VisMe - Um diário pessoal em imagens

Paulo André de Azevedo Gomes

Dissertação para obtenção do Grau de Mestre em Engenharia Informática e de Computadores

Júri

Presidente: Prof. João António Madeiras Pereira
Orientador: Prof. Daniel Jorge Viegas Gonçalves
Vogais: Prof. Carlos Alberto Pacheco dos Anjos Duarte

Novembro 2010
Acknowledgements

First, I have to thank professor Daniel Gonçalves for his continuous dedication, from beginning to end. This work would not be what it is without his knowledge and experience guiding me every step of the way. He always believed in my capabilities, even when I was not sure of what I was doing. Knowing that I had to meet his often intimidating expectations pushed me to work harder than I would have otherwise.

I am also grateful to Sandra Gama for contributing to the discussions that led the development of the solution described in this dissertation, for organizing the heuristic evaluation session, and for her valuable help in composing the papers that documented this work over the last year.

I must acknowledge all of those with whom I discussed this work for their different perspectives and unbiased critiques. Likewise, I have to thank all the people who participated in the heuristic evaluation and in the usability tests. You may remain anonymous, but your dedication shall linger forever in these pages.

Last, but definitely not least, I have to thank my parents for supporting me throughout my life. I think they had a legal obligation to do it for most of it, but they didn’t have to do it as well as they did.
Resumo

Vivemos rodeados por computadores e muitas das nossas actividades diárias centram-se no seu uso. Esta interacção constante deixa no seu rastro vastas quantidades de informação pessoal. Contudo, não somos capazes de observar esta informação como um todo integrado, uma vez que precisamos frequentemente de aplicações diferentes para aceder a tipos de documento distintos e, mesmo assim, conseguimos apenas ter uma visão limitada de um subconjunto da nossa informação pessoal a cada momento. Propomos que técnicas de visualização de informação pessoal podem ser usadas para exibir uniformemente toda a informação heterogénea contida e associada aos vários documentos que constituem as nossas identidades digitais de uma forma que facilite a sua análise, permitindo-nos recordar o passado, descobrir factos interessantes sobre nós próprios, e recuperar documentos quase esquecidos. No processo de estudar as melhores abordagens para visualizar e interagir com esta informação, implementámos e testámos uma ferramenta dinâmica e interactiva de visualização de informação pessoal, denominada VisMe. Nesta dissertação, detalhamos esse processo, começando com uma análise do trabalho já realizado no campo da visualização de informação pessoal; continuando com uma explicação da nossa solução, um sistema que dispõe a informação em linhas temporais interligadas e permite a sua exploração progressiva ao mesmo tempo que preserva o contexto global; e acabando com os resultados de dois testes de usabilidade que validam esta solução mostrando que os utilizadores são capazes de usar a interface disponibilizada para identificar padrões e tendências pessoalmente relevantes e também descobrir documentos com base nas suas propriedades e contexto associado.
Abstract

We are surrounded by computers and many of our daily activities are centered on them. This constant interaction leaves vast quantities of personal information on its trail. However, we are not capable of observing that information as an integrated whole since we often require different applications to access distinct document types and, even then, we can only glance at small subsets of our personal information at a time. We propose that information visualization techniques can be used to uniformly display the heterogeneous information contained in or associated to the various documents that make up our digital selves in a way that facilitates their analyses, allowing us to remember our past, discover interesting facts about ourselves, and retrieve nearly forgotten documents. In the process of studying the best approaches to visualize and interact with this information, we implemented and tested VisMe, a dynamic interactive personal information visualization tool. In this dissertation, we describe that process, starting with a survey of previous work in the field of personal information visualization; continuing with an explanation of our solution, a system that presents information in interconnected timelines and allows users to progressively explore their personal information while preserving the overall context; and ending with the results of two usability tests that validate that solution by demonstrating that users are capable of using the provided interface to identify personally relevant trends and patterns and also discover documents based on the knowledge of their properties and associated context.
Palavras Chave

Keywords

Visualização de Informação
Gestão de Informação Pessoal
Recuperação de Documentos Pessoais
Desenho Centrado no Utilizador

Information Visualization
Personal Information Management
Personal Document Retrieval
User-Centered Design
# Contents

1 Introduction .......................... 1
  1.1 Contributions .......................................................... 2
  1.2 Publications ........................................................... 2

2 Related Work ................................................. 3
  2.1 Email ................................................................. 3
    2.1.1 Themail - Visualizing Email Content: Portraying Relationships from Conversational Histories .......................................................... 3
    2.1.2 PostHistory - Digital Artifacts for Remembering and Storytelling: posthistory and Social Network Fragments ........................................... 4
    2.1.3 Using intimacy, chronology and zooming to visualize rhythms in email experience .................. 6
    2.1.4 Dynamic Coordinated Email Visualization .......................................................... 7
    2.1.5 Visualizing electronic mail ................................................................. 7
    2.1.6 Email Visualizations to Aid Communications ....................................................... 8
    2.1.7 Bloom: An Interactive, Organic Visualization of Starred Emails ....................................... 10
    2.1.8 Graphical Browsing of Email Data: A Usability Based Comparative Study .......................... 10
  2.2 Instant Messaging .................................................. 11
    2.2.1 CrystalChat: Visualizing Personal Chat History ..................................................... 11
    2.2.2 Egocentric Analysis and Visualization of Instant Messaging Activity ........................... 12
  2.3 Text Documents .................................................... 13
    2.3.1 Gist icons: Seeing meaning in large bodies of literature .............................................. 13
    2.3.2 ThemeRiver: Visualizing Theme Changes over Time .................................................. 14
    2.3.3 DocCube: multi-dimensional visualization and exploration of large document sets ............... 15
    2.3.4 Visualizing the Non-Visual: Spatial analysis and interaction with information from text documents ................................................... 16
    2.3.5 Info Navigator: A Visualization Tool for Document Searching and Browsing ....................... 17
    2.3.6 Jigsaw: Supporting Investigative Analysis through Interactive Visualization ........................ 18
  2.4 Multiple Sources .................................................. 20
    2.4.1 LifeLines: Visualizing Personal Histories .......................................................... 20
    2.4.2 MyLifeBits: Fulfilling the Memex Vision .......................................................... 20
    2.4.3 TimeMachine Computing: A Timecentric Approach for the Information Environment ............... 22
    2.4.4 Lifestreams: an alternative to the desktop metaphor ................................................ 22
    2.4.5 Stuff I’ve Seen: A System for Personal Information Retrieval and Re-Use ....................... 23
    2.4.6 Milestones in time: The value of landmarks in retrieving information from personal stores .................................................. 24
    2.4.7 Fast, Flexible Filtering with Phlat - Personal Search and Organization Made Easy .............. 25
List of Figures

2.1 Themail .......................................................... 4
2.2 PostHistory ...................................................... 5
2.3 Social Network Fragments ........................................ 5
2.4 faMailiar .......................................................... 6
2.5 MailView .......................................................... 7
2.6 Visualizing electronic mail ....................................... 8
2.7 Email Visualizations to Aid Communications, document overview .................................................. 9
2.8 Email Visualizations to Aid Communications, prototype client interface ........................................... 9
2.9 Remail, final prototype ........................................... 9
2.10 Bloom ............................................................ 10
2.11 LinearVis .......................................................... 11
2.12 MatrixVis .......................................................... 11
2.13 ChrystalChat, top view .......................................... 12
2.14 ChrystalChat, side view and expanded conversation .......................................................... 12
2.15 Egocentric Analysis and Visualization of Instant Messaging Activity .................................................. 13
2.16 Gist icons .......................................................... 14
2.17 ThemeRiver ......................................................... 14
2.18 DocCube ............................................................ 16
2.19 Visualizing the Non-Visual, ThemeScape visualization .......................................................... 16
2.20 InfoNav, Sammon map ........................................... 17
2.21 InfoNav, Dendro map ............................................. 18
2.22 InfoNav, radial visualization .................................... 18
2.23 Jigsaw. From left to right and top to bottom: the list, graph, scatter plot, and text views .................................................. 19
2.24 LifeLines, criminal records .................................... 20
2.25 MyLifeBits, timeline view ....................................... 21
2.26 MyLifeBits, clustered-time view ................................ 21
2.27 TimeScape .......................................................... 22
2.28 Lifestreams .......................................................... 23
2.29 Stuff I've Seen ...................................................... 24
2.30 Milestones in time ................................................ 24
2.31 Phlat ................................................................. 25
2.32 Personal chronicling tools, closeup on the filters .......................................................... 26
2.33 FacetMap ............................................................ 26
2.34 Feldspar ............................................................. 27
3.1 VisMe sketch, early timeline inspired by Themail, including a traditional interface with a search field and filters .......................................................... 32
3.2 VisMe sketch, contacts in a wordcloud .......................................................... 32
3.3 VisMe, keywords in a timeline. ................................................................. 33
3.4 VisMe, full view of keywords, people, and documents expanded from several different elements. ................................................................. 33
3.5 VisMe, a new timeline being dragged out of the initial element in contrast to the perpendicular timeline extracted before it. ................................. 34
3.6 VisMe, highlighted keywords. Highlights can effectively emphasize the evolution of elements or of groups of elements. Here, one can clearly observe a shift of interests, from the red group of keywords until June, to the blue group from that point forward. ........ 35
3.7 VisMe, text search. ............................................................................. 35
3.8 VisMe sketch, filter input. ................................................................. 36
3.9 VisMe sketch, keywords exchanged between the user and a contact. .... 36
3.10 VisMe, filtered timeline with the current filters displayed in the bottom left corner. ................................................................. 37
3.11 VisMe, collision detection. Above, only one projected bounding box overlaps the opposing bounding box. Below, both projected bounding boxes overlap, meaning there is a collision. 38
3.12 VisMe, a tooltip explains the operation of a timeline control button as the mouse moves over it. ................................................................. 39
3.13 VisMe sketch, early icons on top, final letters below. ........................ 40
3.14 VisMe, use case. The exploration starts with a general keyword timeline. ................................................................. 40
3.15 VisMe, use case. After searching for a keyword, both its instances and the months in which it appears are highlighted. ................................................................. 41
3.16 VisMe, use case. To the right are the results of the exploration: two documents fitting the known context. ................................................................. 41

4.1 VisMe, architecture. ........................................................................... 43
4.2 VisMe, several elements with their bounding boxes and graphical representations. ................................................................. 46

5.1 First user evaluation, average task completion times in seconds (95% confidence interval). ................................. 52
5.2 First user evaluation, average task difficulty in a four point scale, with 4 being the least difficult (95% confidence interval). ................................................................. 53
5.3 Second user evaluation, average task completion times in seconds (95% confidence interval). ................................................................. 60
5.4 Second user evaluation, average task difficulty in a five point scale, with 5 being the least difficult (95% confidence interval). ................................................................. 61
List of Tables

2.1 System comparison in order of appearance in this document. .......................... 28

5.1 Number of problems detected by the four experts for each of Nielsen's heuristics. .... 48
5.2 First user evaluation, document retrieval task times in seconds. .......................... 52
5.3 First user evaluation, document retrieval task difficulty, from hardest (1) to easiest (4). 53
5.4 First user evaluation, pattern recognition task times in seconds. .......................... 54
5.5 First user evaluation, pattern recognition task difficulty, from hardest (1) to easiest (4). 54
5.6 Second user evaluation, task times in seconds. .................................................. 59
5.7 Second user evaluation, task difficulty, from hardest (1) to easiest (5). .................. 60
Chapter 1

Introduction

Considering the near constant presence of computers in our daily lives, it is likely that there is much to be learned about ourselves in all the information we continuously store. From the papers we write to the emails we exchange, most documents we read and manipulate are associated to information about who we are and what we do.

Yet, our contact with our personal computer history is often limited to searching and browsing document folders and email archives, revisiting bookmarks, or accessing our Internet browser’s history. The information is scattered around different locations and contained in multiple document types, which in turn are accessed by means of distinct and often unrelated applications. Within the operating system and many separate applications, hierarchical organization is still the most prevalent organization system, a system flawed by its need for constant and consistent effort to maintain and its inability to convey all possible associations between different files and bits of information beyond rigid hierarchies. It is, then difficult for us to get a global view of our digital lives using current tools.

Knowing this, our goal is to investigate possible approaches for the visualization of the content and associated properties of the documents that constitute a person’s digital presence, including text documents of all sorts and emails, in a way that allows users to identify personally relevant trends and patterns.

This visualization should integrate different kinds of personal information and display them uniformly, to allow users to inspect their personal information in a unified way. It should be interactive, to facilitate exploration and, throughout this exploration, it should preserve the surrounding context, allowing users to interconnect their discoveries and make sense of all that information. Additionally, it would be interesting if such a system could provide some means for finding and retrieving the documents themselves, making use of the contextual exploration to facilitate those tasks. While this is not the focus of our research, providing the means to explore the personal context surrounding the documents may help users locate documents that would otherwise be hard to find, due to a lack of precise knowledge of its properties and location, thus offering a useful supplement to traditional search and browsing tools.

During the course of this thesis, we studied the most appropriate techniques to visualize and interact with personal information in a way that satisfied our goals. This led to a solution in which information (the content of the documents in the form of keywords; the contacts responsible for creating, sending, and receiving those documents; and document names or titles) is presented in timelines. Starting from the broad view of all the available information, users can progressively narrow down their search by opening new timelines from any element in the visualization.

This solution was continuously developed and tested over a year. This included two rounds of usability tests that confirmed that the solution facilitates the discovery of personally relevant trends and patterns, and the retrieval of documents by taking advantage of the visibility of their surrounding context.
This dissertation documents the development, refinement, and validation of our solution, starting with a survey and a discussion of related work in chapter 2, followed by a description of the proposed solution and its implementation in chapters 3 and 4 respectively, and ending with a series of heuristic and usability evaluations as described in chapter 5. Finally, we present the conclusions of this dissertation and possible avenues for further developments in chapter 6.

1.1 Contributions

We have developed a working solution for the visualization of personal information. VisMe, a dynamic, interactive personal visualization system, facilitates the discovery of personally relevant trends and pattern and can also aid in the search and retrieval of individual documents based on the knowledge of its associated properties and temporal context. The visualization and user interface techniques we employed may also be useful for similar applications were one seeks to interactively observe the content of large, heterogeneous data sets.

- Integrated visualization: information extracted from distinct sources is visualized together in timelines. These structures can be applied to all sorts of information with temporal existence, they are trivial to read and understand, and with the addition of different weight markings by way of changes in font size they can also distinguish different levels of relevance within the information under analysis.

- Interaction: the direct interaction with visible elements, from the expansion of timelines from a facet to the combination of several facets by dragging and dropping, is a straightforward alternative to traditional search and browsing that amplifies the potential for exploration with a minimum of complexity attached to the interface.

- Minimal interface: all interactive and descriptive elements of the interface supporting the visualization are mostly rendered as black text over a white background, with few extraneous elements. Not only does it make for an aesthetically pleasing and coherent interface, it also maximizes the amount of useful information, since everything that is drawn on screen has a meaning and a purpose.

- Functional prototype: we have also implemented a functional prototype application in which these techniques can be observed and which was successfully used to demonstrate the ability of our solution to assist in the performance of representative tasks.

1.2 Publications


Gomes, P., Gama, S., Gonçalves, D., Exploração e Visualização de Informação Pessoal in Interacção 2010, IEETA/Universidade de Aveiro, October 2010.

Chapter 2

Related Work

Information visualization is currently a large field of research. Information of all sorts can be visually represented to facilitate its understanding, from particle collision data to photographs. Indeed, several personal information visualization tools have been developed over the years, often focusing on particular sources or aiming at different specific goals, from a practical standpoint of document retrieval to a purely recreational exploration of one’s past. For VisMe, understandably, we are only concerned with representing the kind of personal information that people commonly have in their computers. That includes emails, instant messaging logs, and text documents.

Considering the scope of this work, two important criteria for the selection of papers for this survey were the representation of collections of items and the focus, or at least the approximation, on the point of view of a single person. There are, for instance, many interesting visualizations of the content of individual text documents which do not have direct applicability on a uniform visualization of collections of heterogeneous items. On the same token, social networking websites have become a large part of a lot of people’s lives, yet the current visualizations of the information contained therein are, not surprisingly, focused almost exclusively on the representation of the networks and the similarities and disparities between groups of people, without a clear focus on a single user besides some marginal display options.

The ability to represent information from multiple sources is also important, but there is a limited selection of systems with that ability and we feel there is also valuable insight to be gained from the analysis of visualizations of collections of emails or text documents.

2.1 Email

Email is one of the most extensively used methods of electronic contact, but while its use has increased over the years, email clients have essentially stayed the same. People’s interaction with email is mostly done by browsing large lists with only the title and the sender as the description of their contents. The following papers describe visualizations that attempt to improve people’s understanding of their email histories.

