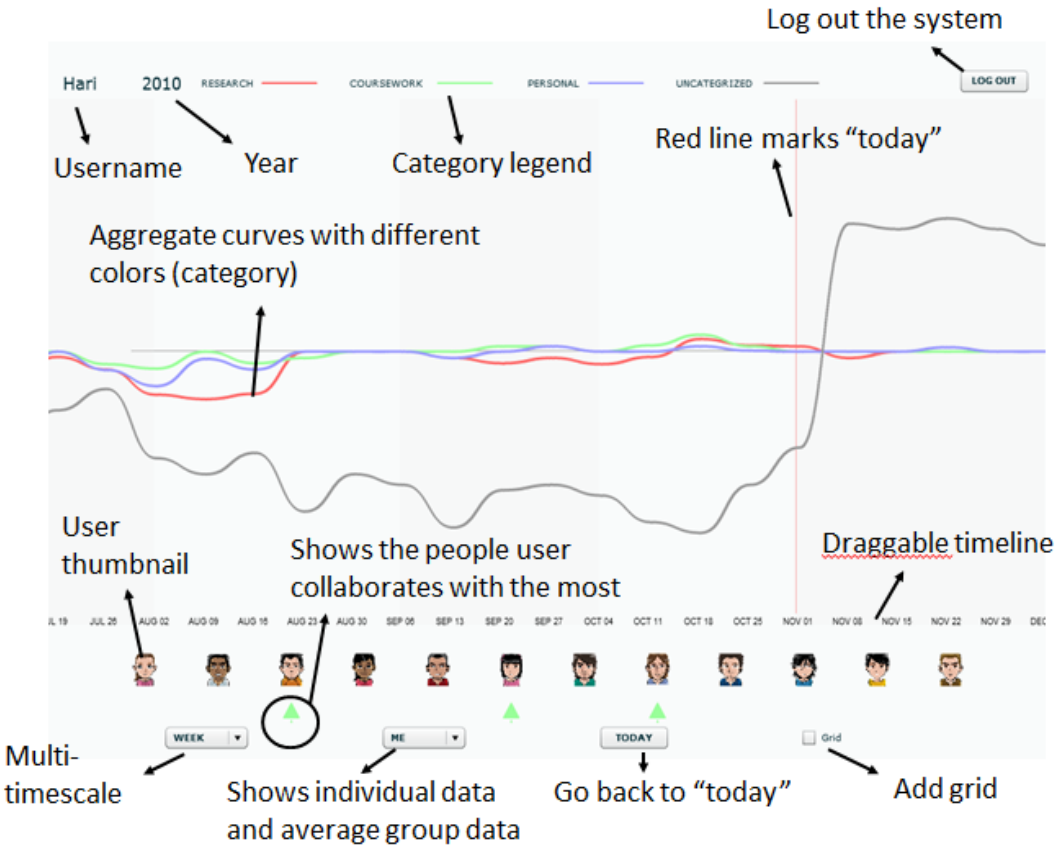



Figure 3.12: View of ActivityStream desktop tool when users browse others' calendar data. Horizontal axis is a draggable and multi-timescale timeline. This view uses curves to display aggregate information of other peers' activity.

The ActivityStream desktop tool is a system that provides both detailed and aggregate information about activity to a small group of people. It allows users to check their own activity details by clicking the blocks and examine their peers' aggregate activity information. ActivityStream has a login page in order to protect every user's privacy. Figure 3.11 shows the visualization when users log in to the system. This screen allows them to browse their calendar data. On the top of the screen, from left to right, are the username, year information, legend and log out box. The legend shows the relationship between color and category. In the main part of the visualization, every block represents a single activity, and information bubbles pop up when users click on the blocks. Filled blocks represent a completed activity while unfilled blocks represent activities that have

not been completed. The system uses four curves to represent aggregate activity information in every category. A red vertical line specifies “today” on the timeline. The bottom of the screen has a timeline along with the users’ thumbnails with arrows underneath as well as three buttons and a checkbox. The arrows indicate which people users collaborate with the most. The functions of the buttons and checkbox from left to right side are: change timescale in the display, compare individual’s activity data and average group activity curve, reset the display and add grids to the display. The draggable timeline with multiple timescales makes it easier to access the calendar information. Figure 3.12 is what is shown when users browse other people’s activity information. This view removes all detailed information and only keeps four aggregate curves to represent the activity information of their peers in each category. We improve upon our earlier work by, designing the login system, block color, block size, user thumbnails, information bubbles, multi-timescale, interaction, aggregate information and group collaboration. We will introduce these design decisions in detail below.

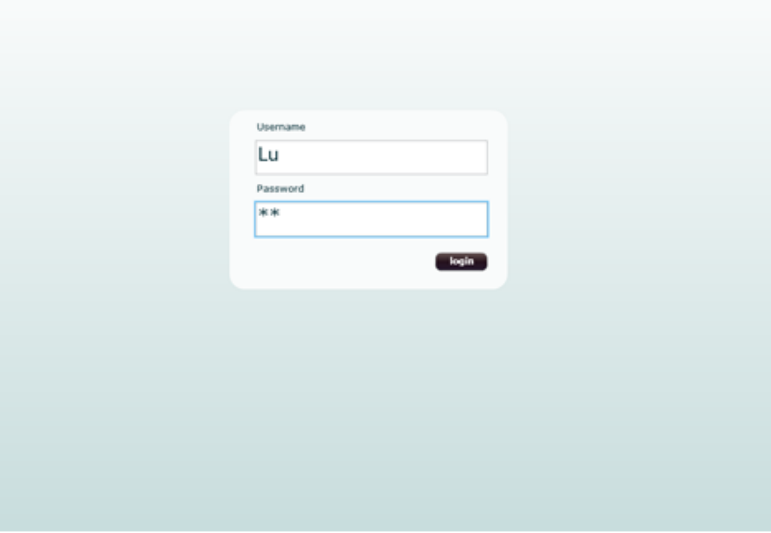



Figure 3.13: Login page. Every user requires a username and password to log in the system

**Login Page Design.** In our earlier work, issues with privacy made users uncomfortable. To address this issue, the system requires every user to login with their own account. Every user has an account that consists of a username and password. The account information is stored in a MySQL database on the server.

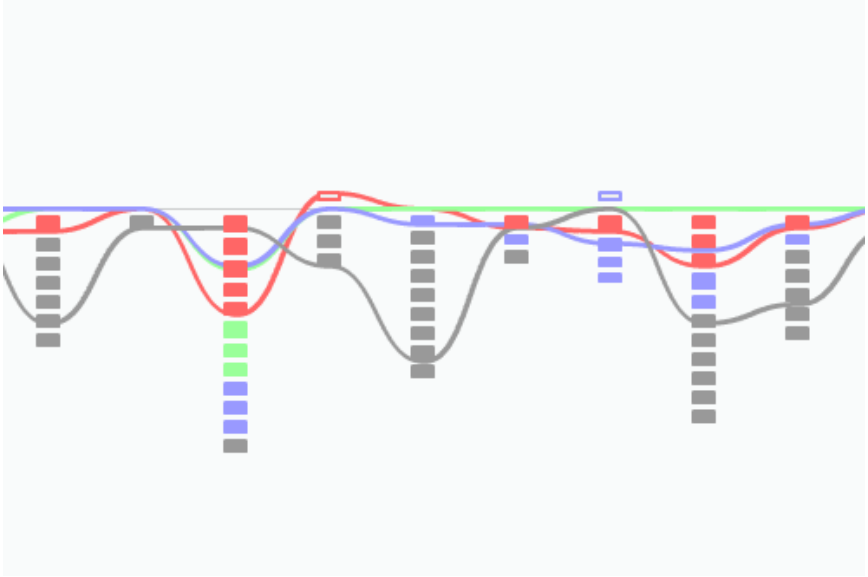


Figure 3.14: Discrete blocks with different colors represent activities. Color indicates the category. Every curve represents aggregate activity information in a category.

**Color Design.** Category information shows what category the activity belongs to and gives users a sense of how they spend energy in different categories. Compared to our earlier system, we make two changes to display category information. First, the earlier system uses three categories to separate task blocks into three columns in the visualization. The stacked blocks in different categories overlap with each other and look cluttered. ActivityStream uses colors to represent category information and uses only one column to show activity data rather than three columns, which makes the visualization cleaner. Second, the earlier system only has three categories: research, coursework and personal. For this iteration, we add an “uncategorized” category in ActivityStream to categorize those activities that do not belong in the other categories. In ActvityStream,

the color red represents research, green represents coursework, blue represents personal and gray represents uncategorized.

**Size Design.** Similar to our earlier system, we use the size of the blocks to represent the activity priority with large blocks indicating high activity, medium blocks indicating medium activity, and small blocks indicating a low amount of activity. However, we tweak the parameters in the formula a little to make the blocks more aesthetically pleasing.

(3.4)

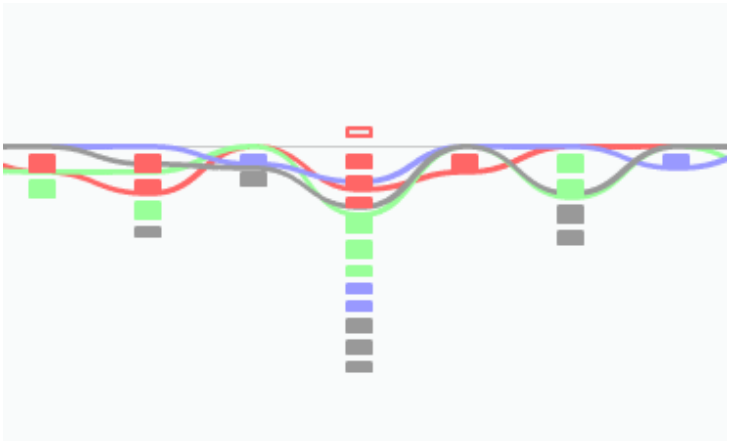


Figure 3.15: Activity blocks are stacked in an order of size. For blocks above the central line, bigger blocks are stacked on the bottom. For blocks below the central line, bigger blocks are stacked on the top.

Similarly, the system sets three priority levels because Remember the Milk supports three levels of priority. When user does not set priority value for an activity, our system set medium as default priority level. Figure 3.15 also shows stacked blocks are ordered by their size. For blocks above the central line, bigger blocks are on the bottom and smaller blocks are on the top. For blocks below the central line, bigger blocks are on the top and smaller blocks are on the bottom. Filled blocks are complete activities while unfilled blocks are incomplete activities. The difference is that in the earlier system, complete task blocks are stacked above the central line and in ActivityStream, complete

activity blocks are stacked below the central line. We reverse this based on feedback from users. It is stated that it is more reasonable to place the incomplete activity blocks above the central line.

