University of Porto
Faculty of Engineering

Rethinking Email Information Visualization: a case study

Pedro Daniel Cardoso Santos

July 2015

Scientific Supervision by

Ademar Aguiar, Assistant Professor
Department of Informatics Engineering

In partial fulfillment of requirements for the degree of
Master in Computer Engineering
by the EUR-ACE Programme

Contact Information:

Pedro Santos
Faculty of Engineering of University of Porto
Department of Informatics

Rua Dr. Roberto Frias, s/n
4200-465 Porto
Portugal

Tel.: +351 916 411 228
Email: ei10021@fe.up.pt
URL: http://paginas.fe.up.pt/~ei10021/pedrosantos

Acknowledgements

First of all, I'd like to thank my supervisor, professor Ademar, and my co-supervisor, professor Emília Costa, for being always available, all the support, motivation and for pushing me forward on this dissertation and to mailcube, on the person of João Anes, for proposing this challenge to me and for giving me all the conditions so far for me to achieve success with this dissertation.

I also want to thank to everyone that helped me during this dissertation, namely to the volunteers on the evaluation sessions - André, Miguel, Gustavo, Rui G., Diogo, Rui C., João, Daniel T., Daniel N. and Victor -, to mailcube's staff (mostly João Anes for the orientation, Bárbara for the usability tips, André Silva for the OS X "coaching" and Pedro for the brainstorming, awesome suggestions, ideas and design tips). Also to Marco and Myrto for the availability to help me with this document.

Then, I'd like to thanks to my parents and my sister for being an endless source of support and motivation throughout the years. All the words in the world are not enough to express my gratitude towards you.

Also, to all of my friends and family who have made this journey by my side, with whom I've shared the best and the worse. You really made *this* worth it!

And the last acknowledgement and biggest thank you goes to my grandfather Francisco, my role model and my best friend, the person who I most wanted to see me succeed. We made it Chico!!!

Thank you all!

Pedro Santos

"Se ser engenheiro fosse fácil, até eu era engenheiro.
Por isso, não desistas."

- Francisco Lopes dos Santos

This page was intentionally left mostly blank.

Abstract

How many times has the phrase "An image's worth a thousand words" been used in the daily life? Indeed, a graphical representation of anything that surrounds us is much better assimilated by the human brain rather than a textual representation, because the vision is the sense that carries more information and faster to the brain. Therefore, this principle may be applied to anything, namely, in a scientific area where scientists have to deal with large sets of data. These datasets were visualized in a less capable way until a few years ago when the area of Information Visualization started to emerge and alerting the specialists to the fact that, with a set of techniques and principles, the cognitive process of acquiring information can be facilitated in such a way that the user can acquire more information and get a better insight over the data he's visualizing, at the same time, that the effort to perform this task is reduced.

The aforementioned science, Information Visualization, can be applied to any field of study. And a particular field that can be tremendously improved with this relation is the email environment. As it is widely known, the email use is massified and it's one of the most used means of communication within several contexts. Still, besides this massification, the structure of email is almost the same as it was 40 years ago. This points to a need of improving this system, analyzing first what are the actions that users execute, what are the user's needs and then adapting the email to the modern user.

This dissertation has three distinct parts.First, a description of this still not mainstream science that is Information Visualization. Next presents the results of a research over the email environment, pointing to its main gaps and the new features that users associate to it. Finally, it discusses the two previous topics - using information visualization techniques to solve a subset, suited for this dissertation's timeframe, of email's problems - trying to achieve a solution that pushes the email system forward in the direction of making the use of the email, once again, a more pleasant and effective experience to the user. The work is applied to a real email client - currently under development - as a case study so that all the work here developed can be validated in a real life scenario, with real users, having feedback and with that to improve the prototypes thus finding a better fit with the real users' needs.