2.1.1 Themail - Visualizing Email Content: Portraying Relationships from Conversational Histories

Themail [30] is an email visualization that uses the textual content of email conversations to represent the most important aspects the user’s relationships.
For each relationship between the owner of the mailbox and a contact, the exchanged words are displayed in a timeline, with a list of words for each month and a selection of words for each year in the background, in sizes that reflect their frequency (bigger fonts for more frequent words). There is also collapsed view, which only shows months when messages were sent. Users can select a word to read the emails from which it was selected. Colored circles in the back represent email messages exchanged during each month, with size indicating the length of the message and color representing the direction of the message (incoming or outgoing). Users can search for words to highlight them. Tests with users show that two types of interaction with the system tend to emerge.

Haystack mode: looking at the bigger picture, how the relationship evolves over time, what words are being exchanged in important periods. People who use the system this way appreciate it when Themail validates something they already know. It is equated with looking at a photo-album to reflect on the past. It is more relevant for personal relationships and it was the preferred mode of interaction for the majority of the participants.

Needle mode: the desire to learn more details about the words that appear on the timeline, sometimes at very specific times, so they can retrieve information that was being exchanged during or at certain events. Users are not satisfied with a simple validation of what they already know. It is more relevant for work-related relationships.

The system presents some limitations related to content parsing: all emails are handled equally, which originates some unrepresentative words (code, signatures, jokes, etc.), and only individual words are considered, not expressions or phrases, which are often important to get the full meaning of the messages.

2.1.2 PostHistory - Digital Artifacts for Remembering and Storytelling: posthistory and Social Network Fragments

PostHistory and Social Network Fragments [29] are systems for the visualization of email histories from the analysis of sent and received email headers. Both designs are centered on the owner of the email account and on how the interactions with other people evolve over time, but PostHistory focuses on the direct interaction with individual people while SNF focuses on the emergence of groups of people within the network of email contacts.
PostHistory displays a calendar with squares representing days and rows of squares representing weeks. The size of these squares represents the amount of emails received that day and the color represents how personal or directed those messages were. It also displays the names of the contacts on another panel, either in a simple ordered table, or distanced from the user's name according to the frequency of contacts. Users can select a name to highlight the messages that contact sent, or they can select a day to highlight the names of the contacts from whom emails were received. This visualization can be animated, with different names appearing and disappearing, moving closer to or further away from the user, as the days pass. This reflects the evolution of the email relationships.

The SNF system defines different types of relationship pertaining knowledge, awareness, and trust between contacts by analyzing the recipients of emails. The roles of these contacts (work, school, personal, etc.) are also considered, being either specifically defined by the user or based on the email addresses. On one panel, the application displays a history of these relationships, with pairs of squares representing the awareness and knowledge connections of each of the time slices each year is divided into. On another panel, clusters of interlinked contact names are displayed. The size of the font represents the strength of the connection, and the color represents the role. Users can zoom in and out, and the visualization can be animated, with people appearing and disappearing as the time slices are progressively highlighted.

In addition to enabling users to perceive patterns of interaction that they had not been aware of
before, these systems can be used as a way of recalling and sharing social experiences, like a photo album.

However, in PostHistory tests, users complained about the inability to mark important days or events after locating meaningful patterns, and about not being able to access the content of emails or even just the subject to get a greater understanding of the interaction patterns they perceived. In SNF tests, users complained about the overall difficult readability of the visualization and about the network of contacts being static (contacts do not move around to show the evolution of the relationships).

2.1.3 Using intimacy, chronology and zooming to visualize rhythms in email experience

faMailiar [14] is a visualization of email exchanges that attempts to provide insight into how the social structures and the interaction patterns of users evolve over time.

The system relies on the concept of intimacy, both the user-defined contact intimacy and the procedurally generated intimacy weight of each email message. Intimacy is shown through the color and, redundantly, the shape of the symbols representing emails, except for outgoing emails, which are always represented by a star.

Email messages are displayed in a calendar with daily and weekly views of email activity. The daily view provides the most detail by showing all the emails sent and received, while the weekly view display aggregated average message intimacy weights with the aforementioned color, and number of messages with size. It is also possible to filter the messages, defining queries either by email headers or by example (when a message is selected). The user can zoom in and out to vary the amount of information displayed, from a single day up to four months, as well as pan either by dragging the background or by using the direction keys. Messages and message aggregations can be selected to view more detailed information about them.

Tests demonstrated that, by using this system, people can perceive rhythms in their email relationships, thus remembering past activities and the contacts involved faster, as well predicting future interactions with more accuracy than by using traditional email interfaces. It also allows users to gain new insight into their lives, potentially enabling them to better manage their time and investment into relationships.
2.1.4 Dynamic Coordinated Email Visualization

Mailview [8] is an interactive email visualization system that uses filters and coordination techniques to give users a sense of the time elapsed between emails.

Emails are displayed in various temporal based scatter plots that can be scaled and zoomed. Each email is represented by a glyph (vertical lines, circles, or squares). The relative size of the email is shown by the size of the glyph. Emails are automatically colored according to the folders in the archive, but users can edit these colors. They can also choose which views are coordinated together to compare different parts of the plots.

![MailView](image)

Users can zoom into any area by dragging a bounding box directly on the plot (changing both date and time range) or by dragging the mouse along an axis (zooming into either a date range or a time range).

All operations that affect the display are stored in a list, so users can undo and redo them, thus encouraging experimentation and facilitating comparisons.

Emails can be filtered and selected to detect trends about particular senders or subjects. Clicking on a glyph selects the corresponding email and fixes the current detail information. Afterwards, when users hover over a detail field such as subject or sender, only emails that have been sent by that sender are displayed. Users can then select another field and the information will be constrained by two fields.

Besides performance issues when visualizing hundreds of emails, the authors also acknowledge that users do not often remember exact dates of events but they remember periods of time, so it would be useful to explore the data through a set of aggregation commands.

2.1.5 Visualizing electronic mail

Visualizing electronic mail [27] presents a visualization system to support email browsing and querying making use of a clustering mechanism based on the hierarchical nature of the domain names present in email addresses.

In a hierarchical view of the messages based on the sender’s email addresses, messages are automatically grouped in an interface similar to the Windows explorer. Initially, only the top levels are displayed, but users can expand the hierarchies. Because the automatic hierarchization is not perfect, users can construct their own hierarchies. Besides this top level domain view, there is also a company/institution view, in which the top level domain name is ignored and the messages are grouped according to the sub-domain, and a sender view, in which the messages are grouped by the first letter
of the sender’s name. This hierarchical representation has several advantages: it presents a high level view of the mailbox; it provides automatic clustering which may facilitate the search for particular messages without having to type any query; it reduces the need to scroll down to find a message by grouping related messages together and displaying only those under the selected sub-tree; and it can be used to reduce the search space by allowing users to specify which sub-tree is relevant for a particular query.

There is also a temporal view, meant to complement the hierarchical view, in which messages are organized by time and sender in a two dimensional grid. To display messages in the temporal view, users select any node on the hierarchical view and click on the display button. All messages that appear as leaves under the selected sub-tree are displayed as small squares in the temporal window. The number of messages received from the same user during a particular time period is represented by a number at the top the message squares. Squares outlined in green indicate that the user is the only recipient of that message. The time axis can also be changed to represent years, months, days, or hours.

A preliminary usability study yielded encouraging results, although a more extensive study is said to be necessary to further evaluate the benefits of the system. The authors are considering adding speech recognition support for queries, and customization of the interface by specifying rules and colors to automatically highlight messages that fit that particular criteria. They also consider it necessary to enhance the thread recognition process to group together only the messages that are directly or indirectly replies to each other.

2.1.6 Email Visualizations to Aid Communications

Email Visualizations to Aid Communications [22] describes the development of information visualization techniques (trees, timelines, and low-resolution overviews) to highlight the relationships between messages and between people who exchange them. The authors identified three features which they consider useful in email visualization: message threads, time, and document content.

Message threads are represented by trees. Nodes are colored to represent different senders. Time is then visualized by combining the thread view with a timeline. Lines represent time frontiers and text is used to display each thread's subject.

Content of the exchanged documents can be displayed in a reduced-resolution overview. Dates are listed across the top and overviews of the messages are shown in columns below. These overviews can also display images contained in those messages. Color highlights indicate the presence of names or dates and that certain messages belong to the same thread.
The authors integrated these visualizations in a prototype email client which combines a traditional list of email messages with a vertical message tree in a timeline. The node for a selected message, highlighted with yellow on the list, is replaced with a reduced-resolution overview. Another highlight connects the messages within the thread. Further developments on this prototype, now named ReMail[11], led to a final prototype which is more similar to traditional email clients, although containing small visualization aids such as thread arcs, a more compact version of the previous thread visualization that attempts to convey the same information in a smaller image.

User studies revealed that while users may find some complex visualizations interesting, they do not use them regularly. On the other hand, smaller and simpler visualizations that complement the regular
browsing by displaying non-critical and easily understood information were widely seen as useful. Other lessons from these studies include the need to connect emails into more digestible groups (with thread arcs, for instance) and the importance of the search capability.

### 2.1.7 Bloom: An Interactive, Organic Visualization of Starred Emails

Instead of visually representing an entire email collection, Bloom [15] focuses on starred emails and displays them as growing flowers in a vase.

![Figure 2.10: Bloom.](image)

Starring an email prompts the creation of a flower with its information. The color of the flower depends on the category of each email (finances, school, etc.), which in the current state of the prototype is done manually but could conceivably be done automatically.

When users touch a flower (as this is meant for touch screen devices), it grows and its textual content is displayed above it. Touching it again brings it back down. Performing a "plucking" gesture (touch, hold, and flick) removes that flower, thus removing the email from the starred status.

### 2.1.8 Graphical Browsing of Email Data: A Usability Based Comparative Study

In Graphical Browsing of Email Data: A Usability Based Comparative Study [20] the authors describe an usability study which compared the effectiveness, efficiency and satisfaction of a standard email client to an experimental email platform with two distinct graphical representations: LinearVis and MatrixVis. Although both approaches present emails using a temporal organization, they make use of different techniques.

LinearVis presents a vertical list of email contacts, each contact followed by its own list of coloured squares representing emails (unread as yellow and replied to as green, for instance). On top, users can select a year to modify the available time periods to observe, either all emails in a year or just the ones in a single day. A second dropdown menu (not visible in the accompanying figure) lets users select other particular dates in the chosen year. The size of individual email squares shrinks to accommodate all available emails. Clicking on an email opens it to display its content and information in a main view, as well as the connection of an email to known contacts using coloured lines (green lines represent reception of said email as a carbon copy, for example).

MatrixVis keeps the main date selection mechanism used in LinearVis, but rather than presenting emails by contact, it lists the emails contained in six four hour segments within each selected period. Email squares are identical to the ones in the previous technique, but now have a fixed size, meaning that temporal segments may vary in size. Emails are listed in rows with up to eleven columns, forming
the the matrixes. Again, clicking an email displays it in the main view, but it does not show its connected contacts in a side view. Instead, passing the mouse over an email will display its sender together with a matrix of other emails sent by that contact in the side view.

The usability tests showed that the graphical presentation of emails used in LinearVis improved the effectiveness of email clients in terms of successful tasks, as well as inferior number of actions and time to perform them, while the large degree of information hiding done in the MatrixView visualization had the opposite effect, to the point of being less effective than typical email clients.

2.2 Instant Messaging

Instant messaging is a very common communication system these days. Due to its real time nature and instant feedback, messages tend to be short and informal, but, given their constant use, they can be a valuable source of personal information. However, there are very few applications focused on visualizing conversational histories from a personal point of view. Two systems that do are described next.

2.2.1 CrystalChat: Visualizing Personal Chat History

CrystalChat [28] is a visualization of personal chat history. Information taken from MSN Messenger logs is displayed and navigated in three dimensions. The conversation history for each contact can be seen as lines of circles coming out of the user in the center. A line is a conversation, and a circle is
a message. The color of the circle identifies the sender, with darker colors revealing larger messages. Circles of different colors alternate according to the order of the messages. Viewed from above, there is a line for each contact.

From the side, each of those lines is revealed to be several lines, from the earliest conversation at the bottom, to the most recent at the top, optionally with gaps indicating actual time between conversations. From this view, there is a semi transparent plane between the contact in focus and the one behind to improve readability. The color of that plane can be altered to show the tone of the conversations according to the use of emoticons and punctuation. Conversations can be expanded to show their content.

Using this system, several patterns can be identified: who starts most conversations, who speaks the most, what contacts have not been contacted for a long time, etc. This allows people to reflect on the past and their relationships with their contacts, perhaps even prompt them to change their behavior towards a contact if they notice undesirable trends.

While CrystalChat may be adapted to use data from other types of messaging applications, it is limited to smaller messages, which means email visualization, for instance, would be complicated. The system does not scale very well because long conversations and large conversation records make the resulting structure tedious to navigate.

2.2.2 Egocentric Analysis and Visualization of Instant Messaging Activity

*Egocentric Analysis and Visualization of Instant Messaging Activity* [1] describes the development of an instant messaging visualization system that attempts to reveal temporal communication context cues within conversations. The system is based on the detection of features such as frequency, dominance, density, longevity, and the dominant semantic concepts.

Contacts are displayed around the user’s name at the center. Both the contacts and the user are represented by a circle and labeled with a name. The size of a contact’s circle indicates the frequency of
communication while the color is used to differentiate between different groups. Alternatively, a picture may be displayed instead of the circle.

Conversations between the user and each of its contacts are represented by dots along the lines between them, with the most recent conversations being closer to the user. These dots can be clicked on to display the actual conversations in a text box to the side. The period of time displayed in these lines can be modified with a slider. When users select a conversation, tags pertaining to the most representative words of that conversation are displayed next to the text box.

Dominant concepts extracted from all the conversations are displayed at the top left corner. Just above it is an indication of how much the users talks compared to a selected contact.

Finally, there is the ability to search conversations based on their tags. Conversations that match those tags are displayed in individual text boxes. Contacts corresponding to returned conversations are highlighted in blue.

The authors conducted an informal user study that revealed that people found the system interesting and enjoyable to use, but they noticed some flaws and suggested improvements: the extracted concepts were often too general and uninformative; it would be interesting to extract files and web links from the conversations and display them in some way; the search mechanism should be improved; and the ability to view emotions extracted from emotional words and emoticons would also be appreciated.

2.3 Text Documents

Text documents account for a considerable part of our computer productivity. While the things we write or read about may not necessarily be of interest to us beyond work obligations, even that is part of our identity. The following papers describe systems that attempt to give an overview of document collections as well as provide ways to search and browse them.

2.3.1 Gist icons: Seeing meaning in large bodies of literature

Gist icons [5] is an interactive visualization of text documents that tries to provide a way for users to easily refine their search for information by visually representing the natural language algorithms used to retrieve documents based on their content.

Each document is represented by a shape: a histogram of word weights displayed around a point, with peaks and valleys showing the frequency of words in that document. Documents with similar content have similar shapes, documents dealing with a specific word have a peak in the same part of the shape,
etc. 50 to 100 documents can be shown at a time, grouped together according to their shapes, which helps users detect patterns and common themes among documents. A shape representing the average weights for the result set is also shown. Single words can be viewed in vertical fish-eye display next to the collection of shapes. This display can be scrolled and, as that happens, each word is highlighted in all the document shapes along with a red circle. By observing the size of these circles, the user can quickly see which documents contain words of interest.

The system takes advantage of people’s ability to perceive visual patterns to allow for a faster understanding of the contents of several documents than it is possible by reading the actual text. However, it still lacks the option to specify queries based on shapes in combination with the ability to manipulate shapes to suit them to specific interests.

2.3.2 ThemeRiver: Visualizing Theme Changes over Time

ThemeRiver [12] is a document visualization that displays thematic variations over time across a collection of documents in an attempt to facilitate the identification of trends, patterns, and unexpected occurrences and non-occurrences of themes or topics.

Each theme is represented by a colored “current” that flows through a horizontal timeline, widening and thinning to indicate its presence in the document at each moment. Currents maintain their integrity over time. If a theme disappears, it will have the same color and be in the same relative position to other currents when it reappears.
Users can hide or display topics, event labels, time and event grid lines, and raw data points. They can also choose alternate line drawing algorithms for the currents and river. Hovering the mouse over a topic displays associated time or topic. The visualization can also be panned and zoomed.

The authors believe the river metaphor is familiar and easy to understand, requiring little effort on the part of the users to interpret the visualization. Compared to simple histograms, in which bars represent the strength of themes in discrete moments, the river metaphor has the advantage of being continuous. However, the data is not continuous so it has to be interpolated. If the user zooms too closely, the representation may not be accurate. On the other hand, if the user zooms to far out, there is the problem of having to much data. This is solved by combining time slices. The amount of time per time slice is increased and weights are combined. This maintains an acceptable level of accuracy while maintaining performance. Still, the authors are investigating faster and more efficient drawing algorithms.