**User Thumbnails Design.** We keep the user thumbnails in ActivityStream as a way to identify the user. We make three changes to the thumbnails. First, we remove the three different faces for each user and only keep the calm face as the user's thumbnail because we are already using curves to show aggregate information. Therefore, it is not necessary to use different facial expressions in order to show aggregate workload. Second, these thumbnails are interactive in the current system. Users are able to examine their peers' aggregate activity information by clicking on their thumbnails. Third, the thumbnails in ActivityStream are smaller which provides less of a distraction to users viewing the display.

**Information Bubble Design.** We keep the information bubble design shown in Figure 3.8 based on positive feedback from the earlier system. Now users are able to watch these bubbles by clicking on activity blocks.

**Multi-timescale Design.** In our earlier work, the public display visualizes data over a small time period with one time scale. ActivityStream uses a multi-timescale visualization because it is a better solution to visualize long periods of data and shows different patterns of data to users. ActivityStream has three timescales: daily, weekly, and monthly. Figure 3.16, Figure 3.17 and Figure 3.18 show these different timescales.

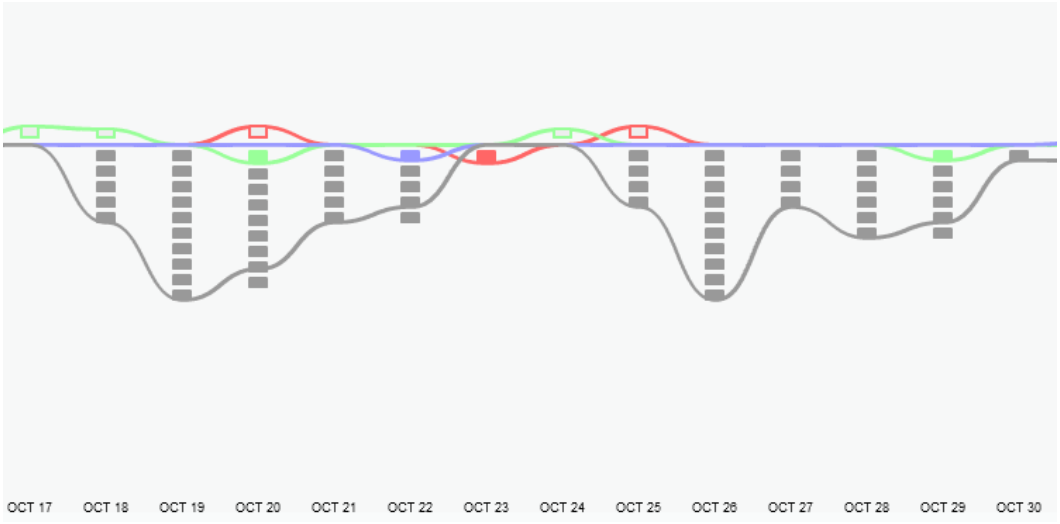



Figure 3.16: Daily view of activity data, all the activity blocks in one day are stacked. This view shows the activity data from Oct. 17 to Oct.30.

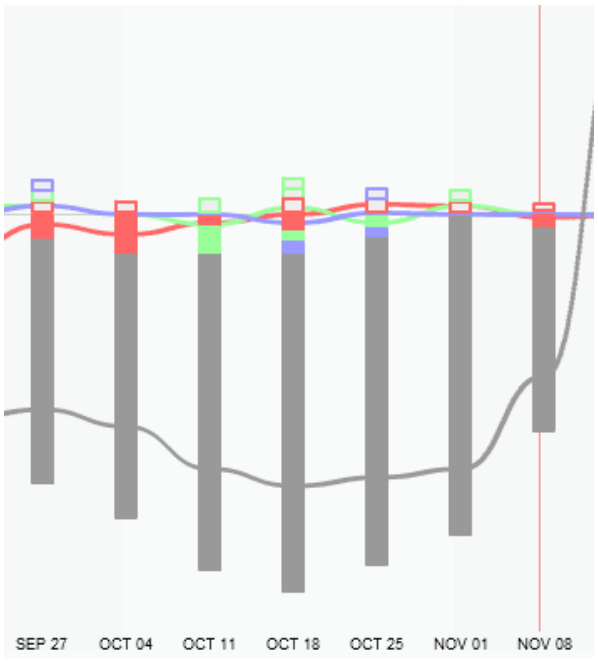



Figure 3.17: Weekly view of activity data, all the activity blocks in one week are stacked. This view shows the activity data from Sep.27 to Nov.08.

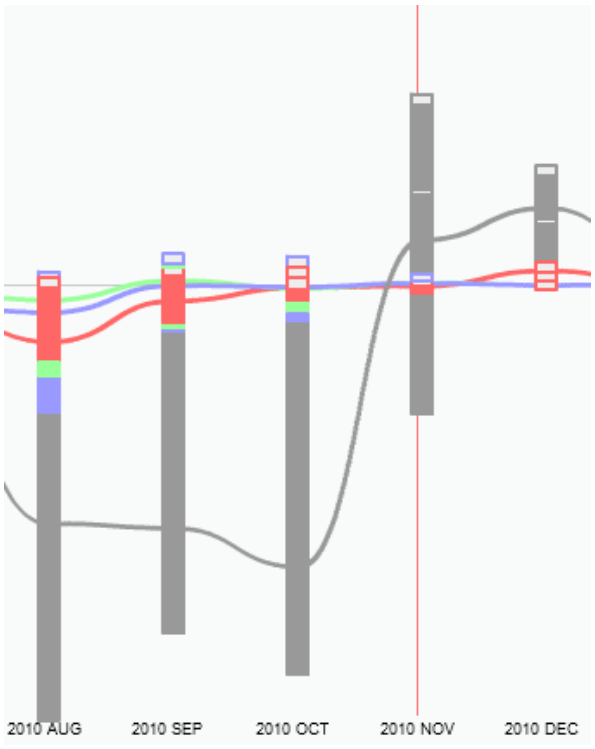



Figure 3.18: Monthly view of activity data, all the activity blocks in one month are stacked. This view shows the activity data from 2010 Aug. to 2010 Dec.

The activity blocks in Figure 3.17 and Figure 3.18 look cluttered because the system normalizes the height of the bricks when the height of the stacked blocks is over half the height of the visualization window, which is 300 pixels. Formula 3.5 defines the $normalizer$ where $N$ is the number of activities in that week or month and $h_i$ is height of activity, $i$.

$$normalizer = \sum_N h_i / 300 \tag{3.5}$$

In formula 3.6, the y value and height of every block are modified based on the $normalizer$. $y_i'$ and $h_i'$ are the modified y value and height value of activity, $i$.

$$\begin{cases} y_i' = y_i \times normalizer \\ h_i' = h_i \times normalizer \end{cases} \tag{3.6}$$

Figure 3.16 shows an obvious pattern in October. The curve section between Oct. 17-23 is similar to the curve section between Oct. 24-30, because recurring activities occur every week. Figure 3.17 shows the number of activities in a week, and the week starting

from Oct. 18 is the busiest week between Sep. 27 to Nov. 8. Figure 3.18 shows the total number of activities in every month and the activities in December shrink to 30% of the average number of activities. As the examples above show, presenting different timescales allows the user to develop insights about the activity of the group.

**Interaction Design.** Our earlier work is non-interactive. Based on user feedback, we add four interactive interfaces in ActivityStream to make it easier to access the information. First, every activity block is clickable. Users are able to get additional information by clicking blocks. Figure 3.19 shows an example of a user clicking on a block. Second, the timeline is draggable. It is convenient for users to go back to past data or examine the future data by dragging and dropping to the point that they want to view. This feature is particularly useful for viewing data over long periods of time. Third, adding a grid to the visualization makes it easier to point the time to blocks. Fourth, curves are interactive. Users are able to email their peers or co-workers by clicking to the curves. The purpose of this feature is to ease group collaboration.

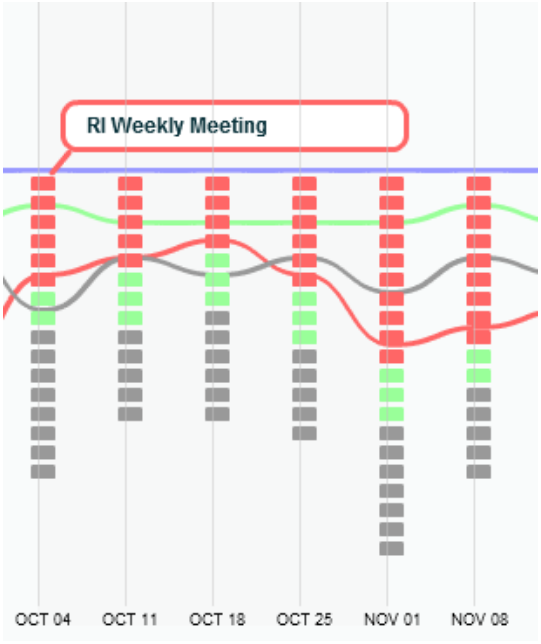



Figure 3.19: Interaction in activity visualization-clicking on a brick to view detailed information

**Aggregate Information Design.** In our earlier work, the task visualization public display only provides detailed information, which promotes reflection. ActivityStream provides aggregate activity information, which is a feature missing in Google calendar, iCal and Outlook. Aggregate information not only gives a high-level awareness about activity but also protects other peers' privacy when they share their calendar data within the group. ActivityStream uses four curves to show aggregate activity information because it is easier for users to compare their workload in different categories. The color of the curve specifies the category. Every curve represents the activity information in one category. The timeline axis defines the x value of the curve. When x is fixed, the user has $n$ incomplete activities and $m$ complete activities. Formula 3.7 defines the y value of curve:

$$y = \left(\sum_n (2.5 - p_i * 0.5) - \sum_m \left(2.5 - p_j * 0.5\right)\right) * normalizer \qquad (3.7)$$

$p_i$ and $p_j$ are the priority value of activity, and normalizer is a constant value. The $y$ value is dependent upon the priority value and number of activities. If a user has many incomplete activities with high priority, the y value of the curve is large demonstrated on the right side of Figure 3.20. If a user has completed many activities with high priority, the y value of the curve is demonstrated on the left side of Figure 3.20. The curve gradually changes as the user continues to complete activities. These curves promote a high-level awareness about workload or activity.