Usability tests comparing ThemeRiver to a histogram created with a spreadsheet revealed that the users found the system easy to understand and useful for identifying macro trends. However, they thought it was less useful for identifying minor trends because the curves tend to hide very small values. Users considered that the connectedness of the river helped them follow trends more easily than in the histogram and that they would rather use ThemeRiver than a histogram. Still, there were features from the histogram, such as the ability to see numeric values, that users would have liked to see added to the ThemeRiver. In fact, some users found the histogram to be more trustworthy because bars represented exact values. And although the abstraction away from individual documents was appreciated, users would have liked to be able to access the text of each document on any time period, as well as selecting a current to see which documents contributed to it.

2.3.3 DocCube: multi-dimensional visualization and exploration of large document sets

DocCube [16] is a system for the global visualization of large document sets that helps users form the appropriate queries for their information needs and access the corresponding documents.

The system makes use of concept hierarchies or ontologies, each corresponding to a facet of the documents, which are then seen as dimensions to structure and visualize document collections. In the case of scientific monitoring, for example, DocCube can provide global information such as the number of publications per author or per topic, the relationship between authors and topics and the strength of these links and their evolution over time.

Users are guided to express their information needs according to the language of the domain and never lose the semantic context of their current query or interest formulation, so they can refine their search in an interactive way.

To begin the visualization, users have to select up to three dimensions that describe the information space they desire. Users are then shown the top level of the dimensions or hierarchies which they can browse until they reach the desired level of detail.

The three-dimensional representation displays a global view of the documents related to the concepts chosen by the user. The axes represent the dimensions and the spheres represent the number of documents that have been categorized in the corresponding dimension value. Users can change the level of aggregation of the data to either get more general or more detailed information. The content of the documents can be accessed by selecting one or several nodes in the cube representation. There is also a slice function to obtain a two-dimensional view set to one of the dimensions.

Queries can be formulated by selecting one or several spheres in the cube. As a result of the selection, the document references are displayed in a ranked list. The terms corresponding to the query are highlighted and are displayed in the search result windows at the same time as the corresponding
document references.

2.3.4 Visualizing the Non-Visual: Spatial analysis and interaction with information from text documents

Visualizing the Non-Visual: Spatial analysis and interaction with information from text documents [31] describes the Multidimensional Visualization and Advanced Browsing project for researching visualizations of text document collections.

The first developed visualization consists of a two dimensional scatter plot representation of document, clustered according to similarity. This galaxy view allows users to quickly gain an understanding of the fundamental topics. Besides exploring clusters to discover deeper thematic groups, users can also partition the visualization into temporal units, slicing the database to explore temporal windows that can reveal connections between the emerging topics and external events of the time.

Eventually, ThemeScapes, a system that displays document clusters in three dimensional landscapes, was developed. Documents are clustered according to thematic content, elevation represents thematic prevalence, and different terrain features represent relationships between documents and their themes. This allows for a faster understanding of the whole document collection. Analysts using ThemeScapes reported significant time savings and increased comprehension of the studied data set.
2.3.5 Info Navigator: A Visualization Tool for Document Searching and Browsing

Info Navigator [2] is a system for searching and browsing sets of documents, making use of both a standard full-text search engine and the computation of keywords from the most relevant features of the document set. These keywords are used to reduce the dimensionality and to improve clustering. Besides a plain list display of the search results, the system provides three different visualizations of the data: Sammon map, Dendro map, and a radial view.

The Sammon cluster view uses a Sammon map (mapping a high-dimensional space to a lower dimension) to generate a two dimensional location from a many-dimensional vector of cluster centroids. Each cluster is represented by a circle labeled with its most frequent keyword and with a radius that indicates its size. The distance between circles is an indication of the similarity of their respective clusters. When the mouse hovers over a cluster circle, a pop-up menu appears. The first item in this menu displays the number of documents in that cluster, and clicking it shows a table of cluster keywords to the left and a list of cluster document links and snippets at the bottom. Users can select keywords on the left to filter the bottom list to display only the documents containing those terms. The "select" item in the pop-up menu flags a cluster as selected. The "drill down" item redisplay the visualization with only the documents of the selected clusters. The other menu items are four significant keywords and are not selectable. This view is useful to quickly weed out irrelevant clusters and drill down into one or more relevant clusters.

The Dendro Map visualization represents documents as leaf nodes of a binary tree according to the same clustering algorithm as before. Different colors indicate individual documents and clusters. The most frequent keyword of the sub-cluster (or document) is printed next to each of lowest level nodes. When users hover over an internal node, its color and that of its associated subcluster branches change from light blue to dark blue while the leaf nodes (document representations) turn bright red. There is also pop-up menu. In this case, the "drill down" option causes the root node of the visualization to be replaced by the selected node. This view allows users to see different levels of the cluster hierarchy so they can quickly narrow down their search to a small subset of documents.

The radial visualization uses the keywords to differentiate the documents. The twelve highest ranking keywords are displayed in a circle and the documents in the search set that contain those keywords are placed within the circle, closer to the most representative keywords. When the mouse hovers over a document node, a bubble displays a descriptive text from the document. To mitigate the ambiguity that dimensionality reduction can introduce, users can click on a document node to highlight the keyword
nodes that affect its location. Selecting a keyword node highlights the nodes of the documents which contain those terms. Dragging a keyword node away from the edge of the circle with the mouse causes all documents that contain this keyword to follow the movement of the keyword. Users can create manual clusterings by dragging several keyword nodes together. Users can also zoom in and out. The problem with this visualization is that it becomes difficult to read when there are too many keywords, which is something the authors believe is solved by the two previous visualizations. This view is useful for a more experimental approach, due to the ease with which users can control the clustering of groups by interactively moving keywords.

Tests with users gave the authors evidence that the described clustering mechanisms and visualizations can be effective for searching and browsing documents.

2.3.6 Jigsaw: Supporting Investigative Analysis through Interactive Visualization

Rather than focusing on themes or concepts within the text documents, Jigsaw [26] identifies entities (person, place, date, and organization) and displays the connections between them (two entities are connected if they appear on a document together). The prototype provides several visualizations as different views of the same document collection.

The list view lists entities (alphabetically or by frequency of appearance) on the left and on the right
Figure 2.23: Jigsaw. From left to right and top to bottom: the list, graph, scatter plot, and text views.

and draws connections between them. Users can select several different types of entities on either side. For instance, they can observe the connections between people on one side and places plus dates on the other side. Clicking on an entity highlights it in bright yellow and all connected entities are highlighted in a shade of orange, with brighter highlights standing for stronger connections (more appearances together). Scroll bars are used to mitigate the problem of displaying large amounts of entities, but users can also select an option to move the entities connected to a selected entity further up on the list.

The graph view displays reports and entities as interconnected labeled circles. The graph is not all displayed at once, instead, users can expand or collapse entities or documents to show or hide the entities or documents which in turn are related to them.

The scatterplot view represents bidimensional relation between entities by showing the documents in which two entities appear together in a cluster. The entities in each axis can be selected in a similar way it is done in the list view. When two entities are connected, a labeled diamond is displayed in the intersection. The same diamond can appear in multiple positions in the scatterplot depending on the connections. To counter the emergence of clutter with large collections, users are allowed to zoom in and out by moving range sliders on each axis. The color of the documents’ diamonds can also be persistently modified.

Finally, the text view displays the actual text of the documents with added highlights on identified entities. Clicking one of these entities on the text view sends that selection over to the remaining views. That same level of interoperability exists between each one of the four views.

Using these visualizations together, one can get an overall view of the entities contained in an entire collection of documents and then interactively explore that information to make sense of the available content and guide further analysis of individual items.
2.4 Multiple Sources

The information that people access on their computers can come from multiple sources and exist in various formats. However, each file type is usually tied to a particular application, so, if they wish to search and browse their information, people either jump from program to program and access different kinds of visualizations of specific document collections, or they just inefficiently browse folders and directories. The following papers describe systems which attempt to provide a unified interface for searching and browsing information from multiple sources.

2.4.1 LifeLines: Visualizing Personal Histories

LifeLines [18] is a system for the visualization of personal history records that provides a general visualization environment for the information contained in the multiple sources that compose those records. Data records are displayed on an overview screen using timelines. Continuous events with variable status are represented by lines on the timeline, with color and thickness representing different properties. Discrete events are represented by icons. Data can be accessed directly from this overview and relationships between the various data can be highlighted. To represent all the information at the overview level without relying on scrolling when all lines and labels cannot be seen, the lines are brought closer to each other and the labels disappear. In this silhouette view the relative importance and distribution of the data records is still visible. It is also possible to use hierarchies that can be expanded and collapsed to show different levels of detail.

LifeLines facilitates the observation of complex and extensive data records. Seeing the information clearly displayed on a timeline reduces the chances of missing information, aids in the detection of anomalies or trends, and streamlines the access to details. It can also be adapted to various applications.

However, the described prototypes leave too much space unused and do not have mechanisms for data entry (appending or correcting existing records). The authors also mention the difficulty in designing appropriate data encoding schemes for this sort of system, as choice of icons, color, and thickness can introduce bias.

2.4.2 MyLifeBits: Fulfilling the Memex Vision

MyLifeBits [9] is a system to save, organize, and display multimedia content. Its authors’ aim is to fulfill the Memex vision of storing and efficiently accessing a person’s entire collection of documents,
while also extending it to better support multimedia data types. The system is essentially a database of resources (that can be annotated by any number of other resources), and links (that indicate one resource is annotated by another).

It was developed with four principles in mind: organization based on collections and search; multiple visualizations; easy annotations; and authoring via "transclusion" (including part of a document in another document through two-way links).

Query results can be visualized in multiple ways: detail, with a list of resources and their properties; thumbnail, with miniature images of the resources in a grid; timeline, with thumbnails on a linear time scale; and clustered-time, with thumbnails clustered by similar time.

In all views, the authors attempt to provide maximum information density in order to avoid the need to perform extra clicks or even open new windows to display all the necessary information. For instance: the representation of a collection reveals how many items it has (either by text in the detail view or graphically in the thumbnails), thus saving the user from having to click that it to find out if it is empty or not; hovering the mouse over a thumbnail prompts the display of a bigger thumbnail, facilitating the understanding of what that item is; and there are optional windows to show various information.

The system also has a story creation module that lets users make queries and drag-and-drop items from the result into a story. There are two story types: slide show, a sequence of images with audio and captions which also allows the user to click an image at any time to see what resources it is connected to in a separate window; and time sheet, a composition of multiple timelines which can be scrolled together to allow for comparisons. Stories are stored as annotations linking to all the resources they include.

Initial experiences with MyLifeBits were successful, as it appears to be a very useful memory aid. Still, it has scaling and performance issues which have to be resolved to accommodate the large quantity
of information people store over their lives.

2.4.3 TimeMachine Computing: A Timecentric Approach for the Information Environment

TimeScape [19] allows users to visit past and future states of a computer. When users need to see a document on which they were working at a given moment, they can travel in time and restore the status to that moment. TimeScape provides different views of the information space: desktop, timeline, and calendar views.

![Figure 2.27: TimeScape.](image)

The desktop view is similar to the usual computer desktops, but with additional enhancements to facilitate the visualization of temporal information. For example, the color of Post It notes attached to the desktop gradually changes over time to indicate its age, and the background color of desktop surface changes to indicate whether the system is in the current, past, or future mode. The transparency level of the background is also controllable, so that past (or future) information can be seen from the current time. The left and right edges of the desktop also indicate the current time and date. When a desktop item is placed near these edges, its duration appears on a time band.

The timeline view represents desktop items as horizontal lines on a timeline. The left and right endpoints of these lines respectively represent the dates of object creation and deletion. The current desktop is visualized as a semitransparent slanted rectangle in the middle of the screen, and the left and right parts represent the past and future of the desktop. The user can use zoom in and out to change the timeline scale to browse activities from a day, a week, a month, or an entire lifetime. The labels of objects that have shorter durations fade away when the screen is zoomed out.

The calendar view displays items in a calendar. Each cell contains items created on the date of the cell. This view is suitable for browsing a schedule and appointments.

These views are switched with a smooth animation to show the relationships between them, and the combination of these views helps a user recall the contexts of past activities. For example, users can travel to a specific time in the desktop view and then switch to the timeline or to the calendar view to see what they were doing around that time.

Informal observations revealed users found the system useful and enjoyable, although they found some difficulties in the manipulation of desktop items.

2.4.4 Lifestreams: an alternative to the desktop metaphor

Lifestreams [7] is a system that organizes documents in a time-ordered stream instead of the conventional files and directories.
All documents (pictures, emails, papers, etc.), both created by the users and sent to them by other people, are stored in a stream. At the beginning of the stream there are documents from the past, in the middle there are documents from the present, and at the end of the stream there are the documents from the “future” (reminders, calendar items, to-do lists, etc.).

![Figure 2.28: Lifestreams.](image)

Documents are displayed in a sequence of overlapping rectangles. Contents can be examined by hovering the mouse over each rectangle. Beneath this structure there is a scroll-bar with which the user can go back and forth in time. Color and animation are used to indicate document features: unread documents have red borders; writable documents have thicker borders; open documents are shown offset; incoming documents are shown sliding from the left side; and newly created documents appear from the top and push the stream back, which means old documents eventually move out of the view. Users can also create sub-streams of documents by defining queries over the main stream (unanswered emails, for example). Documents that match the search criteria are dynamically added to this sub-stream.

The authors believe that this system is more fluid and natural than the traditional desktop metaphor, and that it successfully reflects the way people work.

### 2.4.5 Stuff I’ve Seen: A System for Personal Information Retrieval and Re-Use

Stuff I’ve Seen [6] provides a unified index of information from multiple sources (emails, web pages, documents, media files, etc.).

The indexed documents are shown in a scrollable list view. The visualization does not go beyond icons and textual descriptions. Either at the top or to the left of this list there are filters with which users can specify queries and manipulate the results. Instead of choosing search parameters and then pressing a search button, every time the user checks a filter that particular query is performed. This provides an iterative and interactive search experience that allows users to start with broad searches and then continuously refine their queries and sort the results.

Double-clicking on an item opens it the appropriate application. Right-clicking displays a context menu that allows the user to go to the folder containing the item or to the cached version.

Initial tests reveal that users can find information more easily using Stuff I’ve Seen, and that they tend to use other search tools less frequently when this system is available.
2.4.6 Milestones in time: The value of landmarks in retrieving information from personal stores

Milestones in time: The value of landmarks in retrieving information from personal stores [21] describes a system for the visualization of personal content search results that relies on users’ episodic memory to search and retrieve information. It makes use of the indexing and search system in Stuff I’ve Seen, described above.

For that effect, search results are displayed in a timeline next to important events, called landmarks, both for the general public (holidays and news headlines) and for the user’s personal life (appointments and photographs). These landmarks serve as memory cues.

The visualization provides an overview timeline from where a detailed view is expanded to the right. The overview displays the distribution of retrieved events, with more recent events at the top. The user can scroll through the highlighted portion of the overview to view the events from that segment of time in detail. The detailed view is divided into landmarks on the left and the actual data (documents, emails, etc) on the right, displayed according to the last time they were modified. Data is represented by filenames (or subjects, in the case of emails) and respective icons. The granularity of the dates (hours, days, months, or years) depends on the level of zoom. When the user hovers the cursor over a search result, a pop-up with more detailed information about the item appears. Clicking on a result opens the item in its respective application.
User tests revealed the system containing landmarks allowed for faster searches than an interface that displayed only the dates. In addition, participants generally considered the vertical, time based visualization of results useful, although some users found it confusing and difficult to navigate the search results by scrolling the selection of the overview timeline. The option to reverse the presentation order of the items was also missed by some.

2.4.7 Fast, Flexible Filtering with Phlat - Personal Search and Organization Made Easy

Phlat [4] is a system for personal information search and organization that attempts to take advantage of powerful search features so that users can reliably and intuitively find their content.

The interface has three main areas: the query area, the filter area, and the results area. The query area has query controls and a query box for entering and displaying queries. Queries can be typed by users or selected from filters. Each new query is placed on top of the previous query in the query area. By integrating filters in the query area, they are made more visible to users, who have to look at all active filters when adding a new query, thus avoiding forgotten filters that disrupt iterative queries, a typical problem in many search systems. Users can click on queries to remove or edit them, or to summon a context menu with which they can change filters into exclusions (from IS to IS-NOT), a change that is reflected in red coloration of the corresponding button. The filter area has a set of buttons that can be expanded to reveal each filter’s properties.

The results area displays documents in a list view with small icons and a textual description. The title of each document in this visualization is the actual filename, rather than any existing Title tag, since the authors consider the former to be more recognizable. The textual description contains a snippet of the document’s content, which can be hidden to make room for more results. Documents can be dragged, dropped, opened and deleted. The parent folder of a search result can also be opened, giving access to collections of documents. Results can be used to refine or replace the current query according to their properties.

User feedback revealed people appreciated the system, particularly the ability to search all of their personal documents in one application and features such as the option to refine searches using individual search results. Users reported continuous use of Phlat in their daily work. They made several requests, such as visual previews of the documents, integration with other applications, and extending searches into the contents of individual documents. Users also complained about the fact that changes in documents take time to propagate to the index, and about issues with automatic tagging and the inability to apply tags outside of Phlat.
2.4.8 Personal chronicling tools for enhancing information archival and collaboration in enterprises

Personal chronicling tools for enhancing information archival and collaboration in enterprises [13] describes four personal chronicling tools to support information archival and retrieval: event monitoring; interactive annotation; browse and search; and edit and publishing.

The event monitors run on user client devices and automatically capture user events such as emails, web pages browsed, instant messaging sessions, and edited documents (new sources can be added in as plug-ins), while the event annotator allows users to manually tag and annotate documents.

The event browser provides semantic search and the ability to follow threads. Documents are displayed in a typical list view with icons and textual descriptions. Search terms are highlighted in red and a blue icon indicates which documents were tagged by the user. There's a text box to insert search terms and a "More" button which opens a sub-menu with several filters as well as buttons to save and load searches. The publishing tool enables the publication of events by simply copying and pasting them to the appropriate groups of people.

The authors believe the system facilitates the process of continuous archival and retrieval of personal information.

2.4.9 FacetMap: A Scalable Search and Browse Visualization

FacetMap [25] is a system for the visual searching and browsing of large databases. It can be applied to heterogeneous data sets with arbitrary facets and with a wide range of sizes.

The attributes of the dataset items are grouped into facets that are then used to filter the information. Facets are displayed as labeled bubbles in a scalable, space-filling visualization according to the distribution of attributes among the items in the data set, with more space being given to facets with more items. If enough space is available, data items are displayed inside each facet. If not, items are grouped
into different labeled bubbles which also display the size of the group, together with a "More..." bubble to show other groups of that facet if they do not fit. If not enough space is available for that either, the facet just displays the number of items it contains. Larger screens can display more information. Users select facets to add them to the filtered region of the visualization. As more facets are selected, that region gets larger and the data set gets smaller, devoting more screen space to items of interest. Conventional text queries can also be added as filters. Changes are animated to help users understand how they affect the data set. Hovering over an item displays a pop-up with information and double-clicking it opens it.

User tests comparing FacetMaps to a text based system reveal that FacetMaps is slower (but not discouragingly so) for targeted search but comparable in speed for browsing. Still, some users found the facet mechanism confusing and, sometimes, specific items were hard to locate even though they were visible somewhere on the structure, leading some users to express their desire for a more straightforward list of results.

### 2.4.10 Feldspar: A System for Finding Information by Association

Feldspar [3] is a system that allows users to find personal information in their computers by interactively and incrementally specifying chains of associative queries.

![Figure 2.34: Feldspar.](image)

The interface consists of three main areas: the navigation bar, that lets users switch from one screen, containing a query, to another; the query area, where users select elements sequentially to create meaningful associations (a folder related to a file, which is related to an email, sent by a specific person, for example); and the results area, where the desired elements matching the specified associations appear.

Feldspars allows users to efficiently find documents based on a combination of known facets. However, though it presents an aesthetically coherent and direct graphical representation of the queries, it does not provide a visualization of the content of the entire information space under analysis.

### 2.5 Discussion

Given the large number of different visualizations, it is important to have a structured way to compare them. Understanding what features are prevalent and which ones are rare or even completely absent can provide important directions for future developments.
Table 2.1: System comparison in order of appearance in this document.

<table>
<thead>
<tr>
<th>System</th>
<th>Overview</th>
<th>Zoom</th>
<th>Filter</th>
<th>Details on demand</th>
<th>Content representation</th>
<th>Information organization</th>
<th>Sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Themail</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>keywords</td>
<td>timeline</td>
<td>email</td>
</tr>
<tr>
<td>PostHistory</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>calendar</td>
<td>email</td>
</tr>
<tr>
<td>SNF</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>cluster (relationship)</td>
<td>email</td>
</tr>
<tr>
<td>iMailiar</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>intimacy</td>
<td>timeline</td>
<td>email</td>
</tr>
<tr>
<td>Mailmerge</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>cluster (time)</td>
<td>email</td>
</tr>
<tr>
<td>Visualizing</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>timeline</td>
<td>email</td>
</tr>
<tr>
<td>Email Visualizations...</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>tree/ timeline</td>
<td>email</td>
</tr>
<tr>
<td>Bloom</td>
<td>yes (limited)</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>organic/ naturalistic</td>
<td>email</td>
</tr>
<tr>
<td>LinearVis</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>timeline</td>
<td>email</td>
</tr>
<tr>
<td>MatrixVis</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>matrix</td>
<td>email</td>
</tr>
<tr>
<td>CrystalChat</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>mood</td>
<td>radial</td>
<td>instant messaging</td>
</tr>
<tr>
<td>Gist icons...</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>word frequency</td>
<td>histogram</td>
<td>instant messaging</td>
</tr>
<tr>
<td>ThemeRiver</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>keywords</td>
<td>river</td>
<td>text documents</td>
</tr>
<tr>
<td>DocCube</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>keywords</td>
<td>cluster (similarity)</td>
<td>text documents</td>
</tr>
<tr>
<td>Themescape</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>cluster (similarity)/ topological map</td>
<td>text documents</td>
</tr>
<tr>
<td>InfoNavigator</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>keywords</td>
<td>cluster (similarity)/ tree/ radial</td>
<td>text documents</td>
</tr>
<tr>
<td>Jigsaw</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>list/ graph/ scatterplot</td>
<td>text documents</td>
</tr>
<tr>
<td>LifeLines</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>timeline</td>
<td>multiple</td>
</tr>
<tr>
<td>MyLifeBits</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>timeline/cluster</td>
<td>multiple</td>
</tr>
<tr>
<td>TimeCape</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>timeline/ calendar</td>
<td>multiple</td>
</tr>
<tr>
<td>Lifelines</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>stream</td>
<td>multiple</td>
</tr>
<tr>
<td>SIS</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>list</td>
<td>multiple</td>
</tr>
<tr>
<td>Milestones in time</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>timeline/ landmarks</td>
<td>multiple</td>
</tr>
<tr>
<td>Partial...</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>list</td>
<td>multiple</td>
</tr>
<tr>
<td>FaceMap</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>bubbles</td>
<td>multiple</td>
</tr>
<tr>
<td>Feldspar</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>connections</td>
<td>multiple</td>
</tr>
</tbody>
</table>

Following Shneiderman's Visual Information-Seeking Mantra [24], systems were compared in their ability to provide overview, zoom, filter, and details-on-demand. Also of interest are the ability to represent the content of individual items, the metaphor used for information organization, and, finally, the sources of information they handle.

All applications provide an overview with which users can get a general understanding of all the information. The ability to zoom in, to increase detail on an element or section of the visualization, and to zoom out, to observe the surrounding context, is missing in a considerable amount of systems. Filtering, however, is available in most applications. Details-on-demand, the ability to view an element in greater detail without changing the overall representation and losing context is also present in most systems. It is worth noting that the related ability to access the actual content of individual items, while absent only in a few systems, was often mentioned by the users as an important feature those systems lacked.

Content representation, in the sense of analyzing the content of data collections and displaying its most relevant features, is done in very few applications. Many systems, specially email visualizations, are capable of showing temporal or communication patterns, but they rarely try to represent patterns in the content. This may be because most applications are aimed at searching and browsing, rather than attempting to highlight interesting patterns. We can only clearly see this sort of content representation in Themail, Gist icons, ThemeRiver, and in a more limited way in a few other systems.

One of the most common metaphors for information organization is the timeline. Interesting alternatives for temporal organization include the calendar, the river, and the stream, as well as the addition of landmarks to the timeline. Spacial clustering is also very common, usually attempting to group elements according to some measure of similarity. Trees are often used to display the hierarchical relationship be-
tween data items, as seen on common email threads representations. FacetMap, however, managed to represent the organization and hierarchies of facets with a novel “bubble” interface. Still, a few systems did not attempt to experiment with anything more than simple list displays.

With all this in mind, we can establish a number of features which should be included in a personal information visualization system, besides the obvious ability to handle information from multiple sources. VisMe would not be complete without overview, zoom, filter, and details-on-demand, as these features are important for an effective interaction with information visualization systems. Content representation is also a desirable feature for VisMe. It is something that few systems provide but which Themail shows can be done effectively with emails. A similar level of content representation can easily be applied to numerous sources from which representative keywords can be extracted. Finally, considering all the surveyed organization methods, timelines appear to be the best choice. Simple to read and understand, timelines have been successfully used to display evolving patterns in the content of emails, as seen in Themail, and they obviously also provide temporal context, which has been shown [10] to be very important when dealing with personal information.
Chapter 3

Proposed Solution

In the context of this work, personal information can be understood to be contained in the documents one usually accesses and interacts with in regular computer use, in the wider sense of text documents in several different formats, websites, instant messaging logs, and emails. Attempting to extract and visualize information from all the aforementioned sources brings about the issue of heterogeneity. At the same time, all sources contain common related properties that can be used to integrate them in a single, all encompassing visualization. Our solution for the interactive visualization of personal information tries to solve the problems and take advantage of the opportunities presented by the targeted sources.

The development of VisMe’s visualization technique and interface was carried out through iterative sketching and prototyping. We aimed to achieve three main goals when designing the interface: a uniform representation of the content of heterogeneous document collections; a simple and uncluttered interface; and the possibility of exploring information in context rather than displaying individual queries by themselves. This posed a series of challenges and conditioned the development of the solution.

3.1 Facets

As discussed in the analysis of the related work, time has been shown to be extremely important for the memorization and recall of personal information. Additionally, all document types under consideration for this visualization have some sort of temporal presence. As such, time is a straightforward choice.

From the beginning, we identified the content of the textual documents as something we had strong interest in visualizing. As such, and again considering the content analysis of the document types we are working with, we chose to represent the most significant keywords extracted from each document according to their tf-idf [23] weight.

It is also noticeable that many of these documents exist within a context of interpersonal communication; we decided it would be equally relevant to display contacts, the different people who create, send, or receive the documents that make up a personal collection.

Finally, we need to identify the documents themselves, so the name or title of each document is also integrated into the visualization.

Most of these facets can be reliably extracted from all documents, with a few exceptions: some documents may simply have no defined creator (though if they are attached to emails they can still be connected to the participants); others may come from the same contact, but under different aliases that the underlying indexing system is not able to connect. Still, there is generally still enough information available to create a useful visualization from these facets.
3.2 Organization

We have already acknowledged the importance of time, as well as its prevalence in the information we are contemplating. Thus, we have chosen to organize information using timelines.

Besides being applicable to all the chosen facets, timelines have been shown to be effective in presenting the content of document collections in a way that allows for the detection of personally relevant trends and patterns, as observed in the aforementioned Themail visualization.

Although timelines were considered from the beginning of the development process (figure 3.1), we also studied other possibilities, such as word clouds (figure 3.2). The idea was that users would first choose a facet to visualize and then an organization method. Because of the importance of time, the universal applicability of timelines, and the desire to maintain a minimal and simple interface, these other organizations where discarded.

In VisMe, the most significant elements (keywords, contacts, and documents) appear larger and at the bottom of the timeline (Figure 3.3), with less relevant elements appearing smaller and further up. Font size is kept between fixed minimum and maximum values to assure legibility. The granularity of the timeline can be controlled by clicking on the times at the bottom (to zoom in; for instance, from months to days) and bottom left (to zoom out).
3.3 Exploration

The information to be displayed in the timelines is not selected through a traditional process of search terms and filters. Rather, we allow users to observe timelines containing one of three facets (together with time) as they relate to any particular chosen element.

The visualization starts with a single element, "ego", representing the owner of the information. From this element, users can expand timelines with keywords, contacts, and documents by clicking or dragging from the respective icons ("k", "c", and "d") at the bottom of the word. The same action can then be applied to any element in any timeline (Figure 3.4). These timelines contain all the elements of a chosen facet that are related to the selected element. For instance, we can extract a timeline with the names of all the documents containing a particular keyword, followed by the contacts related to one of those documents, followed in turn by the contacts related to one of those contacts, etc. There is, potentially, no limit to this; users are allowed a progressive and free exploration of any element in the visualization.

3.4 View and Interaction

Since we allow users to expand timelines sequentially from any element in the visualization, we considered implementing two alternative solutions: limiting the view to a single timeline at the time, or giving the users control of the view and allowing them to expand interconnected timelines in an open canvas.

The first option was also the first to be examined. Because of our goal to allow access to the overall context, we considered the possibility of going back and forth through the browsing history, but
we concluded that it would not suffice, as it severely limits the view of the surrounding context and discourages branching exploration.

Understanding these limitations, we devised an alternative in which the same timelines can be maintained in a common canvas, which users can control by translating, zooming, and rotating, and in which the exact positioning and orientation of the timelines can be modified. This allows users to perform both linear and branching explorations and keeps a greater amount of information on screen at once, with the option of focusing on single timelines or panning out to reveal previous or parallel timelines. While it requires a greater amount of skill to manage both the timeline structure as it evolves and the view to encompass it, this has the benefit of allowing users to freely explore information while keeping the overall context, which they can then use to interrelate and interconnect the results of their ongoing research.

As they have control of the view, users can either click on one of the icons to open a perpendicular timeline, or drag and drop a timeline to any position and orientation, as indicated by a line that is drawn on the screen while the action is performed (figure 3.5).

![Figure 3.5: VisMe, a new timeline being dragged out of the initial element in contrast to the perpendicular timeline extracted before it.](image)

To control the view, users can use the mouse to pan (dragging the left mouse button over empty space), rotate (dragging the right mouse button), and zoom in and out (using the mouse wheel and, alternatively, the Page Up and Page Down keys) of the view to observe as much information as they want. Still, we consider the option to focus on a timeline important, so we allow that by double left clicking on a timeline, which gradually and smoothly changes the view to encompass only the timeline in question. Double left clicking on an empty space will change the view to encompass all timelines.

This automatic view control can be extended to become constant, meaning the view will automatically encompass the entire visualization structure after every action that modifies it. This option can be turned on and off at any time by clicking a small button on the top right corner of the screen.

3.5 Highlights

Since the shape of a word alone may not be enough for an optimal appreciation of the evolution of an element along the timeline, we added the possibility of highlighting any instance of an element throughout the visualization by placing the mouse over that element (figure 3.6). This changes color from the regular black to one chosen by the user.

Color selection could be done by presenting the user with a separate palette, either with a minimal selection of colors in a series of squares or with a more fully featured color selection widget, complete with numeral definition and a color wheel. Instead, we developed a different, more minimalistic solution. The position of the mouse along the length of the element determines the exact color, within a range from...
red to blue and with green in the middle, sparing the need for any additional interface elements. A single left click fixes the color and a second left click undoes that action. This method makes it somewhat more difficult to select specific colors, as users have to actively search for different colors by moving the mouse over the element. This can be especially problematic if users find the need to color several elements with the same color to delimit a group of interest. However, extreme and middle positions provide at least three clearly distinguishable colors, and users can copy colors by right clicking on a colored element, followed by its application on any element by left clicking on an uncolored element afterwards. Right clicking on an empty space will clear this selection.

3.6 Text Search

Finding particular keywords, contacts, or document titles in any significantly large collection of documents would be unfeasible without a straightforward search mechanism. Keeping with the minimalistic design of the interface, simply pressing a character on the keyboard prompts the display, on the top left corner of the screen, of the search string as it is written, an indication of the number of search results, and one of those results as gray text around the typed string. Pressing the tab key will complete the string to match the currently visible result, pressing the up and down keys will cycle through results. Clicking on the “k”, “c”, and “d” buttons, or on the right and left arrow keys, will cycle through the results for each of the three facets. If the current string appears anywhere on the expanded timelines, the respective elements will be highlighted. If there is an element that matches the search string but is not
representative enough to be displayed on the timeline, it will appear on top of the respective column. It will be significantly larger than the element before it, showing that the element does not follow the same size convention as the rest of the timeline. Also, whether or not there is an exact match in any time period, the respective time will be highlighted at the bottom of the timeline (Figure 3.7).

3.7 Filters

To give users control over how timelines are populated, beyond the selection of a single base element, we researched several ways of combining several facets in the same timeline. The final filtering mechanism in our solution results from a combination of two separate ideas.

Early sketches of the filtering mechanism contemplated only textual input (Figure 3.8). Users would input a word and later drag the icon of the chosen facet to a timeline to filter it. This form of interaction, with a input box appearing at the press of a character key, was later adopted for the text search mechanism we described above.