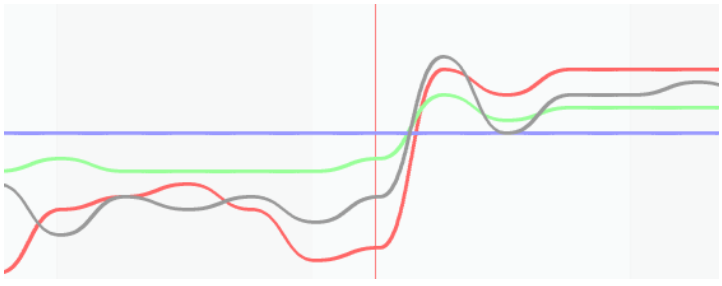


Figure 3.20: The system uses four curves to represent aggregate activity information. The curve color specifies the category. Each curve represents the activity in one category.

The system uses a quadratic Bezier algorithm to calculate curves [34]. There are different ways to construct a Bezier curve: linear curves, quadratic curves and higher-order curves. Assume we want to draw a curve from point $P_0$ to point $P_1$ . Linear curves use one control point that is simply a straight line between the starting point and ending point. Formula 3.8 defines the curve:

$$B(t) = P_0 + \mathrm{t}(P_1 - P_0) = (1 - \mathrm{t})P_0 + tP_1, t \in [0,1]. \quad (3.8)$$

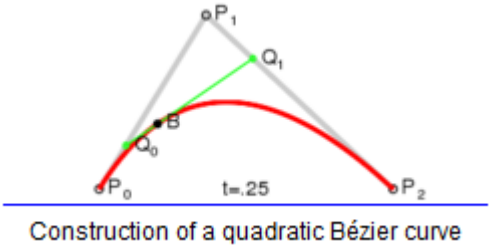



Figure 3.21: Construction of a quadratic Bezier curve. The curve starts from $P_0$ to $P_1$. It has two control points $Q_0$ and $Q_1$.

Figure 3.21 shows that a quadratic Bezier curve starts from $P_0$ to $P_2$ with two control points $Q_0$ and $Q_1$. Point $Q_0$ varies from $P_0$ to $P_1$ and describes a linear Bezier curve. Point $Q_1$ varies from $P_1$ to $P_2$ and describes a linear Bezier curve. Point $B(t)$ varies from $Q_0$ to $Q_1$ and describes a quadratic Bezier curve when $t$ varies from 0 to 1. Formula 3.9 defines the curve:

$$B(t) = (1 - t)^2 P_0 + 2(1 - t)tP_1 + t^2 P_2 \,, \; t \in [0,1]. \quad (3.9)$$

Higher –order curves make the Bezier curve smoother, but the tradeoff is that it is computationally expensive. The system recalculates four curves and refreshes them every time users drag and drop the timeline. When users keep dragging the timeline, the high computational complexity will cause a delay in the system. In order to maintain a reasonable tradeoff between performance and complexity, quadratic Bezier curves were determined to be the best choice. For quadratic Bezier curves, formula 3.10 defines the position of control points $Q_0$ and $Q_1$. $P_0$ is starting point and $P_2$ is ending point:

$$
\begin{aligned}
x_{Q_0} &= (x_{P_0} + x_{P_2})/2 \\
x_{Q_1} &= (x_{P_0} + x_{P_2})/2 \\
y_{Q_0} &= y_{P_0} \\
y_{Q_1} &= y_{P_2}
\end{aligned}
\tag{3.10}
$$

Formula 3.9 shows that when $t$ increases from 0 to 1, every Bezier curve point is calculated based on the position of $P_0$, $P_1$ and $P_2$. When $\Delta t$ is small enough, these points are very close and form a smooth Bezier curve. Our system sets $\Delta t$ equal to 0.01 to draw a smooth Bezier curve. From formula 3.8, 3.9 and 3.10, we calculate every Bezier curve point based on the positions of starting point $P_0$, ending point $P_2$, and control points $Q_0$ and $Q_1$.

$$
\begin{aligned}
x(t) = t^3\big(x_{P_2} + 3\big(x_{Q_0} - x_{Q_1}\big) - x_{P_0}\big) + 3t^2\big(x_{P_0} - 2x_{Q_0} + x_{Q_1}\big) \\
+3t\big(x_{Q_0} - x_{P_2}\big) + x_{P_2}, t \in [0,1]. \\
y(t) = t^3\big(y_{P_2} + 3\big(y_{Q_0} - y_{Q_1}\big) - y_{P_0}\big) + 3t^2\big(y_{P_0} - 2y_{Q_0} + y_{Q_1}\big) \\
+3t\big(x_{Q_0} - x_{P_2}\big) + x_{P_2}, t \in [0,1].
\end{aligned}
\tag{3.11}
$$

**Group Awareness and Collaboration Design.** Our earlier system does not show aggregate information about group activity. Users wanted to compare their activity to the average group activity to see their contribution to the group. ActivityStream has a feature to compare a user's activity data with the average curves of the group's activity. Figure 3.22 shows a user's activity data with group aggregate curves. Four curves represent the group's workload in four categories. Each curve is an average curve of group activity, which represents the average workload in the group. From the left side of Figure 3.22, we see the amount of user's activities is nearly equal to the average amount of activities in the group. The central part shows that the user has fewer activities than average activities in the group. The right part shows the user has more research activities than average research activities in the group. The comparison provides information about an individual's contribution to the group.

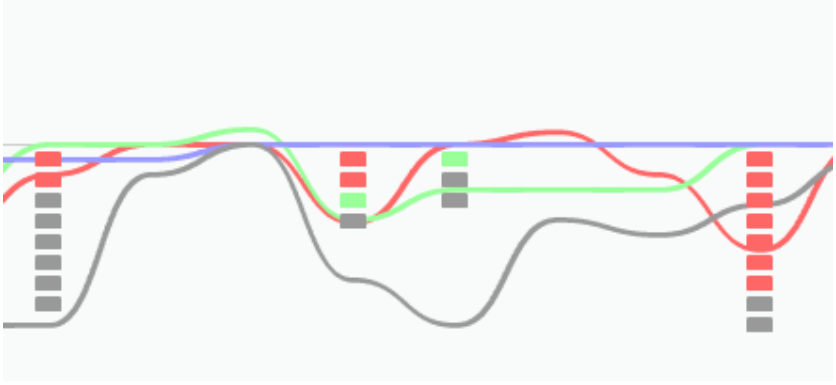


Figure 3.22: User's activity compared with average curve of group activity. Activity blocks show user's detailed information, curves represent average workload of group activity



Figure 3.23: Green arrows show the people users collaborated with

Aggregate curves increase group awareness by keeping users aware of how busy the other members of the group are. To promote group collaboration, ActivityStream is trying to determine which individuals the user collaborates with the most. Our observations show that users are inclined to include names in the activity content when they collaborate with them. The use of names implies group collaboration. Through analyzing the content of activity, ActivityStream sorts the users by the frequency of their names in the content and gets top three people the user collaborates. In Figure 3.23, green arrows below their thumbnails highlight the top three people.

### 3.3.4. *System Weaknesses*

- Even though the ActivityStream desktop tool provides aggregate information about the activities of one's peers, it does not support allow users to view the data from multiple users simultaneously in order to compare them. Users want to compare the aggregate curves of other users, but they can only view one person's curves at a time.

Comparing multiple calendars is useful when users want to find a spare time slot for several people. Google calendar and Outlook offer a similar function that enables users to examine multiple calendars simultaneoulsy. However, the events in Google calendar will overlap when the occurrence time is the same. Outlook shows multiple users' calendar by splitting the screen into several columns, but the screen for each column becomes narrower as the number of calendars increases.

- The ActivityStream desktop tool does not address scalability issues. It only works for a small group of people. When the group size increases, the system is not reliable. One problem is that when the group size increases, the computational complexity increases when the system calculates the Bezier curves. Now the system only has 12 users, but it has delays when users keep dragging the timeline. All the data will be refreshed and recalculated after users drag the timeline. The delay problem will be more problematic when group size increases. A possible solution is increase the $\Delta t$ in the Bezier curve formula that will save some time by computing fewer curve points. The other problem is the user thumbnails which will be very small when the user group is large, which is not convenient for user to identify the person with the thumbnail.
- The ActivityStream desktop tool does not provide hourly calendar visualization, which is useful when scheduling activities with others. The hourly visualization is normal in most calendar tools like Google calendar that users use to organize daily events and tasks.

## 3.4. Latest Work- ActivityStream Public Display

### 3.4.1. *Motivation and Design Principle*

Based on feedback in ActivityStream desktop tool, the new system should solve three problems: scalability issues, privacy issues and multiple calendar visualization. We want

to create a system to promote group awareness for a large group of people, and support multiple calendars visualization with privacy concerns. Considering the detailed information may cause privacy issues in large groups, the new system only provides high-level information about group activity.

The purpose of this new system is to increase high-level group awareness to a large group of people, and promote reflection. Reflection, defined in [22], is technology design outcome for HCI. It brings unconscious aspects of experience to conscious awareness, and thereby makes them available for conscious choice. Our system is a reflective design that promotes group awareness and reflection. In [35] the author presents slow technology, a description of the basic and intentional slowness in learning / understanding and in presence of a work of art, a piece of music or any other object designed for reflection. The technology could be slow in many ways:

1) Takes time to learn how it works
2) Takes time to understand why it works the way it works
3) Takes time to apply it
4) Takes time to see what it is
5) Takes time to find out the consequences of using it

The key issue in slow technology, as a design philosophy, is that we should use slowness in learning, understanding and presence to give people time to think and reflect.

The reflective system is evaluated by checking whether this system is capable to provoke such emotion so that increased awareness makes users reflect to their experience [26][27]. Therefore, our design principle is to convey an emotion to users to make them reflect. The data displayed in the system is calendar data in the group. Observation shows this is a very busy and hectic group. Group activities convey very stressful emotion.

Therefore, we want to build an emotional system that conveys stressful emotion with slow design technology.

### 3.4.2. *System Architecture and Data Collection*

Figure 3.24 shows the architecture. The system architecture and data collection is similar to one in earlier work-ActivityStream desktop tool. See section 3.4.2 for details. We make two changes in the new system: First, on the client side, we use Adobe Flex to build a public display system. We set a timer to control the system to retrieve data once every 24 hours from MySQL database. Second, after processing the data, the system stores the results in text file on client side. When the application on client side stops and restarts, it initializes the visualization in five seconds by reading processed data from local text file. Large groups may have more than 100 people. Calculating 100 Bezier curves takes almost one minute to run the animation. Reading processed data from local file saves time on initializing the visualization.