We were also interested in combining the elements from multiple timelines to obtain meaningful results. We pondered the possibility of linking two elements and displaying a visualization of their combined information, specifically through something we called “information fluxes” (figure 3.9).

For instance, by linking the ego to a contact we could choose to view the keywords or the documents exchanged, with the most important outgoing information being closer to the ego and the most important incoming information being closer to the contact. The issue with this idea is that such a connection
can not be coherently made between every combination of facets, contacts, and documents. As a case in point, connecting two keywords would not yield an intelligible connection of this sort. We pondered having different structures for different combinations of facets, but this went against our desire for a simple and consistent interface. Also, this sort of direct combination only allowed for a pairing of facets, and making an arbitrary number of connections would, naturally, be preferable. Finally, creating additional structures between elements in the visualization would add additional clutter, something we were concerned with while developing our solution, as we describe in detail later in this chapter.

Thus, rather than the requirement to type filters, and instead of a strict direct connection of visible elements, we arrived to a simpler and more consistent solution that is still powerful enough to contemplate arbitrary combinations of facets. Users can drag any keyword, contact, and file name, into any timeline, as many times and in any combination, to filter it. For example, clicking and dragging the mouse from a keyword in one timeline to the space occupied by a second timeline will add the keyword as a filter to it. Active filters appear to the left of the timeline and a simple click will remove it (Figure 3.10). This has been extended to the search string that appears on the top left corner of the screen, which can also be dragged into any timeline, making it extremely easy to filter a timeline according to any existing facet that users find through textual search.

3.8 Managing Clutter

There is a limit to how much information can be displayed in the same screen before it becomes too cluttered. This has led us to test and implement a number of coping measures. These measures work by separating overlapping timelines or by hiding unnecessary elements, either manually or automatically.

When the mouse is not above a timeline, only the most representative elements at the bottom and any elements at the origin of other timelines are shown, as the rest fades out. This happens gradually and smoothly, with a fast fade in when the mouse enters a timeline, and a slower fade out when it leaves, to let users momentarily view the contents of different timelines without having to constantly move the mouse over them. Moreover, timelines initially appear in a horizontal state, with the most representative elements from all time periods. Together with the previous measure, only relatively thin lines of important elements are seen most of the time.

We also let users hide or reposition existing timelines (and, even entire branches) they may consider less relevant by respectively clicking or dragging small circles located at the bottom left corner of each timeline. By moving the view and reorganizing information, the necessary relevant information can be observed without extraneous information.
Besides manually controlling the positioning of the timelines, we have implemented collision detection with gradual separation of overlapping timelines. Each timeline is encapsulated in a single oriented bounding box and, after creating or modifying a timeline, it is checked for collisions against every other timeline.

For each collision pair, collision detection complies to the following algorithm: the bounding boxes are projected onto the opposing axis; each projected box is checked for overlap against the opposing, non-projected bounding box; if, and only if, both pairs overlap, there is a collision (figure 3.11). Between each of the axis aligned boxes, overlap is determined by checking that the edges of a box are contained in the horizontal and vertical limits set by the edges of the other box.

To handle collisions, we modify the position of the conflicting timeline, that is, of the newly created or manipulated timeline. An alternative to this, and indeed the first option we considered, was to let users position the timelines were they desired and then adapt the remaining timelines to fit around that exact position. This posed three distinct problems: first, there is no guarantee that rearranging the timelines around a single timeline will result in a final resting state, as the reorganization could easily become circular; second, in a significantly dense structure, rearranging the timelines would lead to visual noise, as several timelines move about the canvas until they gradually reach a resting state; third after implementing the collision handling in this manner, we noticed that moving a timeline over a single timeline of similar size was somewhat confusing, as it appeared that the original timeline stayed in its place while the one under control moved away from it, when the opposite was true. The first problem could be minimized by imposing a threshold after which timelines would stop attempting to rearrange, but this would be an imperfect solution. The remaining are perceptual issues that would be even harder to eliminate. As such, we decided to simplify the problem by leaving the structure at rest and simply moving the conflicting timelines as they appeared.

Rather than simply moving the timelines to defined positions, we need to determine the required re-orientation and added length to the lines connecting the timelines to their parent timelines. This is done by gradually modifying a proxy bounding box to advance in both directions (increasing and decreasing rotation, and increasing the length of the connecting line in both, as decreasing it would lead to inadequate positions) until it finds a vacant spot, after which the minimal direction is determined and used to
gradually modify the timeline’s orientation and line length until it reaches a state without collisions. After stopping, the timeline is checked for collisions one more time, to make sure that further manipulations did not originate more overlap, after which all offspring are checked for collisions themselves, recursively going through the same process.

This repositioning is dynamic so users are free to continue their interaction with the application and the timelines are guaranteed to eventually reach a resting state with no overlap.

### 3.9 Help

At any time during the use of the application, users can press a button at the bottom left corner of the screen to activate the display of tooltips next to it. These tooltips show up when the mouse goes over any interactive element in the interface and contain concise instructions related to their use. They are aware of the context, so, for instance, the tooltip associated to a visible timeline’s control button will indicate that a click will hide it (figure 3.12), while the same button on a hidden timeline would indicate the opposite.

![Figure 3.12: VisMe, a tooltip explains the operation of a timeline control button as the mouse moves over it.](image)

### 3.10 Icons

Initial sketches of the VisMe interface included pictographic icons for buttons. Icons can, potentially, convey relatively complex ideas in a small space, but, unless we are dealing with images commonly linked to certain features (a floppy disk to save a document, a magnifying glass with a plus sign to zoom in, etc.) it is not certain that we can achieve completely unambiguous associations. In our case, we would require images for much less conventional needs, specifically to represent keywords, contacts, and documents. While contacts could easily be represented by a humanoid figure, and documents by a page with text on it (figure 3.13), representing keywords is not as straightforward. We experimented with a cartoon thought bubble and an actual key, but the meaning of the former is not very clear and the latter is often associated with security and encryption. At the same time, we began to appreciate the simple look of the functional prototype we were developing, in which these facets were represented by single letters. Since it was clear that both options would require further clarification, be it through the accompanying documentation or an internal help system, we opted for the simpler and more consistent representation through letters, aided by optional clarifying tooltips.
3.11 Use Case

We present a scenario which displays the main features of our solution in an attempt to retrieve a document.

Imagine that, during a time in which you were working on an important project, someone sent you a document that you remember as being of interest to you, but that you did not read. Now you want to retrieve it, but you don’t remember its name nor where you put it. You also don’t remember precisely when you worked on the project, besides the exact year, nor when the document was sent, but you do remember the topic of the project and the person who sent you the document. Using VisMe to search for this document, the following course of action could be taken.

First, you open a timeline with all the keywords in your personal information by dragging it from the initial "ego" element (figure 3.14). Then, you zoom the timeline into the year you remember working on the project and try to locate a keyword related to the project. However, being unable to see it immediately, you use the text search to find it, thus highlighting it on the timeline, indicating two specific months (figure 3.15).

You then open a new timeline with all the contacts and locate the person that you know sent you the document, again using the text search if it is not strong enough to be immediately visible. After doing that, you drag out a timeline with the documents related to that person and navigate to the particular months you identified earlier to find the document (figure 3.16). At this point, even if there are several documents related to that person in those months, the selection will be vastly reduced, making it easier to quickly investigate the different alternatives.
Figure 3.15: VisMe, use case. After searching for a keyword, both its instances and the months in which it appears are highlighted.

Figure 3.16: VisMe, use case. To the right are the results of the exploration: two documents fitting the known context.
Chapter 4

Implementation

As soon as we reached a general idea of the desired visualization and interaction techniques, we began implementing them in a prototype application which we used thereafter to experiment with and validate our solution. This application was programmed in Python using the Pyglet framework for access to OpenGL, window creation and handling, and text drawing.

4.1 Architecture

The program consists of two major components, sitting on top of a third-party indexing solution (figure 4.1): A data processing and access component, which directly interacts with the indexed information and pre-processes it for more efficient access, after which it provides specialized functions to retrieve the information we require for the visualization; and a graphical user interface component, which takes this pre-processed information and displays it according to the visualization and interaction techniques described in our solution.

4.2 Indexing

An automatic indexing application, Scribe, is used to gather and index personal information. It is capable of indexing and interconnecting emails and other document types, maintaining information about basic
file properties, content in the form of keywords, potential authors or associated email participants, among
other details. Scribe also provides the basic functions to access the resulting index.

4.3 Processing and Access

To make use of the index, we need to guarantee direct and efficient access to the information we need to
populate the timelines in our visualization. Essentially, considering the structure of our visualization and
the method of interaction, we need a list of elements, associated to and sorted by score, for each period
under examination and this list must be given according to another list of related keywords, contacts, and
document names. To do that, we developed a component that processes that information, generating
all the necessary intermediate information, and provides access to it.

4.3.1 Pre-Processing

Using the functions provided by Scribe, programmers can query the data to obtain, for instance, all the
documents with a given author. However, this sort of query is not used in our implementation, as it is
too computationally expensive. Instead, we pre-process the available information, by running through
all available documents and creating a list of all documents using our own specialized structure (id,
name/title, keywords, contacts, year, month, day). Additionally, we also examine all available events
(linking emails to documents and people) for contacts. Contacts are identified by a series of aliases or
emails. If no name exists, we use the email. In the case of documents with no author or linked to no
contact through emails, its author is simply set to "unknown" and is not considered for contact weight
purposes. The keywords extracted from each document can be configured, in Scribe, to be selected
by number of occurrences or, as we preferred, by tf-idf (term frequency-inverse document frequency)
weight, a calculation of the relative importance of a keyword to a document in a collection that attributes
less relevance to overly common words thus choosing potentially more interesting keywords.

4.3.2 Relations and Score

Documents are considered related to a keyword if they contain it; to a contact if they were authored,
sent, or received by it; and to another document if that document contains any of the same keywords
or contacts. Keywords and contacts are considered related to other keywords or contacts if they are
related to the same documents.

The relevance, or score, of individual facets determines their strength (size and position) in the visu-
alization. Keyword and contact scores depend on their frequency in the documents, while a document's
score is determined by the scores of its keywords and contacts.

These scores are first gathered for each element within individual documents, but document scores
cannot be determined individually, and keyword and contact scores must later be adapted dynamically
to the group of elements retrieved to be visualized. The final score of a keyword or of a contact within a
group is then the sum of all the individual scores of its occurrences. Regarding document scores, these
are determined in three steps. First, a list of keywords and contacts related to all the documents in a
group is gathered, adding the scores for each individual occurrence; then, for each document, its key-
word and contact scores are set to be the product between their original values and the summed scores
of its occurrences in the group; afterwards, since keywords and contacts present values of different mag-
nitudes, these products are divided by the highest keyword or the highest contact; finally, the score of
the document is set to the sum of the average of the keyword and of the contact processed values (there
are many more keywords than contacts, so even if there are no matching keywords, this will ensure that
a matching contact will result in a reasonably high relevance). This will give us an approximated notion of the relevance of the document within the collection, and also assures that the associations between documents contained in the same email are represented in the visualization by the fact that all of their contacts are the same, giving these documents higher scores and placing them in the top positions of relevance within the resulting timelines. There is no separation between contacts and keywords in this score, so another document, unrelated by any contact, can be given the same level of prominence if it contains many of the same most relevant keywords.

We are not directly concerned with the absolute values calculated for these scores, but rather with their relative value within a group. To be visualized, the values in a group of elements (retrieved with two of the functions mentioned previously) must be changed to fit in a range from zero to one, which is done by dividing all scores by the value of the highest score. This quotient is then used as a size modifier, with the highest rated element being assigned the highest possible size in the timeline while the other elements are sized down accordingly, until a minimum limit is hit. The relative order of these scores is also used to place the elements in the columns corresponding to their temporal presence, with the highest value below and the lowest values progressively further up.

The algorithms and heuristics used to gather and interconnect information, as well as rate its importance in relation to the collection of documents, are not meant to be optimal, as they are not the focus of this dissertation. Rather, they are straightforward calculations meant to offer a reasonable indication of the contents and the contacts associated to a person’s collection of digital documents which allow us to implement a testable application of our visualization and interaction techniques to validate our solution. Further improvements in this realm can have an impact on the adequacy of the results and are thus open for future research.

4.3.3 Retrieval

To retrieve documents, we use a function that takes as parameters a date (year, month, and day; these can be selectively provided or omitted to give all documents of all time, of an entire year, of a month in a year, or of a specific day in a month in a year) and an arbitrary number of document names, keywords, and contacts and returns a list of documents related to the given facets. In the particular case of the given document names, this function will generally only return the documents with the given names, as for purposes of analyzing the associated keywords and contacts we are interested in a more exact relation. To observe a timeline of documents, the function is used with a special parameter to return the documents indeed related to the given names.

Finally, three auxiliary functions take the list of documents gathered with the previous function and return a list of the contained keywords, contacts, or document names to populate the timelines seen in the graphical interface, each with a score that represents their relative importance within the given set of documents.

4.4 Interface and Visualization

In the description of our solution, we identified a restricted and well defined number of elements required for both the user interface and the visualization itself. Thus, implementing VisMe, beyond the underlying information processing, was a matter of creating these elements and the interactions between themselves and the user by developing the appropriate structures, graphical representations, and manipulations.
4.4.1 Interface and Visualization Elements

All the elements of the interface, including buttons and visualization elements (keywords, contacts, and documents), consist of a bounding box (figure 4.2) that stores their size and position and is used for mouse interaction and, in the case of timelines, for collision detection; and a graphical representation, typically a label. This is not an absolute requirement, the timeline control buttons, for instance, are graphically represented by a circle and their bounding area is a corresponding circle used only to detect when the mouse is above using the distance to the center and the radius. Additionally, individual elements also contain other specific properties, including links to other elements.

![Figure 4.2: VisMe, several elements with their bounding boxes and graphical representations.](image)

The position of these interface elements is either set to the screen, in the case of the text search, the help button and tooltips, the automatic view control button, and also the line that indicates the final position and orientation of timelines as they are extracted; or to an infinite canvas, a portion of which is visualized at any time by a camera controlled by way of translation, rotation, and zoom. There is no concrete canvas structure, the camera itself contains values for position, rotation, and scale, while the elements not in screen space define absolute positions independently. The exception to this is when their actual position must be centered on the camera to perform correct camera rotations, at which moment their are iteratively translated and the camera matches the inverse translation.

4.4.2 Visualization Structure

The visualization, consisting of interconnected timelines, is internally structured as a graph in which the visualization elements populating each timeline can give rise to up to three timelines containing other visualization elements. A visualization element has three child buttons (one for each facet) and a parent timeline (in the case of the initial "ego" element, this is null); a button has a parent visualization element and a child timeline; and a timeline has a parent button, a child control button, and several child visualization elements, filters (the interface representation, not the internal information about the filters which is kept in a different structure), and periods (the interface elements used to visually indicate time and also to control the granularity of the timeline). The timeline control button, filters, and periods do not have child elements of their own.
Chapter 5

Evaluation

VisMe was subjected to an heuristic evaluation at an early stage of its development, shortly followed by a first round of usability tests. These evaluations served not only to validate it, but also to identify problems that should be corrected. Once we completed the final prototype, we conducted final usability tests to fully validate our solution.

The reason for two separate user evaluations is twofold. First, they were done at distinct stages of the development of the our solution (one after the completion of the first functional prototype, and the second after the conclusion of the final prototype) and thus the first of these allowed us not only to validate our solution, but also to guide its development throughout the remainder of our work. At this point we were mostly interested in verifying that our design choices made sense and were understood by users.

Second, while ideally we would prefer to test our solution with real user data, that is not always feasible. It would be impossible to ask users to perform very specific tasks, such as finding a particular document from its title, since we cannot predict what information exists in the indexed data of each user. On the other hand, if we asked very generic questions, like remembering the title of a document to retrieve it, we would be left with potentially incomparable results, as users would be working with very different data sets.

However, not testing with real data makes it impossible to determine VisMe's viability in terms of discovering interesting trends and patterns in one's information, since users would have no personal connection to the information.

As such, we devised a first set of tasks that targets specific and directly comparable document retrieval and pattern detection operations, making sure the data set allows for their execution (which does not mean that the results are offered to the users with little effort) and produces comparable results, and a second set that uses more generic tasks (but still specific enough to be comparable) to test the fundamental operations in a manner that we can be sure to be doable under all circumstances and, more importantly, also examines the possibility of using VisMe to find personally relevant patterns in the visualized information.

5.1 Heuristic Evaluation

An early prototype of our solution was submitted to an heuristic evaluation with the purpose of identifying any usability problems with the interface.
5.1.1 Method

Four usability experts were asked to perform three tasks, meant to exercise all the functionalities provided by the prototype, and to evaluate the interface regarding its compliance with Nielsen's heuristics [17].

The tasks were performed over a collection of 106 text documents authored by 13 individuals over a span of 3 years.