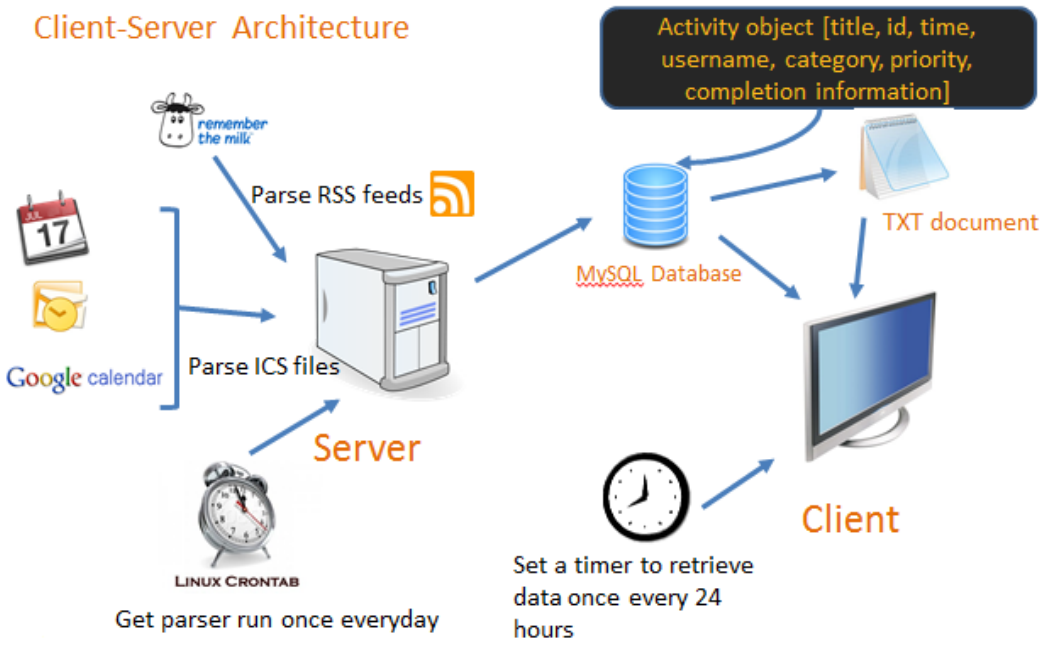



Figure 3.24: Client-server architecture. The server collects data from RTM by parsing RSS feeds, collects data from iCal, Outlook and Google calendar by parsing ICS files

every day, then stores data into database. A timer on client side controls the system to retrieve data once every 24 hours.

### 3.4.3. *Approach and Design*

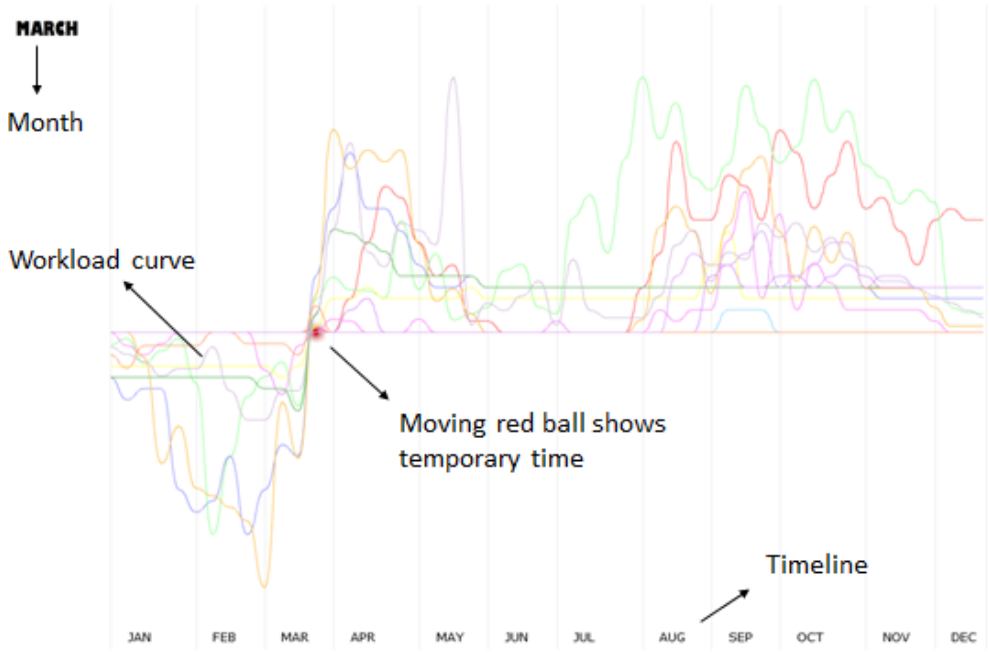



Figure 3.25: View of ActivityStream Public Display during the animation. Every curve represent a users' total workload. The colors represent different users. The red ball keeps moving and workload curves animate according to the workload at that temporary time point. Horizontal axis shows the timeline. Grids show the length of every month.

ActivityStream pubic display shows a looped animation. It shows how users' workloads changes over time and conveys stressful emotion to users to promote high-level group awareness and reflection. The whole animation shows the change of group workload in year 2010. Figure 3.25 is a screenshot of the animation. Every curve represents a user's workload. The color defines different users. A red ball is moving on the central line to represent temporary time point. It starts from Jan. 1 to Dec. 31. Left top corner shows the temporary month information. The horizontal axis is a timeline. The grids show the length of every month in 2010. Some curves are above the central line while some are below the central line. Above curves represent incomplete workload

curve and below curves represent complete workload curve, which is the same meaning with the curves in ActivityStream desktop tool. When the animation starts, the red ball moves slowly and the curves move depending on the workloads at the temporary time. The whole animation shows the change of workload very slowly. The aim is to use high-density, animated curves to convey stressful emotion to users, and keep them aware of how busy this group is. Next section we will discuss the color, aggregate information, emotional system and animation design.

**Color Design.** We design a color palette {0xff6666, 0x99ff99, 0xffff66, 0x9999ff, 0xfd8dfc, 0xfcc253, 0x6f6f6f, 0xdd99ff, 0x99ddff, 0xffa676, 0xdfa9f6, 0xd4bede}. Every color in the palette has high saturation that matches to the white background. The color of curve is randomly picked from the palette.

**Aggregate Information Design.** The system provides aggregate information about group activity to promote high-level group awareness. Similar to ActivityStream desktop tool, we use curves to represent aggregate information with two reasons. First, the system will support multiple users' visualization by using curves. Using curve to represent one's workload makes it easier to compare a group of people's workloads at a time. The visualization is still clear when the curves overlap a little bit. By comparing the curves, users know someone has higher workload while someone has lower workload. Second, curve visualization solves the scalability issues. The system only draws one more curve when the group has one more user. The system computational complexity is linear to the increase of group size. Section 3.3.3 explains the details how to calculate the curves.

**Emotional System Design.** Two methods could be used to convey stressful emotion. One way is using animated curves to represent stressful emotion. The curves move up and down slowly to show dynamic change of group activity over time. The animation is slow enough to give people time to think and reflect. The workload change will convey

stressful emotion. The other way is increasing the curve density. High-density curves show workload of a large group of people and convey a stressful emotion about the group activity.

**Animation Design.** To build a slow design to promote reflection, the whole animation lasts 60 minutes. Slow animation requires people to watch at least 10 minutes to observe the curve change in the display. Curves represent workload in 52 weeks. Every time the workload changes after a week passes because some activities are completed and some sections of curve push down. The change from old curves to new curves is regarded as a transfer between two stages. In Figure 3.26, the system draws 100 temporary curves between the old curve and new curve to make a smooth move from old curve to new curve. Formula 3.12 defined these temporary curves, is a progressively decreasing parameter from 1 to 0. When , ; when , .

(3.12)

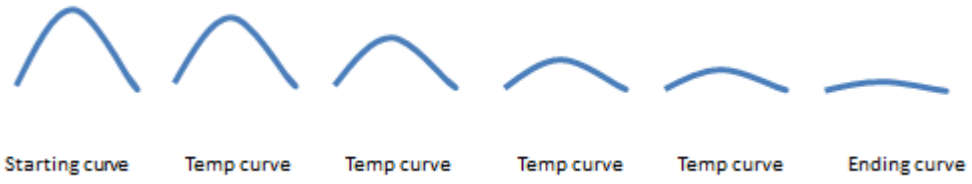



Figure 3.26: Curve changes from starting curve to ending curve. The system draws 100 temporary curves between starting curve and ending curve.

ActivityStream public display draws 510 points to make Bezier curve smooth. When the system runs the animation, it recalculates curve points, is the number of curves. When the group size is large, the high computational complexity will delay the animation. Every time the old curve turns to new curve after one week passes, only the curve points in that week update. Figure 3.27 shows an example, the blue curve turns to red curve, only the points in the block update. The data in one week involves 20 curve

points. To speed up recalculation, the system only updates the 20 curve points rather than the whole curve points.

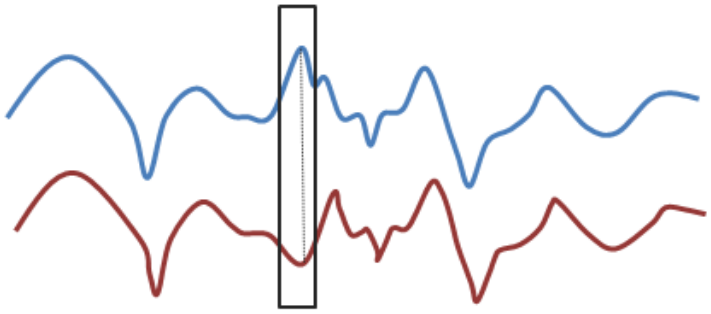


Figure 3.27: Recalculate one section of the curve. Blue curve is old curve and red curve is new curve. Compare the blue curve and red curve, only one section of the curve changes at a time. The system only update that section to speed up recalculation.

### 3.4.4. *System Weaknesses*

- This system processes task data and event data in the same way. Task data and event data has different properties. Users may fail to finish tasks, but events always occur. The incomplete and completion information displayed in this system only works for task data in RTM because RTM system set completion information for each task. For most calendar data, there is no such option to confirm the occurrence. That is why most curves in ActivityStream public display push down finally. Most calendar data are from Google calendar. The system cannot track completion information from that calendar system.
- User group is too small to prove the reliability of the system for large groups. When we do user study, we only have 10 users' data. After the user study, we keep adding more users into the group. The system has 30 users' data now and still performs well. To prove the reliability of the system for large groups, the system should have at least 100 users. We did an experiment with 100 dummy user data, the system takes almost one minute to initialize the display. There is no delay during the animation.