**Task 1** Find a document based on the given knowledge of its author, main keyword, and year.

**Task 2** Find a document based on the context (the period of time in which an unrelated given keyword was the most relevant).

**Task 3** Describe the evolution of a given keyword over a given year (in which month it appeared, in what month it disappears, if it evolved to become more or less important, or stayed the same).

These tasks follow a progression that covers the possible strengths of the VisMe system: locating a document from the facets directly connected to it (time, content, and contact); finding a document based on the context surrounding it (i.e. the document existed at a time in which a certain keyword was known to be very relevant to the user's personal life); and understanding the evolution of an element in the provided timelines (a basic operation for the successful detection of more complex trends and patterns).

The experts had one hour to work on these tasks and take note of all perceived problems with the interface.

5.1.2 Results

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visibility of system status</td>
<td>3</td>
</tr>
<tr>
<td>Match between system and the real world</td>
<td>2</td>
</tr>
<tr>
<td>User control and freedom</td>
<td>4</td>
</tr>
<tr>
<td>Consistency and standards</td>
<td>3</td>
</tr>
<tr>
<td>Error prevention</td>
<td>3</td>
</tr>
<tr>
<td>Recognition rather than recall</td>
<td>3</td>
</tr>
<tr>
<td>Flexibility and efficiency of use</td>
<td>3</td>
</tr>
<tr>
<td>Aesthetic and minimalist design</td>
<td>2</td>
</tr>
<tr>
<td>Help users recognize, diagnose, and recover from errors</td>
<td>3</td>
</tr>
<tr>
<td>Help and documentation</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 5.1: Number of problems detected by the four experts for each of Nielsen's heuristics.

Besides identifying where the interface failed to meet the heuristics under consideration, the experts detected several problems with the interface and suggested possible solutions.

The lack of status visibility was mainly associated to navigation, specifically the tendency of some experts to lose themselves after expanding and manipulating several timelines over a long period of time. One evaluator suggested the implementation of a compass, pointing to the center of the structure or to the initial "ego" element. A more direct approach, smoothly and gradually moving and zooming the view to encompass the entire visible structure automatically or with a double click was eventually implemented.

The experts generally considered that there was a match between the system and the real world, deeming it simple and easily understood, although two experts expressed their concern over the difficulty in understanding the meaning of the different icons ("k", "c", "d") without a previous explanation,
something we later attempt to solve with the addition of descriptive tooltips when the mouse moves over those icons; and another evaluator posed the possibility of the control method not being suitable for people without ample experience with computers.

All but one expert considered that the system allowed for user control and freedom. That particular expert’s consideration was later found to be linked to a bug which destroyed a timeline and made it impossible to manipulate it in any way without dragging a new one out of the same original element. Other concerns were the fact that the system only worked with mouse input and the impossibility of directly returning to the “ego” element. This last problem is linked to the problem of orientation described earlier, and can similarly be eased by the ability to automatically encompass the entire timeline tree, which that can then reveal the position of this initial element.

All evaluators acknowledged the quality of the system in terms of consistency and standards. although there were some remarks about the possibility of exploring different uses of color and allowing users to link different elements rather than exclusively extracting timelines out of single elements. That sort of facet combination was later implemented in the form of a filtering mechanism.

The absence of error prevention observed by two of the experts was due to both the possibility of producing overlapping timelines, causing buttons and other elements to block those bellow them, as well as the lack of a clear distinction between clickable and unclickable elements, such as the days under the timelines (it was impossible to go into a finer granularity). The second problem (absent for keywords, contacts, and document names, as they were highlighted on mouse over) was solved by changing the text font to bold on mouse over on buttons and times (and later by making the timeline drag/hide circles bigger on mouse over). The second problem is closely related to the issue of clutter that we identified from the beginning of the project. This particular situation was solved by adding collision detection, followed by smooth and gradual separation of timelines.

All experts spotted shortcomings in providing functionalities to facilitate recognition rather than recall. The meaning of the buttons (“k”, “c”, “d”) or of the different sizes of the elements and of their sequential distribution through the timelines, although explained in the initial demonstration, was observed to not be unequivocally recognizable and, as suggested by on evaluator and later implemented, was later handled by providing tooltips on mouse over.

Flexibility and efficiency of use were understood to be met by all experts. Still, one evaluator mentioned the lack of keyboard shortcuts and alternatives to mouse control, specifically for zooming in and zooming out, to which we responded with the use of Page Up and Page Down keys for the effect, and the absence of a direct indication of the center or origin of the timeline tree without the need to zoom out and pan out, which prompted the already mentioned automatic view control. A second expert noted the lack of a button to return to the highest level of granularity in the timelines, which we promptly corrected.

The interface design was accepted to be aesthetically and minimalist by all but one expert who pointed out the overlap of timelines as being contrary to these principles.

Recognizing, diagnosing, and recovering from errors was not facilitated in any way by the interface. The situations in the prototype that would have benefited from better handling led only to empty or disappearing timelines. A simple text box with a description of the error and the possibility to recover the lost state was suggested by one of the experts. Another evaluator mentioned the lack of undo and redo.

Help and documentation were completely absent beyond the initial demonstration of the prototype. Tooltips, already suggested and implemented to improve recognition, also contribute to fill this void. One expert also suggested providing a help screen to be displayed at the start of the application and on demand with an explanation of all the elements of the visualization and the available features. We believe the tooltips serve this purpose well enough and have not implemented this particular solution.
5.1.3 Discussion

The evaluators detected a series of important flaws in the interface. This contributed to the improvement of the solution, sometimes by more or less directly following the suggestions of the experts, but often with different approaches more befitting of our simple and minimalist design goals.

As important as it was to detect problems at an early stage of development, it was also significant to observe the very positive reactions towards the prototype. The experts displayed great enthusiasm while using the tool, generally considering it aesthetically pleasing and simple to use, some of the properties we strived for when designing this solution.

5.2 First Usability Tests

Between the heuristic evaluation and the first user tests the only addition to the prototype was the text search mechanism explained earlier. The need to use a realistically sized data set meant that locating specific keywords, contacts, or document names could become nearly impossible without a direct search mechanism.

5.2.1 Method

To validate our solution we asked 20 volunteers, aged 17 to 29 ($\bar{x} = 23.7$, $\bar{\tilde{x}} = 23.5$, $\sigma = 2.7$, mode = 24) and with self reported high level of experience with computers ($\bar{x} = 3.7$, $\bar{\tilde{x}} = 4$, $\sigma = 0.5$, mode = 4, four point scale), to perform a series of tasks using the prototype over a set of 1004 text documents authored by 102 people. This data set was crafted to be representative of a real, albeit small, collection of documents, with realistic trends and patterns for each tested combination of facets and enough documents and authors to make it hard to just stumble upon them without significant help from the interface. Having full control of the data set allowed us to craft more complete and coherent set of tasks, to validate our interface design decisions. A second evaluation, described later, was performed to validate the solution using actual personal information from the users.

Eight of these tasks consisted of finding a document based on the knowledge of one or of a combination of its facets (time, most representative keyword, and author, as well as a single task in which the actual file name was given). For instance, "Find a document written by Bob about the Internet that you read in May of 2009" or just "Find a document about guitars".

**Task 1.1** Finding a document from a given date (month and year).

**Task 1.2** Finding a document containing a given keyword.

**Task 1.3** Finding a document related to a given contact.

**Task 1.4** Finding a document with a given name.

**Task 1.5** Finding a document from a given date and containing a given keyword

**Task 1.6** Finding a document from a given date and related to a given contact.

**Task 1.7** Finding a document containing a given keyword and related to a given contact.

**Task 1.8** Finding a document from a given date, containing a given keyword, and related to a given contact.
These tasks were meant to evaluate the basic document retrieval functionalities of the proposed solution, specifically the ability to obtain a document timeline related to any facet and, more generally, the interaction method and visualization technique.

The remaining nine tasks required users to determine the most significant keyword, contact, and a combination of the two in a time period (and the other way around) as well as its evolution throughout that period (when it started, when it ended, and whether it went up, down, or stayed the same). For example, “Who did you contact the most in January 2010” and “How did that contact evolve during that year”.

**Task 2.1** Find the year in which a given contact was most relevant and describe how that contact evolved over that year (in which month it appeared, in what month it disappears, if it evolved to become more or less important, or stayed the same).

**Task 2.2** Find the year in which a given keyword was most relevant and describe how that keyword evolved over that year (in which month it appeared, in what month it disappears, if it evolved to become more or less important, or stayed the same).

**Task 2.3** Find the most relevant contact in a given year and describe how that contact evolved over that year (in which month it appeared, in what month it disappears, if it evolved to become more or less important, or stayed the same).

**Task 2.4** Find the most relevant keyword in a given year and describe how that keyword evolved over that year (in which month it appeared, in what month it disappears, if it evolved to become more or less important, or stayed the same).

**Task 2.5** Find the contact more strongly connected to a given keyword.

**Task 2.6** Find the keyword more strongly connected to a given contact.

**Task 2.7** Find the most relevant contact connected to a given keyword in a given year.

**Task 2.8** Find the most relevant keyword connected to a given contact in a given year.

**Task 2.9** Find the year in which a given contact was more strongly connected to a given keyword.

The purpose of these tasks was to determine if users were capable of discerning different levels of importance of keywords and contacts as they were displayed in the timelines and recognizing the evolution of such elements, the basic requirements for the detection of trends and patterns.

Both sets of tasks also provided some insight into the general quality of the interface in terms of interaction (expanding timelines, controlling the view, etc.) and organization of information using timelines. Additionally, these tasks are not merely abstract exercises specifically designed to validate the available features, they are meant to be representative of actual uses of the application.

These tasks were timed and recorded for later analysis. Users were also asked to grade the difficulty of each task with a four point scale upon completion. The time limit for each task was 150 seconds, after which users would be told to move on to the next task. The order in which tasks were performed was random, in order to prevent result biases.

Users were first given a 5 minute demonstration of the prototype and all its features followed by 5 minutes of free experimentation with the interface. They also answered a questionnaire at the end of the session to determine their satisfaction with the interface using a four point scale (for example, “This example was: confusing 1 2 3 4 clarifying”, with 1 being the most confusing, 4 the most clarifying, and the rest of the values standing for something in between).
5.2.2 Results

We recorded success, completion time, and difficulty for each task (as rated by the users), as well as the answers to the questionnaire, and later performed a statistical analysis of that data. Below, we present the major findings of that analysis relative to three main groups: document retrieval, pattern recognition, and satisfaction. These results are shown through averages ($\bar{x}$), medians ($\tilde{x}$), standard deviations ($\sigma$), confidence intervals (CI, 95%), and, in the difficulty measurements and in the questionnaires, modes. We also use Student's t-tests, with a confidence of 95%, to determine the statistical significance of the differences between the recorded results of distinct tasks.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{Figure_5.1.png}
\caption{First user evaluation, average task completion times in seconds (95% confidence interval).}
\end{figure}

Document Retrieval

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|}
\hline
Task & Average & Median & Standard Deviation & Confidence Interval (95\%) \\
\hline
1.1 & 47.6 & 39.0 & 19.4 & 12.0 \\
1.2 & 36.2 & 35.5 & 14.6 & 9.0 \\
1.3 & 34.6 & 34.5 & 12.8 & 8.0 \\
1.4 & 38.6 & 32.5 & 20.3 & 12.6 \\
1.5 & 48.4 & 44.5 & 18.6 & 11.5 \\
1.6 & 58.6 & 51.0 & 19.0 & 11.8 \\
1.7 & 84.5 & 77.0 & 27.5 & 17.0 \\
1.8 & 88.7 & 84.0 & 27.6 & 17.1 \\
\hline
\end{tabular}
\caption{First user evaluation, document retrieval task times in seconds.}
\end{table}

Finding documents based on a single facet (tasks 1.1 to 1.4) or on a combination of one facet with time (tasks 1.5 and 1.6) were generally performed successfully (2 failures in 20 sections for task 1.5, finding documents based on a keyword and a date; no failures for the remaining tasks) in under a minute, and they were considered very easy.
There is a small but statistically significant difference between finding a document based on time alone and based on a single keyword (t-test value = 0.04) or contact (t-test value = 0.02), while the difference is not as significant between finding a document based on time and the document title (t-test value = 0.16) or based on a single facet other than time (t-test value = 0.72, t-test value = 0.46, and t-test value = 0.66). To retrieve a document based on time, users had to manipulate the granularity of the timelines until they reached the desired date (a specific month in a year). This process is different from a text search for a keyword, contact, or document name, which immediately reveals the desired element in the respective timeline, and is, perhaps, slightly faster, although users could alternatively browse the timelines to look for the elements. Still, retrieving documents based on a single facet other than time is done with uniform ease, which can be attributed to the uniform handling of these three facets by the interface.

Two tasks (tasks 1.7 and 1.8), in which the users had to find a document based on both a keyword and an author (and, in one, also time) as opposed to doing the same based only on a keyword or author (with and without time), clearly stand out by having the greatest number of failures (6 and 5 in 20 sessions, respectively), the highest average completion times ($\bar{\tau} = 84.5$, $\bar{\tilde{\tau}} = 77$, $\sigma = 27.5$; and $\bar{\tau} = 88.7$, $\bar{\tilde{\tau}} = 84$, $\sigma = 27.6$) and being considered the most difficult tasks ($\bar{\tau} = 2.6$, $\bar{\tilde{\tau}} = 3$, $\sigma = 1.4$, mode = 3; and $\bar{\tau} = 2.6$, $\bar{\tilde{\tau}} = 2.5$, $\sigma = 0.9$, mode = 2). To complete these two tasks, users had to either expand documents
from two facets and cross check the results, or expand documents from one facet and expand the other facet out of each document one by one. Although that is not overly complicated, especially given the fact that the number of documents that matched one of the facets (or one of the facets plus time) ranged from only two to a dozen, it was not straightforward for many users. This strengthened the need for a more direct method of combining facets, which we attempted to provide with the addition of the filtering mechanism described earlier.

Pattern Recognition

<table>
<thead>
<tr>
<th>Task</th>
<th>Average</th>
<th>Median</th>
<th>Standard Deviation</th>
<th>Confidence Interval (95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>36.9</td>
<td>30.0</td>
<td>20.6</td>
<td>12.8</td>
</tr>
<tr>
<td>2.2</td>
<td>44.2</td>
<td>40.0</td>
<td>21.2</td>
<td>13.2</td>
</tr>
<tr>
<td>2.3</td>
<td>48.2</td>
<td>47.0</td>
<td>18.3</td>
<td>11.3</td>
</tr>
<tr>
<td>2.4</td>
<td>46.8</td>
<td>43.5</td>
<td>19.8</td>
<td>12.3</td>
</tr>
<tr>
<td>2.5</td>
<td>28.1</td>
<td>25.0</td>
<td>19.0</td>
<td>11.8</td>
</tr>
<tr>
<td>2.6</td>
<td>18.3</td>
<td>17.5</td>
<td>7.9</td>
<td>4.9</td>
</tr>
<tr>
<td>2.7</td>
<td>31.5</td>
<td>32.0</td>
<td>10.8</td>
<td>6.7</td>
</tr>
<tr>
<td>2.8</td>
<td>30.8</td>
<td>28.0</td>
<td>13.0</td>
<td>8.0</td>
</tr>
<tr>
<td>2.9</td>
<td>35.6</td>
<td>28.0</td>
<td>17.7</td>
<td>11.0</td>
</tr>
</tbody>
</table>

Table 5.4: First user evaluation, pattern recognition task times in seconds.

Trend and pattern detection tasks were generally completed successfully (4 out of 9 tasks with 1 failure in 20 sessions each) in under a minute (tasks 2.1 to 2.9). They were also considered very easy.

<table>
<thead>
<tr>
<th>Task</th>
<th>Average</th>
<th>Median</th>
<th>Standard Deviation</th>
<th>Confidence Interval (95%)</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>3.8</td>
<td>4.0</td>
<td>0.5</td>
<td>0.2</td>
<td>4</td>
</tr>
<tr>
<td>2.2</td>
<td>3.9</td>
<td>4.0</td>
<td>0.5</td>
<td>0.2</td>
<td>4</td>
</tr>
<tr>
<td>2.3</td>
<td>3.6</td>
<td>4.0</td>
<td>0.6</td>
<td>0.3</td>
<td>4</td>
</tr>
<tr>
<td>2.4</td>
<td>3.6</td>
<td>4.0</td>
<td>0.7</td>
<td>0.3</td>
<td>4</td>
</tr>
<tr>
<td>2.5</td>
<td>3.7</td>
<td>4.0</td>
<td>0.6</td>
<td>0.3</td>
<td>4</td>
</tr>
<tr>
<td>2.6</td>
<td>4.0</td>
<td>4.0</td>
<td>0.0</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>2.7</td>
<td>4.0</td>
<td>4.0</td>
<td>0.2</td>
<td>0.1</td>
<td>4</td>
</tr>
<tr>
<td>2.8</td>
<td>3.9</td>
<td>4.0</td>
<td>0.3</td>
<td>0.1</td>
<td>4</td>
</tr>
<tr>
<td>2.9</td>
<td>3.7</td>
<td>4.0</td>
<td>0.7</td>
<td>0.3</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 5.5: First user evaluation, pattern recognition task difficulty, from hardest (1) to easiest (4).

Tasks 2.1 to 2.4, besides asking users to find the most relevant keyword or contact in a year, or the year in which a keyword or contact were the most relevant, demanded a description of the evolution of that element over the respective year, while the remaining tasks asked only for the identification of the most relevant facets in different combinations without the need to describe their evolution. These four tasks present similar times and difficulties, evidencing no statistically significant difference between them. Among the other subgroups of directly comparable tasks (2.5 and 2.6; 2.7 and 2.8) there is also no statistically relevant difference in times and difficulties. This gives us an indication that the unified representation of different facets in timelines allows users to identify the most significant elements in a time period and their evolution over time with uniform ease.

Satisfaction

Note These values are presented in a four point scale, from 1 to 4, with higher values being associated to greater degrees of agreement.
**Question 1** General satisfaction: \( \bar{x} = 3.4, \tilde{x} = 3, \sigma = 0.5, \text{mode} = 3. 

Users were generally satisfied with the system.

**Question 2** Ease of use: \( \bar{x} = 3.2, \tilde{x} = 3, \sigma = 0.7, \text{mode} = 3. 

For the most part, the users did not find the system to be difficult to use, which fits with the measured ability to perform the tasks they were given. This is an indication that we succeeded in creating a simple interface that directly aids in the execution of the tasks we intended to facilitate.

**Question 3** Ease of first use: \( \bar{x} = 3.2, \tilde{x} = 3, \sigma = 0.8, \text{mode} = 4. 

Initiation in the system was not generally found to be difficult, which may, again, result from the intended simplicity of the interface.

**Question 4** Ease of learning: \( \bar{x} = 2.8, \tilde{x} = 3, \sigma = 0.9, \text{mode} 3. 

A few users found VisMe somewhat difficult to learn. The initial unfamiliarity of the interface may contribute to this, as it was their first experience with an unconventional interface. Also, although users were instructed on how to use all available functionalities, there was no clear indication or memory aid in the interface itself, since tooltips were only introduced in a later version of the prototype.

**Question 5** Ease of remembering commands: \( \bar{x} = 3.1, \tilde{x} = 3, \sigma = 0.6, \text{mode} = 3. 

Users did not consider it difficult to remember commands. With VisMe’s interface, this is facilitated by the minimalistic interface with few, or possibly none, complex commands.

**Question 6** Adequacy of the available functionalities: \( \bar{x} = 2.9, \tilde{x} = 3, \sigma = 0.6, \text{mode} = 3. 

Some users did not find the available functionalities adequate. This may stem from their frustration when performing a few of the more complicated tasks, such as tasks 1.7 and 1.8, which an additional facet combination functionality could have facilitated considerably.

**Question 7** Flexibility of the system: \( \bar{x} = 3.3, \tilde{x} = 3, \sigma = 0.8, \text{mode} = 3. 

Users were generally satisfied with the system’s flexibility. It was our intention to provide an interface that is minimal and simple while still flexible enough to handle a relatively wide range of tasks, from document search and retrieval, to the identification of trends and patterns, and although a quick judgment from the users does not fully validate this flexibility, we also have measured results that confirm the ability of the interface to facilitate the aforementioned tasks.

**Question 8** Predictability of the result of an operation: \( \bar{x} = 2.9, \tilde{x} = 3, \sigma = 0.6, \text{mode} = 3. 

Operations were not thought to be unpredictable or very predictable, although some users leaned to the former. Their issue with predictability may stem from not knowing what information will be presented with every extraction and manipulation of timelines, or it can result from not using the system for a long enough period of time to become completely familiar with the outcome of every functionality. During the execution of the tasks, we did not observe continuous surprise with the consequences of different actions within the application, but we did notice ongoing experimentation as users tried different things to solve the problems at hand.

**Question 9** Clarity of the feedback after an operation: \( \bar{x} = 3.3, \tilde{x} = 3, \sigma = 0.6, \text{mode} = 3. 

Users were generally satisfied with the clarity of the feedback provided after each operation. Although there are no obvious indicators of the completion of an action, such as messages or visual signs, all interactions have an effect on the visualization and users did not report complications in this area.
Question 10 Adequacy of the number of steps required to complete a task: $\bar{x} = 3.5$, $\tilde{x} = 3$, $\sigma = 0.5$, mode = 3.
Most users did not view the number of steps necessary to complete a task as excessive.

Question 11 Logic in a sequence of steps to complete a task: $\bar{x} = 3.6$, $\tilde{x} = 4$, $\sigma = 0.6$, mode = 4.
The required sequence of steps to complete a task was also deemed logical.

Question 12 Ease of view control: $\bar{x} = 3.1$, $\tilde{x} = 3$, $\sigma = 0.8$, mode = 3.
Manipulating the view by way of translating, rotating, and zooming in and out was considered easy, although not very easy. We did observe some apparent awkwardness in some users when they controlled the view, but that may just have been due to a lack of proper time to adjust to the method of interaction.

Question 13 Ease of extracting timelines: $\bar{x} = 3.2$, $\tilde{x} = 3$, $\sigma = 0.5$, mode = 3.
Users did not find the timeline creation process to be difficult. The value could be higher, since the operation itself is very simple, but it is possible that the need to adjust the view and sometimes the timelines themselves after an extraction associates some difficulty to this method of exploring information.

Question 14 Adequacy of the mouse as a control method: $\bar{x} = 3.7$, $\tilde{x} = 4$, $\sigma = 0.5$, mode = 4.
The mouse was considered a viable control method by most users, although no alternative method was presented to them for consideration.

Question 15 Ease of automatically centering the view on a timeline: $\bar{x} = 3.3$, $\tilde{x} = 3.5$, $\sigma = 0.8$, mode = 4.
For most users, the operation of automatically centering the view on a timeline by double clicking was found to be easy, even if, according to our observations, users preferred to focus on controlling the view manually.

Question 16 Clarity of the meaning of the facet buttons ("k", "c", and "d"): $\bar{x} = 2.9$, $\tilde{x} = 3$, $\sigma = 0.6$, mode = 3.
On the surface, the "k", "c", and "d" buttons’ meaning is not clear. After being explained what they mean and to what facets they are related, users reported no great difficulty in recalling that information. While users generally considered the meaning clear, if not very clear, there were some who had reservations. The addition of tooltips may alleviate the need for external training in this and in other similar cases.

Question 17 Desirability of pictures instead of letters for the facet buttons: $\bar{x} = 2.8$, $\tilde{x} = 3$, $\sigma = 0.8$, mode = 3.
As seen above, most users considered the buttons clear. Still, most users were also slightly receptive to a substitution, although not overly so. However, possibilities for such icons were not shown nor did the users suggest examples. As explained before, we eventually moved away from this possibility due to our desire to maintain a level of aesthetic cohesion and simplicity which we believe is more directly accomplished by limiting the interface to textual imagery. We attempted to correct the problem of conveying the exact meaning of the buttons through simple and tooltips and unobtrusive tooltips. Icons, while potentially better suited to conveying meaning in a small space, cannot be assured to be completely unambiguous, so their use would not dispense the need for clarification.

Question 18 Legibility of the text in the timelines: $\bar{x} = 3.3$, $\tilde{x} = 3$, $\sigma = 0.6$, mode = 3.
The text as it was displayed in the timelines was considered legible. Some users may have taken
issue with the text in zoomed out views or in overlapping timelines, thus why it was not considered completely legible.

**Question 19** Clarity of the general text search results (top left corner): $\bar{x} = 3.5$, $\tilde{x} = 3.5$, $\sigma = 0.6$, mode = 4.

The way the general text search results were presented to the users was considered almost completely clear by the majority. Because the presentation follows common text search result presentations, understanding them posed little difficulty.

**Question 20** Clarity of the localized text search results (marked in the timelines): $\bar{x} = 3.6$, $\tilde{x} = 4$, $\sigma = 0.5$, mode = 4.

The search results marked on the timelines by highlighting elements and periods of time with possible results was mostly considered clear.

**Question 21** Ease of selecting different search results: $\bar{x} = 3.3$, $\tilde{x} = 4$, $\sigma = 0.9$, mode = 4.

To scroll through the visible search results, users had to make use of the up and down keys. These actions were considered very easy by users.

**Question 22** Adequacy of the use of color: $\bar{x} = 3.4$, $\tilde{x} = 3$, $\sigma = 0.7$, mode = 3.

The use of color was considered adequate, although not completely adequate by all users, who did not care to elaborate on their judgments in this regard.

**Question 23** Ease of selecting highlight colors: $\bar{x} = 3.7$, $\tilde{x} = 4$, $\sigma = 0.5$, mode = 4.

Selecting colors by moving the mouse over an element was generally considered very easy. However, it should be noted that manipulating the color of the highlights was not a requirement for the fulfillment of any of the tested tasks, being an optional aid for following the evolution of an element throughout the timeline. We did not observe extensive use of this feature by any of the users, so their judgment may not fully contemplate the quality of this method of interaction in more complex situations, such as attempting to color several groups of elements with different colors. It is, however, an indication that color selection in this manner poses no immediate difficulty to the average user.

### 5.2.3 Discussion

The first usability tests proved that, using the VisMe prototype, users were capable of identifying and following the relative importance of keywords and contacts over time. They were also capable searching for documents with the initial knowledge of different facets. While these results do not fully validate our solution, they show that interface is suited for the performance of the essential pattern detection and document retrieval operations we aimed to facilitate. Also, understanding what worked and what did not work guided the development of the solution and its prototype from this point onwards. That was the case with the addition of a filtering mechanism capable of facilitating the combination of multiple facets in the same timeline, something noticeably lacking from the prototype during these tests, as it was highlighted by the results.

### 5.3 Final Usability Tests

Following the conclusion of the final VisMe prototype, we conducted a second set of usability tests. Having validated the main ideas behind the solution using an artificial collection of documents, this final evaluation was expected to not only test the modifications to the interface but also to more fully validate
the entire solution by testing it over real user data. This can determine if the same interaction and visualization techniques that were shown to work with a controlled data set, are also effective with the added size and indexing noise of real user data. It was also crucial to evaluate the applicability of these techniques when it comes to discovering personally interesting patterns in the visualized information.

5.3.1 Method

We asked 10 volunteers, aged 17 to 53 (\(\bar{x} = 25.6, \tilde{x} = 24, \sigma = 10.2, \text{mode} = 24\)) and with a high level of self evaluated experience with computers (\(\bar{x} = 4.3, \tilde{x} = 4.5, \sigma = 0.8, \text{mode} = 5\), five point scale), to index their email accounts and their document folders prior to executing a series of tasks designed to test the essential features of the VisMe prototype and also to validate its ability to cope with actual personal information and provide personally relevant results.

The tasks can be split into two main groups: one with three subgroups of simple pattern detection and document retrieval tasks; and the other with a single large task of free exploration. They were necessarily more generic than those of the previous study as it was impossible to know beforehand what would be found in the users’ own information.

Two pattern detection and document retrieval tasks required users to alternately identify the most important keyword and contact in a given year and how it evolved over the months, as well as to retrieve the documents related to that keyword and the ones related to that contact.

Task 1.1 Find the most relevant keyword in a given year.

Task 1.2 Describe how that keyword evolved over that year (in which month it appeared, in what month it disappears, if it evolved to become more or less important, or stayed the same).

Task 1.3 Obtain the documents related to that keyword.

Task 2.1 Find the most relevant contact in a given year.

Task 2.2 Describe how that contact evolved over that year (in which month it appeared, in what month it disappears, if it evolved to become more or less important, or stayed the same).

Task 2.3 Obtain the documents related to that contact.

These tasks were meant to verify that the essential pattern recognition and document retrieval capabilities still functioned correctly when applied to real user data.

The third task asked users to find the most important keyword related to the contact found in one of the previous tasks before requesting users to find a document related to both those facets.

Task 3.1 Obtain the documents related to both the contact found in a previous task and the most important keyword connected to that contact (this keyword was to be identified before starting the actual task since both a contact and a known related keyword were required for its execution).

This task was used to test the filtering mechanism added since the previous evaluation.

Each individual task had a time limit of 150 seconds, after which users classified its difficulty using a five point Likert scale.

The exploration task, with a 15 minute time limit, required users to explore their personal information freely in search of interesting trends and patterns. As they did this, they were instructed to vocally communicate their thought processes and their findings to the accompanying examiner.
The first two tasks were randomized, with half the testers performing the task pertaining to a contact first, and the second doing the same with a keyword. Otherwise, the tasks were performed in the order in which they were described here. Before performing the tasks, each user was given a 5 minute demonstration of the prototype, followed by a 5 minute period of acclimatization to the software. This was done with a separate collection of documents.

After the tasks, the users answered a questionnaire with a series of statements about general and specific aspects of the VisMe interface to be rated according to a five point Likert scale (for example, “This example was: confusing 1 2 3 4 5 clarifying”, with 1 being the most confusing, 5 the most clarifying, and the rest of the values standing for something in between). While we chose a four point scale in the first usability tests to ensure users would make a decision regarding their satisfaction with the different topics, we opted for a more familiar five point scale in this second user evaluation. Still, the questionnaires cannot be taken as definitive measurements of the validity if the solution. Rather, they indicate, within the tested users, a tendency to be satisfied or dissatisfied with different properties and functionalities of the solution, as even if they are generally apathetic or even apologetic in regard to questions towards which they have no particularly strong feelings, if something truly bothers or satisfies users, they will make it clear in the relevant answers. As such, the analyses between the results of comparable topics across the two questionnaires can still be made without directly comparing exact numbers.

5.3.2 Results

As in the previous evaluation, we recorded success, completion time, and difficulty for each task (as rated by the users), as well as the answers to the questionnaire, and later performed a statistical analysis of that data. Bellow, we present the major findings of that analysis relative to four main groups: document retrieval, pattern recognition, exploration, and satisfaction. Again, these results are shown through averages (\( \bar{x} \)), medians (\( \tilde{x} \)), standard deviations (\( \sigma \)), confidence intervals (CI, 95%), and, in the difficulty measurements and in the questionnaires, modes. Likewise, we use Student’s t-tests, with a confidence of 95%, to determine the statistical significance of the differences between the recorded results of distinct tasks.

<table>
<thead>
<tr>
<th>Task</th>
<th>Average</th>
<th>Median</th>
<th>Standard Deviation</th>
<th>Confidence Interval (95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>16.3</td>
<td>16.5</td>
<td>7.2</td>
<td>4.5</td>
</tr>
<tr>
<td>1.2</td>
<td>36.3</td>
<td>27.0</td>
<td>26.4</td>
<td>16.3</td>
</tr>
<tr>
<td>1.3</td>
<td>7.3</td>
<td>5.5</td>
<td>5.3</td>
<td>3.3</td>
</tr>
<tr>
<td>2.1</td>
<td>16.2</td>
<td>15.0</td>
<td>7.8</td>
<td>4.9</td>
</tr>
<tr>
<td>2.2</td>
<td>27.9</td>
<td>29.0</td>
<td>6.3</td>
<td>3.9</td>
</tr>
<tr>
<td>2.3</td>
<td>8.5</td>
<td>6.0</td>
<td>6.3</td>
<td>3.9</td>
</tr>
<tr>
<td>3.1</td>
<td>25.1</td>
<td>27.0</td>
<td>6.5</td>
<td>4.0</td>
</tr>
</tbody>
</table>

Table 5.6: Second user evaluation, task times in seconds.

Document Retrieval

Obtaining the documents related to a keyword (task 1.3, \( \bar{x} = 7.3 \), \( \tilde{x} = 5.5 \), \( \sigma = 5.3 \)) and to a contact (task 2.3, \( \bar{x} = 8.5 \), \( \tilde{x} = 6 \), \( \sigma = 6.3 \)) are fast operations. These were also considered very easy tasks (task 1.3, \( \bar{x} = 5 \), \( \tilde{x} = 5 \), \( \sigma = 0 \), mode = 5; and task 2.3, \( \bar{x} = 4.9 \), \( \tilde{x} = 5 \), \( \sigma = 0.3 \), mode = 5). This is true regardless of the type of facet (there is no statistically significant difference), as the interface allows users to directly expand timelines filled with the documents from any keyword or contact in the visualization in a uniform
way. These times reflect only the act of obtaining the documents; they do not include the initial search for the desired facet as it was required in the document retrieval tasks of the first evaluation.