## 3.5. Summary

This chapter introduces the approach and design methodology used to design the calendar visualization tools. In our earlier work, we design a task visualization public display to provide detailed information for small groups. Users become more aware of tasks in the group. However, the system raises privacy issues, and does not provide aggregate information about tasks. Users use other calendar tools to manage their tasks, the task data collected in the system is not complete. The non-interactive system only provides single time scale visualization. Based on the limitation in task visualization public display, we refine it and design a new desktop tool-ActivityStream desktop tool. The new system allows users to examine their own detailed activity data and their peers' aggregate activity information. The system collects calendar data from all mainstream calendar tools. The login system and aggregate curves protect users' privacy. Interactive interfaces and multi-timescale visualization make it easier to browse calendar data. However, the limitation is that the system only works for small groups. Besides, the system does not support multiple calendars visualization. To overcome these limitations, we design another system-ActivityStream public display, which shows aggregate information about group workload, and increases high-level group awareness for large groups. This system follows slow design and reflective design principles, conveys stressful emotion about group workload and gives users time to think and reflect. The weakness is that the user group is still relatively small, which does not prove the reliability of the system for large groups.

# CHAPTER 4

# USER EXPERIMENTS AND RESULT ANALYSIS

We do a user study for each system. Every system collects calendar data from Reflective Living (RL) group. RL group is a multidisciplinary research group at a large state university. They focus on designing and developing dynamic media computation systems that promote reflective thinking, increase group awareness and support creative collaboration. For our earlier work-task visualization public display, we recruit nine users, six of them are from RL group. We ask every user to watch the display for ten minutes, then finish a questionnaire and a personal interview. For latest work-ActivityStream desktop tool, we recruit eight users. Since this system is designed only for RL group people, all users are from RL. We ask them to do a pre-questionnaire first, play the system for ten minutes, then do a post-questionnaire and finally finish a personal interview. For another latest work-ActivityStream public display, to test the influence to a large group of people, we recruit 13 users, eight of them are from RL group. We ask them to watch the display for 10-15 minutes, and then finish a personal interview. The three user experiments and result analysis will be discussed below separately.

## 4.1. User Experiment and Result Analysis for Task Visualization Public Display

### 4.1.1. *User Experiment Design*

Task visualization public display provides detailed task information for RL group. We want to examine the affection of the system to RL group, and we want to see whether this system is useful to people outside the group, help them get more aware of RL people's work. In this user study, we recruit nine users, six of them are from RL group, and the rest people are outside RL group. All users are students between 20-30 years old. Six of users are male and three of them are female. Every user is asked to watch the public

display for ten minutes, and then finish a questionnaire based on their observation. Finally, we have a personal interview with every user.

The questionnaire has five sections: usability test, functionality test, visual design test, additional functionality test and a survey for calendar tool use. Usability section examines whether the system is easy to use and understand. Functionality section examines whether the system is functional, increasing group awareness and promoting reflection. Visual design section examines whether users like the interfaces designed in the system. Additional functionality test examines whether users know who are busiest people and when RL people are busiest. The survey for calendar tool use examines what calendar tools they use.

### 4.1.2. *Questionnaire Results and Analysis*

Since we only have nine users, the user group is very small. The quantitative result (questionnaire result) is easily affected by single user's answer. The qualitative result (interview result) needs to be paid more attention. Scores ranges from 1 to 5 (1=strongly disagree, 5=strongly agree).

Table 4.1: Result of usability test

| **Usability** | **Questions** | **Scores** |
|---|---|---|
| Q1 | I can easily determine when a user has a deadline. | 4.3 |
| Q2 | It is easy to learn how to use this tool. | 3.8 |
| Q3 | This public display is adequately detailed. | 3.5 |
| Q4 | I would like to use this tool for my group. (Non RL only) | 3.0 |
| Q5 | I would like to use this tool for myself. | 3.1 |

The answer to Q1 reveals that users think this system helps them to determine when others have deadlines. The answer to Q2 shows most users agree that the system is easy to understand and learn. The answer to Q3 reveals the information in the display is not adequate. The reason is that the system only displays two information bubbles at a time,

which will be discussed in interview summary section later. The answer to Q4 shows non-RL people would not like to use this tool for their group, because the data is RL group's task data, which is unrelated to them.

The answers to Q1 is close to "Strongly agree", reveals that the work pattern is obvious in the system by displaying discrete task blocks. Users are interested to know the aggregate task information and observe the pattern. Answer to Q4 reveals the system does not increase knowledge of RL group's tasks for non-RL people. The reason is that RL people have prior knowledge about group activity, and the system increases that knowledge easily. However, non-RL users do not have the prior knowledge about RL group's tasks. They may feel confused about some task contents displayed in the system. Answers to Q2, Q3 reveal the system helps users to know who is busy and when the group is busy. The answer to Q5 shows the system helps users to go over their task data.

Table 4.2: Result for functionality test

| Functionality | Questions | Scores |
|---|---|---|
| Q1 | Different users have different task distribution patterns | 4.6 |
| Q2 | This tool helps identify the busiest member in the group | 3.8 |
| Q3 | This tool helps identify busiest period for people in Reflective Living group | 3.9 |
| Q4 | This tool helps me have a general understanding about Reflective Living group's past, current and future tasks (i.e. what they were about) (Non RL only) | 3.3 |
| Q5 | I can review tasks that I have completed and the tasks that I will work on in the future (RL only) | 3.7 |

The result reveals the top three features users like are color design, cartoon thumbnails and information bubbles. Colors help them to define the category information, cartoon thumbnails convey anonymous information about users, and the information bubble is a direct way to show task details. Answer to Q6 reveals the moving bar's moving speed makes users uncomfortable, the moving speed needs to be modified.

Table 4.3: Result of visual design test

| Visual Design | Questions | Scores |
|---|---|---|
| Q1 | The size of the deadline icons enables me to understand priority of the deadline | 3.6 |
| Q2 | The color of deadline icons enable me to recognize a person to the deadline | 4.3 |
| Q3 | The relationship between the three deadline categories (personal, coursework and research) makes the deadline information better organized | 3.8 |
| Q4 | The types of cartoon icons (happy, moderate and exhausted) enable me to understand deadline density (how many deadlines in that week) | 4.0 |
| Q5 | The moving vertical bar enables me to understand how the tasks change over time | 3.9 |
| Q6 | The speed of scrolling bar is proper so that I can review the deadline details | 2.6 |
| Q7 | Popping out bubbles understand the details of the deadline | 4.1 |
| Q8 | Popping out bubbles is distracting | 2.2 |

The additional functionality test asks two questions: A. “Who has the most tasks?” B. “For each user, which month the user has the most tasks?” Table 4.4 shows the answer to question A:

Table 4.4: Vote count for the busiest person depending on amount of tasks. Every user is asked to provide three busies people in RL group

| **Users** | #1 | #2 | #3 | #4 | #5 | #6 | #7 | #8 | #9 | #10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **Votes** | 9 | 2 | 0 | 0 | 7 | 2 | 0 | 6 | 0 | 1 |

For question A, every user provides three busiest people in RL group. The result in Table 4.4 shows the top busies people are: user #1 (9 vote), user #5 (7 vote) and user #8 (6 vote). Table 4.5 shows statistical analysis of task data in 4 month for each user. According to counts of total tasks, top three busiest people are user #1 (150 tasks), user #5 (32 tasks) and user #8 (31tasks). The results in Table 4.4 and 4.5 are matched.

Table 4.5: Count of complete tasks, incomplete tasks, total tasks for each user

| Users | Counts of complete tasks | Counts of Incomplete tasks | Counts of total tasks |
|---|---|---|---|
| #1 | 145 | 5 | 150 |
| #2 | 2 | 4 | 6 |
| #3 | 0 | 2 | 2 |
| #4 | 10 | 2 | 12 |
| #5 | 12 | 20 | 32 |
| #6 | 1 | 15 | 16 |
| #7 | 1 | 1 | 2 |
| #8 | 24 | 7 | 31 |
| #9 | 0 | 5 | 5 |
| #10 | 21 | 8 | 29 |

For question B, first, we count every user's task data based on statistical analysis. Figure 4.1 shows the results. Every single line represents a user. X-axis is a timeline, and Y-axis represents count of tasks. Based on Figure 4.1, Table 4.6 shows the busiest months for users.

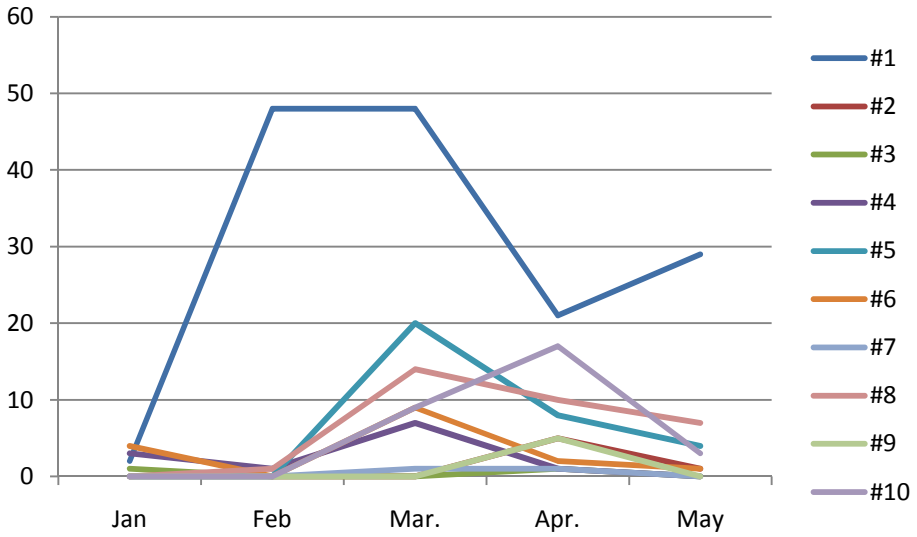



Figure 4.1: Task counts for each user in 4 months. Each line represent a user, x axis represents timeline, y axis represents counts.

Table 4.6: The busiest months for users based on statistical results

| Users | The busiest months |
|---|---|
| #1 | February or March |
| #2 | April |
| #3 | January or April |
| #4 | March |
| #5 | March |
| #6 | March |
| #7 | March or April |
| #8 | March |
| #9 | April |
| #10 | April |

Table 4.7 shows the users' feedback to question B, the busiest month for each user. Comparison of the results in Table 4.6 and Table 4.7 reveals that in the ten answers in Table 4.7, sever answers are correct. The answers for user #2, user #8 and user #10 are incorrect. Answers to question A and B reveals that the public display helps user to understand when people are busy and the busiest people in RL group.