<table>
<thead>
<tr>
<th>Task</th>
<th>Average</th>
<th>Median</th>
<th>Standard Deviation</th>
<th>Confidence Interval (95%)</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>5.0</td>
<td>5.0</td>
<td>0.0</td>
<td>-</td>
<td>5</td>
</tr>
<tr>
<td>1.2</td>
<td>4.5</td>
<td>5.0</td>
<td>0.7</td>
<td>0.4</td>
<td>5</td>
</tr>
<tr>
<td>1.3</td>
<td>5.0</td>
<td>5.0</td>
<td>0.0</td>
<td>-</td>
<td>5</td>
</tr>
<tr>
<td>2.1</td>
<td>4.7</td>
<td>5.0</td>
<td>0.5</td>
<td>0.3</td>
<td>5</td>
</tr>
<tr>
<td>2.2</td>
<td>4.3</td>
<td>5.0</td>
<td>1.3</td>
<td>0.8</td>
<td>5</td>
</tr>
<tr>
<td>2.3</td>
<td>4.9</td>
<td>5.0</td>
<td>0.3</td>
<td>0.2</td>
<td>5</td>
</tr>
<tr>
<td>3.1</td>
<td>3.8</td>
<td>5.0</td>
<td>1.8</td>
<td>1.1</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 5.7: Second user evaluation, task difficulty, from hardest (1) to easiest (5).

We also tested the filtering mechanism through which users can combine several facets in the same timeline, particularly its application to document retrieval. Users were generally capable of performing this task successfully in under 20 seconds (task 3.1, \( \bar{x} = 25.1 \), \( \tilde{x} = 27 \), \( \sigma = 6.5 \)). Some users found it difficult and this was considered more complicated than directly retrieving documents from a single facet, but still easy (task 3.1, \( \bar{x} = 3.8 \), \( \tilde{x} = 5 \), \( \sigma = 1.8 \), mode = 5). This means that the combination of a keyword and a contact to obtain a document is a feasible and not overly lengthy or complicated operation.

**Pattern Recognition**

The ability to perceive the relative importance of keywords and contacts, and to follow their evolution over time is fundamental to being able recognize trends and patterns. Identifying the most relevant keyword and contact was found to be fast (task 1.1, \( \bar{x} = 16.3 \), \( \tilde{x} = 16.5 \), \( \sigma = 7.2 \); and task 2.1, \( \bar{x} = 16.2 \), \( \tilde{x} = 15 \), \( \sigma = 7.8 \)) and easy (task 3.1, \( \bar{x} = 5 \), \( \tilde{x} = 5 \), \( \sigma = 0 \), mode = 5; and task 2.2, \( \bar{x} = 4.7 \), \( \tilde{x} = 5 \), \( \sigma = 0.5 \), mode = 5), with no statistically significant difference between tasks.
Describing the evolution of these elements over time takes longer (task 1.2, $\bar{x} = 36.3$, $\tilde{x} = 27$, $\sigma = 26.4$; and task 2.2, $\bar{x} = 27.9$, $\tilde{x} = 29$, $\sigma = 6.3$) and is slightly more complicated (task 1.2, $\bar{x} = 4.5$, $\tilde{x} = 5$, $\sigma = 0.7$, mode = 5; and task 2.2, $\bar{x} = 4.3$, $\tilde{x} = 5$, $\sigma = 1.3$, mode = 5). Task 1.2, concerning the description of the evolution of a keyword over time, has a larger standard deviation than its contact related counterpart. Removing one clear outlier in this task’s recorded times, however, lowers its standard deviation to a considerably closer value ($\sigma = 8.3$).

**Exploration**

The extended and unrestricted exploration task is more subjective. Users were told to report any interesting findings and the level of interest for similar discoveries naturally varies from person to person. We also need to take into account that people may not have been comfortable exposing certain parts of their personal information, no matter how relevant to the study. It is, however, possible to gain valuable insights about the possibility of finding personally relevant trends and patterns with VisMe, along with a better understanding of how people use the interface, by observing the task in progress and by analyzing what information the users were able to gather.

All users were capable of identifying something interesting within their personal information, from singular events to more spread out communication patterns.

Two distinct users noted similar stories featuring the sudden appearance of a contact, followed by a strong presence for a few months, and ending with an abrupt disappearance, after which the person is never seen again.

One user expressed his surprise and frustration over the fact that most of his top keywords were coming from a few websites that sent him periodic automatic emails. He did not consider these to be spam, but he did admit he did not really care for their specific content. He also noted that it would be interesting to be able to filter out these emails, something which ideally should be possible with the current interface, were it not for the lack of exclusive filtering, a clear flaw that we will discuss later.
Six users described the visible appearance and ensuing constant presence of certain friends or colleagues. This was met with no surprise from the users, with the exception of one user who was, on one occasion, led to remember the few times when emails were sent to a particular contact, a friend with whom this person did not usually communicate online.

**Satisfaction**

**Note** These values are presented in a five point scale, from 1 to 5, with higher values being associated to greater degrees of agreement.

**Question 1** General satisfaction: $\bar{x} = 4.5$, $\bar{x} = 5$, $\sigma = 0.7$, mode = 5.

Users were generally satisfied with the system.

**Question 2** Graphical appearance: $(\bar{x} = 4.9, \bar{x} = 5, \sigma = 0.3, \text{mode} = 5$.

Users considered VisMe aesthetically pleasing.

**Question 3** Ease of use: $\bar{x} = 3.8$, $\bar{x} = 4$, $\sigma = 0.9$, mode = 4.

Users considered the system easy to use, although not very easy. Just as the results of the first questionnaire, this is a good indication that the interface facilitates the tasks under consideration.

**Question 4** Ease of comprehension: $\bar{x} = 3.6$, $\bar{x} = 4$, $\sigma = 0.8$, mode = 4.

The system was not found to be difficult to understand.

**Question 5** Ease of first use: $\bar{x} = 3.7$, $\bar{x} = 4$, $\sigma = 0.9$, mode = 4.

Again, beginning the use of the system, while not difficult, is also not very easy.

**Question 6** Ease of learning: $\bar{x} = 3.9$, $\bar{x} = 4$, $\sigma = 0.7$, mode = 4.

The users seem to have found it slightly easier to learn the system in this second evaluation. It is possible that the addition of tooltips contributed to this improvement.

**Question 7** Ease of remembering commands: $\bar{x} = 3.6$, $\bar{x} = 3.5$, $\sigma = 0.7$, mode = 3.

Remembering commands was not considered difficult, but it was not considered easy either, although it leans towards the latter.

**Question 8** Predictability of the result of an operation: $\bar{x} = 4.1$, $\bar{x} = 4$, $\sigma = 0.6$, mode = 4.

Users considered the results of the different operations to be predictable. Unlike the first usability tests, there does not seem to be a tendency towards unpredictability, something which is perhaps also connected to the addition of descriptive tooltips.

**Question 9** Clarity of the feedback after an operation: $\bar{x} = 3.9$, $\bar{x} = 4$, $\sigma = 0.6$, mode = 4.

Most users consider the feedback after an operation to be clear, again matching the previous questionnaire.

**Question 10** Adequacy of the number of steps required to complete a task: $\bar{x} = 4.8$, $\bar{x} = 5$, $\sigma = 0.4$, mode = 5.

The number of tasks required to complete a task was considered very adequate. This is a more positive result than the one gathered in the previous evaluation, and is possibly a result of the greater simplicity of the tasks, particularly of the document retrieval based on a combination of facets, which was facilitated by the newly added filtering mechanism.

**Question 11** Logic in a sequence of steps to complete a task: $\bar{x} = 4.6$, $\bar{x} = 5$, $\sigma = 0.5$, mode = 5.

Matching the previous questionnaire, the sequence of steps to complete a task was deemed logical, leaning strongly towards very logical.
Question 12  Ease of extracting timelines: $\bar{x} = 4.7, \tilde{x} = 5, \sigma = 0.7, \text{mode} = 5$.
Extracting timelines was considered very easy. These results are more positive than the one obtained before, which also revealed a general ease in this regard, although there was no change to this particular process.

Question 13  Legibility of the text in the timelines: $\bar{x} = 4.5, \tilde{x} = 5, \sigma = 0.7, \text{mode} = 5$.
Again, there was a clearer agreement regarding the proper legibility of the text in the timelines from the users of this second evaluation. The only change which could have had an impact in this regard was the addition of collision detection and handling to avoid overlapping timelines, and thus, overlapping text.

Question 14  Ease of use of the text search: $\bar{x} = 4.7, \tilde{x} = 5, \sigma = 0.5, \text{mode} = 5$.
The text search was considered very easy to use. Unlike the first usability tests, there was no task that strongly required the use of this functionality. As such, it was not extensively used by any of the users, although their limited experience with it was apparently very positive.

Question 15  Ease of filtering timelines: $\bar{x} = 4, \tilde{x} = 4, \sigma = 0.9, \text{mode} = 4$.
The newly added filtering mechanism was considered easy to use, although not very easy. Dragging and dropping different elements, while simple enough to be considered easy, may be too unconventional to elicit a very positive response in such a short experience. Although it required for one of the tasks, the functionality was not observed to be frequently used in other contexts.

Question 16  Ease of manual view control: $\bar{x} = 3.9, \tilde{x} = 4, \sigma = 1, \text{mode} = 4$.
Manual view control was considered easy, but some users did not find it clearly easy. This matches our observations and the results of the last questionnaire. Some users were not completely comfortable with the method of interaction but they still used it successfully without finding it obstructive.

Question 17  Adequacy of the automatic view control (occasional): $\bar{x} = 4.7, \tilde{x} = 5, \sigma = 0.5, \text{mode} = 5$.
The occasional automatic view control (centering the view on a timeline or expanding it to encompass all timelines) was considered very useful.

Question 18  Adequacy of the automatic view control (constant): $\bar{x} = 3.7, \tilde{x} = 3.5, \sigma = 0.8, \text{mode} = 3$.
The constant automatic view control was not considered clearly useful, in contrast with the occasional automatic view control. This matches our observations, as users rarely turned on the first option, but many made ongoing use of the second. While the constant automatic control of the view may be useful for quick, branching explorations, as it keeps the evolving structure on screen, it severely limits the control over what information to observe at any time. The functionality was presented as an option and was not enabled by default. As such, users did not make use of it beyond an initial experimentation, preferring instead to control the view manually, occasionally choosing to automatically focus on a timeline or encompass the entire structure.

Question 19  Usefulness of the timeline collision handling: $\bar{x} = 4.8, \tilde{x} = 5, \sigma = 0.4, \text{mode} = 5$.
Collision handling prevents overlapping timelines, which was generally considered very useful.

Question 20  Adequacy of the available facets: $\bar{x} = 4.9, \tilde{x} = 5, \sigma = 0.3, \text{mode} = 5$.
The choice of facets was considered adequate, although users may not have been aware of the implications of the different possibilities.

Question 21  Clarity of the meaning of the different buttons: $\bar{x} = 4, \tilde{x} = 4, \sigma = 1.2, \text{mode} = 5$.
The meaning of the buttons was considered very clear. In the previous questionnaire, users considered the specific meaning of the facet buttons to clear, although not very clear. It is possible
that the readily available description of each element in the interface through tooltips helped in this regard.

**Question 22** Usefulness of the tooltips: $\bar{x} = 4.6$, $\bar{\mu} = 5$, $\sigma = 0.7$, mode = 5.

Tooltips were considered very useful. No user was observed to turn off their display during the tests, nor did anyone complain about their intrusiveness.

**Question 23** Adequacy of the text search position: $\bar{x} = 4.6$, $\bar{\mu} = 5$, $\sigma = 0.7$, mode = 5.

Users agreed with the positioning of the search results on the top left corner of the screen, although no alternative was presented to them.

**Question 24** Adequacy of the help (tooltips) button position: $\bar{x} = 4.6$, $\bar{\mu} = 5$, $\sigma = 0.7$, mode = 5.

Similarly, placing the help button on the bottom left corner of the screen was met with no objections.

**Question 25** Adequacy of the automatic view control button position: $\bar{x} = 4.6$, $\bar{\mu} = 5$, $\sigma = 0.7$, mode = 5.

Still on the same token, the position of the automatic view control button on the top right corner of the screen was considered very adequate.

**Question 26** Adequacy of the use of color: $\bar{x} = 4.9$, $\bar{\mu} = 5$, $\sigma = 0.3$, mode = 5.

The use of color in the system was considered very adequate by most users, clearly a more positive result than the one obtained in the first questionnaire, although in that instance there were also no concrete complaints about this aspect.

**Question 27** Ease of selecting highlight colors: $\bar{x} = 4.6$, $\bar{\mu} = 5$, $\sigma = 0.7$, mode = 5.

Matching the results of the first questionnaire, the color selection mechanism was considered very easy to use.

### 5.3.3 Discussion

On the whole, taking into consideration the results of both evaluations, the system behaved as expected and provided the results we worked for. Trend and pattern recognition was shown to be facilitated by the visualization, as users could successfully detect the relative importance of a variety of contacts and topics as well as their evolution over time.

We verified that document retrieval is also well supported, as users have the ability to obtain timelines with the documents related to any element they find throughout their research. The filtering system also allowed users to combine multiple facets in the same timeline with relative ease to find the documents related to those facets. This functionality also has implications in pattern recognition (although that was not tested directly), and it was on the extended exploration task that a flaw in this functionality was made evident. Although it is easy to obtain a timeline with the elements related to an undetermined number and combination of different facets, it is currently impossible to do filter out the elements according to selected filters. The solution does not require an extensive reworking of the interface or of the underlying document access layer. To maintain the simplicity, dragging and dropping elements into a timeline will create inclusive filters, while clicking on an existing filter will toggle it from inclusive to exclusive and vice versa. Exclusive filters can be identified by a black background and white text, in the same manner that textual search results are highlighted. To remove a filter, users will have to double click over it.

Users were also capable of identifying trends and patterns of personal interest by observing the evolving relative importance of different keywords and contacts.

Thus, it has been shown that, by integrating information extracted from several different document types and presenting that information in interconnected timelines which users can freely manipulate to progressively explore their personal information, our approach potentiates the discovery of personally
relevant trends and patterns and facilitates the search and retrieval of individual documents based on
the partial memory of its associated information and surrounding context.
Chapter 6

Conclusion

A significant portion of our lives is documented throughout our daily use of computers. A competent visualization of that personal information could help us in many ways, from simple reminiscing about nearly forgotten moments, to efficiently locating lost documents based on their content and surrounding context. But existing solutions do not provide an integrated visualization of the content of the heterogeneous collection of documents wherein that personal information is contained. In pursuit of a suitable solution, we studied several possible interaction and visualization techniques, resulting in a system, VisMe, in which the content of different document types is extracted into representative keywords which, together with the connected contacts and document titles, are displayed in uniform timelines from which more timelines, pertaining to the facets one wishes to explore, can be extracted. A person’s evolving patterns of activity and communication are made visible, while the possibility of discovering the actual documents from keywords or contacts of interest remains throughout the exploration.

Usability tests validated VisMe’s approach to the problem, providing evidence for its ability to facilitate finding and retrieving documents based on the knowledge of different associated facets, detecting the most relevant keywords and contacts together with their evolution over time, and identifying personally relevant trends and patterns in the users’ own personal information. Displaying information in timelines makes their relative importance and how it changes over time clearly visible, and opening, keeping, and manipulating several timelines in the same canvas allows users to freely investigate several avenues while having access to the branching results of their explorations to make sense of all their personal information.

But there is still work to be done, whether one wishes to enhance the solution or to investigate different applications for the techniques we researched. Besides augmenting the filtering mechanism by adding exclusive filters, there are some underlying technical issues which can be worked on to improve the implementation and may require alternative solutions for visualization and interaction.

One is the direct access to the documents found through the visualization. The indexing application keeps a path of the individual files, but accessing the full email messages would require some alterations and it would leave us with an interesting choice regarding how to display those emails: either in the application itself or in an email application specified by the user. The former option would drag with it the problem of being a simple preview without the features one expects when browsing emails, unless we add significant complexity to its interface, while the latter can degrade the user experience if it is not handled correctly, since it could potentially require an erratic configuration and not very seamless transitions to other applications. Further research will be necessary to determine the best solution.

Also concerning the indexing application, its range is still limited, lacking proper handling of instant messaging logs, website, and, most notably in this day and age, of social networking sites. Still, the information coming from those sources could be formatted in terms of keywords and egocentric com-
munication networks, meaning the system is ready to accommodate it. This does not exclude potential improvements to take full advantage of all the particular features of each new source, as long as the coherence is preserved and the visualization is kept uniform.

Besides a more comprehensive indexation of personal electronic documents, we can consider the possibility that different sources of data, not necessarily personal, can also be visualized using the same basic principles of the visualization and interaction techniques developed for VisMe. As an example, we can imagine exploring the results of typical web search engines by iteratively dragging searches out of any visible word, starting with an initial search term and eventually developing into a rich, branching exploration.
Bibliography