Table 4.7: The busiest month for each user

| Users | The busiest months |
|---|---|
| #1 | February |
| #2 | March |
| #3 | April |
| #4 | March |
| #5 | March |
| #6 | March |
| #7 | March |
| #8 | February |
| #9 | April |
| #10 | March |

The last section is a survey for calendar tools use. Table 4.8 shows what are the calendar tools users are using. The red block means they use this tool, while green block means they use this tool as major tool. The result shows top three calendar tools are

paper-based notebook, Google Calendar and Remember the Milk. That explains why someone has small data while someone has large data. Many users prefer using other calendar tools rather than RTM. Collecting calendar from more mainstream calendar tools will make the data in our system more complete. The data in RTM is task data, but the data in other calendar tools includes both task data and event data. Tasks and events are different data, and both them reflect activities. We already discuss that in section 3.3.2. Our latest calendar visualization system visualizes both data to increase group awareness.

Table 4.8: Survey for calendar tool use, red areas mean that the user uses the calendar tool, green areas mean that this tool is a major tool for that user

| Users | Outlook | iCal | Google Calendar | RTM | Paper based notebook | Others | None of above |
|---|---|---|---|---|---|---|---|
| #1 | | | | | | | |
| #2 | | | | | | | |
| #3 | | | | | | | |
| #4 | | | | | | | |
| #5 | | | | | | | |
| #6 | | | | | | | |
| #7 | | | | | | | |
| #8 | | | | | | | |
| #9 | | | | | | | |

4.1.3. *Interview Summary*

- Most users think this system help to understand the detailed task information in the group. They usually only know their own tasks, but do not know other peers' tasks. They like the information bubbles, the problem is that they are not allowed to choose the bubbles they want to examine. Those information bubbles pop up randomly. They

want to see more information bubbles in the display to show more details. They think the timeline should be extended to go over long-period data.

- Users think the work pattern in the display is obvious. By comparing the task data along the timeline, they find out most people have a few tasks in January, February and May because that is the beginning and end of the semester. For research group, most tasks are finished in the semester. They also notice many people have more research tasks in April because of paper deadlines, and more coursework tasks at the beginning and end of the semester. Displaying discrete blocks is not proper to show data pattern. To display the pattern clearly, we could use continuous objects to show aggregate information.
- Users notice the data in the system is not complete. Some users have much more tasks than others do, which is not normal. Users' definition to the tasks is different, which affects the way they use RTM. Someone uses the task management tool to record every tiny task, while someone only records important tasks into RTM. Besides, some users use different calendar tools such as Google calendar. The data in RTM does not reflect the real workload in RL group. The system should collect data from more calendar tools to make data complete.
- Users want more interactive interfaces in the system to access information easier. They want to see information bubbles by clicking the task blocks. The timeline should be draggable to allow them to go over calendar data at any time. They want to control the information and check other peers' data by clicking their thumbnails. All these interfaces make users easier to access calendar information.
- Users feel uncomfortable to share details outside the group. We set the public display in the lab, and we regard it as a semi-public display, but it still raises privacy issues when some visitors come into our lab. Displaying detailed information in public

display always raises privacy issues. One solution is only showing aggregate information in the public display; the other solution is showing detailed information in a desktop tool.

- User feedback shows many users do not set category information for their tasks, and some of them never check "complete" box in RTM. They think that costs extra effort. They set priority value for the tasks in RTM to organize their tasks easier.

#### 4.1.4. *Summary*

This user study result is positive. Most users are more aware of the detailed task information in the group. They think this system help them to determine who is busy and when the group is busy. They are interested in the patterns showed in the display. However, the task data in the system is not complete and the detailed information in the public display raises privacy issues. Users want more interactive interfaces to access information easily.

## 4.2. User Experiment and Result Analysis for ActivityStream Desktop Tool

### 4.2.1. *User Experiment Design*

ActivityStream desktop tool enables to examine detailed task information of one's own tasks, within the context of his/her peers' anonymized task data. The user study examines whether this system makes RL group people more aware of their peers' workload and promotes reflection. This system is built for RL group people's use. We recruit eight users. All of them are RL group people between 20-30 years old. Five of them are male and three of them are female. Pre-questionnaire is a questionnaire before users use the system, and post-questionnaire is a questionnaire after users use the system. We ask users to finish a pre-questionnaire first, then they play ActivityStream desktop tool for a while to test the functionality and usability. After that, they finish a post-questionnaire. Finally, we conduct a personal interview to collect more feedback.

Many questions in pre-questionnaire and post-questionnaire are similar for comparison. The pre-questionnaire has three sections: a survey for calendar tool use; a set of general questions to investigate participants' knowledge about their own activities and their peers' activities; a survey for interactive interfaces design. Post-questionnaire includes the second section in pre-questionnaire: a set of general questions to investigate participants' knowledge about their own activities and their peers' activities.

### 4.2.2. *Questionnaire Results and Analysis*

Table 4.9: Survey for calendar tool use, green areas represent the calendar tools users use

| Users | Outlook | iCal | Google Calendar | RTM | Paper based notebook | Others | None of above |
|---|---|---|---|---|---|---|---|
| #1 | | | | | | | |
| #2 | | | | | | | |
| #3 | | | | | | | |
| #4 | | | | | | | |
| #5 | | | | | | | |
| #6 | | | | | | | |
| #7 | | | | | | | |
| #8 | | | | | | | |

Table 4.9 shows results of survey for calendar tool use. One user uses Outlook; two users use iCal; eight users use Google Calendar; three users use Remember the Milk; six users use paper based notebook; two users use others such as calendar applications in cell phone or sticky notes. The result shows the top tools they use are Google calendar paper-based notebook, and RTM. ActivityStream system collects calendar from Google calendar, iCal, Outlook and RTM, which covers all mainstream calendar tools.

Since we only have eight users, the user group is very small. The quantitative result (questionnaire result) is easily affected by single user's answer. The qualitative result

(interview result) needs to be paid more attention. Scores ranges from 1 to 5 (1=strongly disagree, 5=strongly agree).

Table 4.10: Results of pre questionnaire and post questionnaire, this table compares the results for eight questions in pre-questionnaire and post-questionnaire. The difference shows a percentage change from the results in pre-questionnaire

| Index | Questions | Pre | Post | Difference |
|---|---|---|---|---|
| Q1 | I am aware of my past workload (how many activities completed in the past) | 4 | 3.5 | -12.50% |
| Q2 | I am aware of my future workload (how many activities to do in the future) | 3.6 | 3.9 | 8.33% |
| Q3 | I am aware of the amount of activities completed by people in my group | 2.3 | 3.6 | 56.52% |
| Q4 | I have a good understanding of how busy people are in my group | 3.1 | 3.8 | 22.58% |
| Q5 | I am aware when people get busy in the RL group | 4.2 | 3.5 | -16.67% |
| Q6 | I am aware of collaboration in our group | 3.7 | 3.3 | -10.81% |
| Q7 | I am aware of my past interaction with other people in the group | 4 | 3.8 | -5.00% |
| Q8 | I am aware of the differences in workloads between me and my peers | 3.2 | 3.8 | 18.75% |

We test the statistical significance from the results to see if the answers in post-questionnaire have different distribution with the answers in pre-questionnaire. Table 4.11 shows p-value, t-test and standard deviation for each question. P-value is the main way to check statistical significance. In statistics, a result is called statistically significant if it is unlikely to have occurred by chance. The amount of evidence required to accept that an event is unlikely to have arisen by chance is known as the significance level or critical p-value [36]. A popular level of significance is 5% (0.05). The result is statistically significant when p-value is lower than 0.05. Based on the results in Table 4.11, Q3 and Q6 show statistical significance and the rest questions show non- statistical significance. That means the answers in Q3 and Q6 is different after users take the study. T-test results show whether the data follows the same distribution. The absolute of t-test value is large when the data does not follow the distribution. T-test results for Q3 and Q6

are large number, which also prove that the answers to Q3 and Q6 are statistical significance. The small group size is the reason that causes non-statistical significance The user group only has eight users. These quantitative results are easily affected by single user's answer. We have to recruit more users in the user study to solve this problem.

Table 4.11: Statistical significance test

| Questions | P-value | T-statistic | Standard deviation |
|---|---|---|---|
| Q1 | 0.35 | 1.00 | 0.71 |
| Q2 | 0.52 | -0.68 | 1.04 |
| Q3 | 0.03 | -2.76 | 1.28 |
| Q4 | 0.10 | -1.93 | 0.92 |
| Q5 | 0.65 | 0.48 | 1.49 |
| Q6 | 0.03 | 2.65 | 0.53 |
| Q7 | 0.35 | 1.00 | 0.71 |
| Q8 | 0.17 | -1.53 | 1.39 |

In Table 4.10, only the answers to Q3 and Q6 have statistical significance. Other answers do not change and are from the same distribution. The difference shows the percentage change from answers in pre-questionnaire. Formula 4.1 defines difference value. The difference in Q3 is 56.52%, which means users get more aware of group activity. The difference in Q6 is -10.81%, which means this system does not promote group collaboration.

$$Difference = \frac{Post\ score - Pre\ score}{pre\ score} \cdot 100\% \quad (4.1)$$

We ask another two questions to find out whether ActivityStream is useful to find spare time slots for multiple people: Q9: "*which week you feel is the best to schedule meeting with user A?"*; Q10: *"what is the best day if you plan to schedule group meeting with user A, user B and user C next week?"* Five options offered for question 9: this week; next week; the week after next week; three weeks after this week; at least one month after

this week. Five options offered for question 10: Monday; Tuesday; Wednesday; Thursday; Friday. Table 4,12 shows the result of Q9 and Q10. We observe the calendar data and find out the best answer to Q9 is “this week” or “at least one month after this week”, and the best answer for Q10 is “Monday” or “Wednesday”. Compared to our observation, the results shows six users are able to find a proper time to schedule activity with others.

Table 4.12: Statistic results of question 9 and question 10 in post questionnaire. Counts represent the number of users who choose that option

| Q 9 | Counts | Q 10 | Counts |
|---|---|---|---|
| This Week | 2 | Monday | 1 |
| Next week | 2 | Tuesday | 0 |
| The week after next week | 0 | Wednesday | 5 |
| Three weeks after this week | 0 | Thursday | 0 |
| At least one month after this week | 4 | Friday | 2 |

We conduct a survey to see users’ feedback to interfaces design in the system. Table 4.13 shows how many users like and dislike those features. The results show most interfaces get positive feedback. The top three features liked by users: categorize calendar data with colors; multi timescale visualization; interaction with canvas (drag timeline). Users like the system uses color to display category information. In our earlier work, we separate tasks in three columns in the display to show category information, which makes the display very cluttered. Using color to show category makes the display cleaner. Multi-timescale visualization reveals different activity patterns to users. Draggable timeline allows users to browns data easily. The top dislike interfaces: aggregate information for group and individual and tasks are ordered by priority (Size). Users want to see multiple users’ aggregate information for comparison, which is useful to find spare time slot by checking the lowest point of curves. Most users do not notice the change of block size because they do not set priority value for activities. Most blocks have the same size.

Table 4.13: Feedback to ActivityStream interfaces

| Interfaces | Like | Dislike |
|---|---|---|
| Brick metaphor for task | 5 | 1 |
| Interaction with bricks | 4 | 1 |
| Interaction with canvas | 6 | 0 |
| Multi timescale visualization | 6 | 1 |
| Aggregate information for group and individual | 4 | 2 |
| Grid | 5 | 1 |
| Tasks are ordered by category (color) | 8 | 0 |
| Tasks are ordered by priority(Size) | 2 | 1 |
| Separate complete and incomplete tasks | 4 | 0 |

### 4.2.3. *Interview Summary*

- Users' feedback is positive. They think this system makes them more aware of group activities. They think the multi-timescale visualization is useful to go over long-period data. The aggregate information shows their peers' workload, and they can compare them to find a proper time to collaborate. They want another time scale to show hourly calendar visualization.
- They feel more comfortable to share calendar in the group by showing aggregate information rather than details. They like the login system and think that solves the privacy issues raised in our earlier work.
- They feel the data is more complete since the system collects data from all mainstream calendar tools: Google calendar, Outlook, iCal and RTM. Users notice the difference of task data and event data. Some users do not like setting priority for activities because it costs extra effort.
- Some users notice the pattern in the display. The aggregate curves look similar, which means many recurring events. They want to highlight those recurring events in the display to save time on clicking them for details.

- Users think the visualization is cleaner, compared to our earlier work. They like the interactive interfaces in the system. It is easier to browse data with draggable timeline and clickable blocks. Compared to earlier work, the complete task blocks and incomplete task blocks are flipped. Many users think this visualization is more intuitive.
- Two users point out ActivityStream conveys a sense of collaboration to them, which is not clear in other online calendars. The green arrows below thumbnails imply group collaboration. One user says it is convenient to email co-workers by clicking the curves.

#### 4.2.4. *Summary*

The user study result shows users become more aware of group activities with the detailed and aggregate calendar information. The system promotes users' self-reflection by jogging their memory. They think the login system and aggregate information solve the privacy issues. The aggregate information shows patterns of workloads in the group. However, most users do not think the system is useful to promote collaboration. They think this system is more useful to increase group awareness. Users want the system to support multiple users' calendar display to compare workloads easier.

## 4.3. User Experiment and Result Analysis for ActivityStream Public display

### 4.3.1. *User Experiment Design*

ActivityStream public display shows aggregate information about group activity to large groups. We want to examine whether the system is increasing high-level group awareness and conveys stressful emotion about group workload. We recruit 13 users in the user study, eight of them are RL people and five of them are outsider RL group. All the users are students between 20-30 years old. Eight users are male and five users are female. Most users are the same as the users in the user study for earlier work-task

visualization public display and latest work-ActivityStream desktop tool. We ask users to watch the public display for 10-15 minutes, and then finish a personal interview.

Our interview includes three questions:

1) Did the public display raise any emotions with you? What kind of emotion you feel?
2) Are you able to read any developmental stages from the display?
3) What is the public display according your views? Interpret it.

The first question is to examine whether the system conveys stressful emotion about group workload to users. The second question is to examine whether the slow design gives users time to think and reflect. The third question is to examine whether users can interpret this emotional system correctly.

4.3.2. *Interview Results and Discussion*

- *Did the public display raise any emotions with you? What kind of emotion you feel?*

Six users sense emotions from the display. Users #4 and #9 say, “The density of curve conveys a hectic, crazy emotion. When the temporary time passed by, we notice that we still have a lot of work to do, which gives me a hectic emotion”. Users #5 and #8’s answer is different, “The high curve conveys positive emotion and the low curve conveys negative emotion. The curve animation gives me a sense of negative emotion”. Users #11 and #12 feel very stressful from the display, “the density of curves looks disordered, very stressful”.

Results show six users sense emotions from the display and four of the six users feel hectic, busy, or stressful emotion. In the 13 users, two females and four males sense emotions, and three females and four males sense nothing from the display. The results high likely have nothing to do with genders. Users #4 and #9 feel hectic and crazy emotion. Both them are PHD students. PHD students may do lots of research, and their

background leads to such hectic feeling. Users #5 and #8 are master students. Their understanding to the curve is very intuitive: Going down curves represent negative emotion. Users #11 and #12 are female PHD students. Their answer and description are more accurate than answers by users #4 and #9 (male PHD students). They feel stressful because of the high-density curves. The rest users do not feel any emotion from the display. Only six of users sense an emotion from the display and four of the six users sensed the right emotion. User say the curve density is one reason to raise stressful emotion, but the density is not high enough.

- *Are you able to read any developmental stages from the display?*

Users #7 and #13 have different answer from others. "At first beginning, I think it is static display, which shows beautiful curves. Even though I stare at it for 1 minute, I do not see any change in it. However, I notice the curve is changing after 10 minutes. The animation is incredible slow and the change is subtle." The rest users thought the animation is obvious, and they gave the answer firmly: "The red ball is moving, which is a temporary time point. The curve is going down when temporary time point is moving."

Only users #7 and #13 feel it is static. The rest users say they can read the change in the display firmly. The result is not surprising. Users #7 and #13 are outside RL group, this is their first time to watch this display. The public display uses a slow design to provoke reflection, which requires users to watch for a while to observe the subtle changes. When users #7 and #13 watch the display for the first time, they do not understand the slow design and it takes at least ten minutes to learn and reflect. The rest users are composed of RL group people and some people visit this lab very much. They already watch this display and understand the developmental stages before the interview. In sum, the change in the display is subtle and takes time to observe.

- *What is the public display according your views? Interpret it.*

It is very open question. Some users give multiple answers. The best match with correct intentions is by users #1, #4, #6, #8, #9, #10, #11, and #12, “I think it reflects group activity, the workload. It shows how busy the group is. The workload decreases when time is moving on. There are still incomplete tasks in the past, and I feel curious about that.” Four users give another possible answer (users #2, #4, #10, and #11): “The visualization shows time series data such as stock market index or financial data. At the beginning of the animation, it shows predicted curves. Then these predicted curves turn to real curves gradually”. User #4 claims that it is weird because the real data curve are always lower than predicted curves. Another interpretation is the weather report (users #3, #6, #7, and #13), “Every curve represents an attribute in weather report, such as rainfall, temperature, moisture, cloud, wind even earthquake.” Other users give unique answers. User #5 think it is a hormone curve. The curve represents the hormone level in human body, “The person feels very high when hormone curve is going up, but he/she feels depressed when the hormone curve is going down”. User #7 think it is GDP curve or national tourism income curve, “Every color means a state or something”. User #8 gives a very imaginative answer: “The animation looks like the whole process of forming stalactites”. User #12 connects this public display to information technology. She thinks the curve animation show data transfer process.

Results show users’ background and personality lead to the answers. Eight users give correct interpretation to the display. All of them are RL people. RL group is a research group focusing on visualizing group activity, that research experience gives them a sense about the display. Another reason is that the eight users’ data is related to the display. It is easier for them to reflect based on their own data. Besides, the pattern of curve is obvious: most curves have relatively low values in June, July and August. Users connect the data to academic activities naturally. Some curves’ shape are similar, which means the activity

patterns are similar. As an individual, user is aware of his/her activities, but they do not have general sense about how busy this group is until they watch these curves. For the rest users, these curves have nothing to do with them. It is difficult to explain unknown experience based on the display. They interpreted the display by guessing the unknown experience. Therefore, the rest users tend to interpret these curves as stock market curves or weather curves. Users' background has an impact on their interpretations. User #12 is a Computer Science PHD students, one of her interpretation is data package transfer. That interpretation is highly connected to her computer science background. In sum, correct interpretation depends on users' background and the emotion they feel from the system. When users realize that the display is related to their experience and shows their data, they get correct interpretation easily.

#### 4.3.3. *Summary*

Most users can interpret the display correctly. ActivityStream public display shows aggregate information about group workload. In the interview, users think this display supports multiple users' calendar display, which is easier for comparison. They think this system could support larger groups with more users. Showing aggregate workload curves in the system increases high-level group awareness in the group. However, some users think the emotion is not clear enough. The curves density could solve this problem, but the system only shows ten curves.

## 4.4. Summary of User Experiments

Three user experiments show our systems get better feedback with iterative design. In the user study for task visualization public display, users think the system provides detailed task information, but it raises privacy issues, the system is not interactive, the timeline is limited. In the user study for ActivityStream desktop tool, users think the system provides both detailed and aggregate information of group activity. Users become

more aware of other peers' work. Users like the clean visualization and interactive interfaces. Users like using multi-timescale to go over their data. The login system and aggregate curves make them more comfortable to share data, and protect their privacy. However, users want the system supports multiple calendar visualization. In the user study for ActivityStream public display, many users think the system provide aggregate information about group workload. The system increases high-level group awareness to the group by showing all the workload curves in the group. The system visualizes multiple users' calendar data by displaying workload curves, which makes users easier to compare. Some uses think the system is reliable for large size group, but the stressful emotion is not clear because the density of curves is not high enough. The stressful emotion about group workload will be clear with the increase of group size.

# CHAPTER 5

# CONCLUSION AND FUTURE WORK

## 5.1. Conclusions

### 5.1.1. *Conclusions of Iterative Design*

Table 5.1: Summary of iterative design, shows how the systems are refined. The symbol "√" means the system has that property.

| **Properties** | **Task Visualization Public Display** | **ActivityStream Desktop Tool** | **ActivityStream Public Display** |
|---|---|---|---|
| Public display | √ | | √ |
| Desktop tool | | √ | |
| Detailed information | √ | √ | |
| Aggregate information | | √ | √ |
| Privacy concerns | | √ | √ |
| Discrete objects | √ | √ | |
| Continuous objects | | √ | √ |
| Static | | √ | |
| Animation | √ | | √ |
| Single time scale | √ | | |
| Multi-timescale | | √ | |
| Scalability concerns | | | √ |

The research question in this thesis is to design calendar visualization tools to increase group awareness and promote reflection. We use iterative design methodology to refine our systems. Table 5.1 shows how we refine our tools and change the design. Our earlier work is task visualization public display, which shows detailed task information to small groups. The purpose is to provide up-to-date information, increases low-level group awareness in the group. It is using discrete objects to visualize task data. To show tasks in the group and tasks for individual, the system runs a looped animation to show tasks in the group and tasks for each individual. Based on users' feedback, the Pros and Cons are shown below.

The Pros are:

- The system provides detailed task information and increases low-level group awareness.
- The system provides group view mode and individual view mode to show the group's task data and each individual's task data separately.
- Cartoon faces show lightweight aggregate information – individual workload in a week.

The Cons are:

- The system raises privacy issues when users share task data.
- The timeline is limited and it only has one time scale.
- The system does not provide enough aggregate information to observe the pattern of task data.
- The system is non-interactive and the visualization is cluttered.
- The task data is not complete because many users use other calendar tools.

To solve the problems in earlier work, we design ActivityStream desktop tool, which shows both detailed and aggregate information about group activity. The system uses login system and aggregate curves to solve privacy issues, shows aggregate information with continuous curves to protect users' privacy. Many interactive interfaces in the system make users easier to access information. Multi-timescale visualization in the system shows different pattern of calendar data. The system collects data from many mainstream calendar tools: Google calendar, Outlook, iCal and RTM. Based on users' feedback, the Pros and Cons are shown below.

The Pros are:

- The system provides interactive interface to examine detailed information.
- The system provides aggregate information about peers' work.
- The system uses login system to protect users' privacy.

- Multi-timescale visualization shows different data pattern.
- Draggable timeline and other interactive interfaces make users easier to access information.
- The system collects data from all mainstream calendar tools to make the data complete.

The Cons are:

- The system does not support visualizing multi calendars.
- The system does not provide hourly visualization.
- The system does not solve scalability issues when group size increases.

To solve these problems, we design another public display-ActivityStream public display, which provides aggregate information about group workload for large groups. The system only shows aggregate information with curves to protect privacy in large groups. The system follows the slow design and reflective design to convey a stressful emotion about group workload and give users time to think and reflect. Based on users' feedback, the Pros and Cons are shown below.

The Pros are:

- The system solves scalability issues by showing aggregate workload curves. The system draws one more curve when the group adds one more user. The computational complexity of the system is linear to the increase of group size.
- The slow design in the system gives users time to think and reflect.
- The system provides high-level group awareness by showing aggregate information about group workload.

The Cons are:

- Small user group does not prove the reliability of the system for large groups. The user group in the user study is ten. After the user study, the group size increases to 30. The user group is still relatively small.
- The stressful emotion is not clear because the density of curves is not high enough. The emotion about group workload will be clearer when the group size increases.

### 5.1.2. *Potential Extension of Current Work*

- In ActivityStream desktop tool system and ActivityStream public display system, the task data and event data are processed in the same way. Tasks and events are different data, and they have different properties. Our future system should separate them and process them in different way.
- We collect data from Google calendar, iCal, Outlook and Remember the Milk. Maybe we should collect data from other sources to make data complete. Physical notebook is hard to track now, but probably in the future, the calendar data on notebook could be synchronized to digital calendars.
- After we finish our user study for ActivityStream public display, we add more users into the system. Now the system has 40 users and it still performs well. To prove the system is reliable for large groups, we will keep adding users to examine the reliability of the system.
- Many users notice the recurring events in our visualization. The future system may highlight that data for users to understand the pattern easily. Recurring events create similar pattern along the timeline.
- Many uses want to compare multiple calendars in desktop tool, the future system will allow users to watch multiple calendars at a time.
- The future system will provide another time scale-hourly visualization, to make user easier to examine the calendar data in one day.

- Many users do not like inputting category information and priority information into the calendar system. The future system probably collects this information by parsing the contents.
- ActivityStream desktop tool only works for small groups. We consider the possibility to use it in a large group. The main issue is that user thumbnails take up too much space in the visualization. The future system will use a list of usernames to replace the thumbnails. When user pick a username from the list, that user's thumbnail will displays on the bottom and the visualization display the user's calendar data.

### 5.1.3. *Future Direction*

ActivityStream desktop tool has good performance on visualizing large time series data. We consider using this system to visualize other group activity. Twitter is a very popular social network in the world. It is feasible to use the framework in ActivityStream to create twitter application for users to browse their tweets and their friends' tweet activity. The new system uses a small block represents a tweet. All the tweets sent by users are stacked above the central line while all the tweets replied by others are stacked below the central line. The multi timescales shows daily, weekly and monthly twitter activity. Users check twitter details by clicking the blocks. The color of blocks identifies the user. By comparing the colors of blocks, users find out the replies from other people easily. In addition, the system will show aggregate information for other people's twitter activities. The system will promote group awareness by visualizing twitter activities.

REFERENCES


[1] “Cubicle - Wikipedia, the free encyclopedia.”

[2] P. Dourish and V. Bellotti, “Awareness and coordination in shared workspaces,” *Proceedings of the 1992 ACM conference on Computer-supported cooperative work*, 1992, pp. 107–114.

[3] J. Tullio, J. Goecks, E.D. Mynatt, and D.H. Nguyen, “Augmenting shared personal calendars,” *Proceedings of the 15th annual ACM symposium on User interface software and technology*, 2002, pp. 11–20.

[4] J.D. Mackinlay, G.G. Robertson, and R. DeLine, “Developing calendar visualizers for the information visualizer,” *Proceedings of the 7th annual ACM symposium on User interface software and technology*, 1994, p. 118.

[5] E.M. Huang and E.D. Mynatt, “Semi-public displays for small, co-located groups,” *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2003, pp. 49–56.

[6] W. Aigner, S. Miksch, W. M\üller, H. Schumann, and C. Tominski, “Visualizing time-oriented data–A systematic view,” *Computers & Graphics*, vol. 31, 2007, pp. 401–409.

[7] J.J. Van Wijk and E.R. Van Selow, “Cluster and calendar based visualization of time series data,” *infovis*, 1999, p. 4.

[8] D. Goren-Bar, Y. Shahar, M. Galperin-Aizenberg, D. Boaz, and G. Tahan, “KNAVE II: the definition and implementation of an intelligent tool for visualization and exploration of time-oriented clinical data,” *Proceedings of the working conference on Advanced visual interfaces*, 2004, pp. 171–174.

[9] D. Vogel and R. Balakrishnan, “Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users,” *Proceedings of the 17th annual ACM symposium on User interface software and technology*, 2004, pp. 137–146.

[10] G.B. Terrell and D.S. McCrickard, “Enlightening a co-located community with a semi-public notification system,” *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, 2006, pp. 21–24.

[11] S. Greenberg and M. Rounding, “The notification collage: posting information to public and personal displays,” *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2001, pp. 514–521.

[12] D. Vyas, M. van de Watering, A. Eli\ëns, and G. van der Veer, “Engineering Social Awareness in Work Environments,” *Universal Access in Human-Computer Interaction. Ambient Interaction*, 2007, pp. 254–263.

[13] D. Vogel and R. Balakrishnan, “Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users,” *Proceedings of the 17th annual ACM symposium on User interface software and technology*, 2004, pp. 137–146.

[14] K. Schmidt, “The Problem withAwareness': Introductory Remarks onAwareness in CSCW',” *Computer Supported Cooperative Work (CSCW)*, vol. 11, 2002, pp. 285–298.

[15] “Workspace Awareness in Real-Time distributed groupware.pdf.”

[16] C. Gutwin, R. Penner, and K. Schneider, “Group awareness in distributed software development,” *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, 2004, pp. 72–81.
[17] B. Kerem, J.M. Rommens, J.A. Buchanan, D. Markiewicz, T.K. Cox, A. Chakravarti, M. Buchwald, and L.C. Tsui, “Identification of the cystic fibrosis gene: genetic analysis,” *Science*, vol. 245, 1989, p. 1073.
[18] F.B. Viégas and M. Smith, “Newsgroup crowds and authorlines: Visualizing the activity of individuals in conversational cyberspaces,” *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on*, 2004, p. 10.
[19] S. Chandrasekaran, O. Cooper, A. Deshpande, M.J. Franklin, J.M. Hellerstein, W. Hong, S. Krishnamurthy, S.R. Madden, F. Reiss, and M.A. Shah, “TelegraphCQ: continuous dataflow processing,” *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, 2003, p. 668.
[20] J.M. Carroll, M.B. Rosson, G. Convertino, and C.H. Ganoe, “Awareness and teamwork in computer-supported collaborations,” *Interacting with Computers*, vol. 18, 2006, pp. 21–46.
[21] B. Whitworth and A. De Moor, *Handbook of Research on Socio-Technical Design and Social Networking Systems*, Citeseer, 2009.
[22] D.B. Lange and M. Oshima, “Seven good reasons for mobile agents,” *Communications of the ACM*, vol. 42, 1999, pp. 88–89.
[23] P. Sengers and B. Gaver, “Staying open to interpretation: engaging multiple meanings in design and evaluation,” *Proceedings of the 6th conference on Designing Interactive systems*, 2006, pp. 99–108.
[24] K. Boehner, P. Sengers, Y. Medynskiy, and G. Gay, “Opening the frame of the art museum: Technology between art and tool,” *Digital Arts and Culture (DAC)*, 2005, pp. 123–132.
[25] K. H\ö\ök, P. Sengers, and G. Andersson, “Sense and sensibility: evaluation and interactive art,” *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2003, pp. 241–248.
[26] \.A. Kirsten Boehnera, R. DePaulab, P. Dourishc, and P. Sengersa, “How emotion is made and measured,” *Int. J. Human-Computer Studies*, vol. 65, 2007, pp. 275–291.
[27] K. Boehner, R. DePaula, P. Dourish, and P. Sengers, “Affect: from information to interaction,” *Proceedings of the 4th decennial conference on Critical computing: between sense and sensibility*, 2005, pp. 59–68.
[28] W. Gaver, “Cultural commentators: Non-native interpretations as resources for polyphonic assessment,” *International Journal of Human-Computer Studies*, vol. 65, 2007, pp. 292–305.
[29] “Iterative design - Wikipedia, the free encyclopedia.”
[30] “Review: Remember the Milk.”
[31] “Five Best To-Do List Managers.”
[32] “cron - Wikipedia, the free encyclopedia.”
[33] “FaceYourManga.com | Shake Yourself!.”
[34] “Bézier curve - Wikipedia, the free encyclopedia.”

[35] L. Halln\äs and J. Redstr\öm, “Slow technology–designing for reflection,” *Personal and Ubiquitous Computing*, vol. 5, 2001, pp. 201–212.
[36] “Statistical significance - Wikipedia, the free encyclopedia.”