Resumo

Quantas vezes a expressão "Uma imagem vale mais do que mil palavras" é usada no dia-a-dia? Uma representação gráfica de qualquer objeto é mais facilmente assimilada pelo cérebro humano do que a sua representação textual dado que a visão é o principal meio de transporte de informação para o cérebro, tanto em quantidade como em velocidade. E este principio pode ser aplicado em qualquer área. Nomeadamente na área da ciência onde cada vez mais é necessário analisar grandes quantidades de dados e tirar conclusões relevantes com eles. Estes conjuntos de dados eram visualizados de uma forma muito pobre até à pouco tempo, altura em que a área de *Information Visualization* surgiu. Com a aplicação de um conjunto de técnicas e de principios o processo de aquisição de conhecimento pode ser facilitado de forma que o utilizador consiga adquirir mais informação e tirar melhores conclusões ao mesmo tempo que o seu esforço cognitivo diminui significativamente.

Uma área onde a *Information Visualization* pode ter impacto tremendo é na área do email. Como é sobejamente sabido, o email é um sistema de comunicação massivamente utilizado e em variadíssimos contextos. Ainda assim, o email, de uma forma geral, está parado no tempo, estando ainda assente na sua estrutura original de há 40 anos atrás. Isto indica que há uma necessidade de atualizar este sistema, analisando numa primeira fase quais são as ações que os utilizadores executam, quais as suas necessidades e depois adaptar o sistema, tendo em conta estes dados.

Esta dissertação tem três partes distintas. A primeira detalha os principais conceitos associados a Information Visualization. A segunda parte investiga o sistema de email, apontando para as suas principais lacunas e os novos usos que os utilizadores associam ao email que não estavam planeados na sua especificação original. Finalmente, a terceira parte discute conjuntamente os dois tópicos anteriores de forma a tentar atingir uma solução que impulsione o email a dar um passo em frente no sentido de voltar a fazer com que este sistema seja uma experiência agradável para o utilizador. Os resultados do trabalho são aplicados a um cliente de email (em desenvolvimento) como caso de estudo onde os protótipos desenvolvidos são integrados para serem validados pelos seus utilizadores, sendo depois a opinião destes tida em conta para a melhoria destes protótipos.

Contents

1 Introduction **1**

1.1 Context . . . 1

1.2 Motivation . . . 2

1.3 Goals . . . 2

1.4 Dissertation structure . . . 3

2 Information visualization **5**

2.1 Definition . . . 6

2.2 Fundamentals . . . 9

2.2.1 Visual properties . . . 9

2.2.2 Information exploring process . . . 12

2.3 Conclusions . . . 16

3 Email analysis **17**

3.1 Modern usages . . . 18

3.1.1 Conversational threads . . . 18

3.1.2 Team/Project management . . . 19

3.1.3 File transfer and version management . . . 19

3.1.4 To do list . . . 19

3.2 Email usage gaps . . . 20

3.2.1 Threads . . . 20

3.2.2 Information separation and organization . . . 22

3.2.3 Lack of context . . . 23

3.2.4 Process automation . . . 23

3.2.5 Social Email . . . 24

3.2.6 Email archiving . . . 24

3.3 Solving email gaps with Information Visualization . . . 25

4 Problems and Approach **27**

4.1 Problems addressed . . . 27

4.1.1 Thread visualization . . . 28
4.1.2 Individual message visualization and comparison . . . 28
4.1.3 Thread participants analysis . . . 28
4.2 Approach . . . 28
4.3 Technologies . . . 29
4.3.1 Visualization . . . 29
4.3.2 Application . . . 30

5 Prototypes **33**
5.1 Thread visualization . . . 35
5.1.1 Integration within the email client . . . 35
5.1.2 Tree graph . . . 35
5.1.3 Timeline . . . 41
5.1.4 Chat . . . 45
5.2 Email comparison and visualization . . . 47
5.2.1 Identifying inline messages . . . 48
5.2.2 Email clients/service identification . . . 50
5.2.3 Principles and relation to Information Visualization . . . 55
5.3 Sociogram . . . 55
5.3.1 Concept . . . 56
5.3.2 Principles and relation to Information Visualization . . . 56
5.3.3 Implementation . . . 59
5.3.4 Main features . . . 60

6 Prototype Validation and Results **61**
6.1 Methodology . . . 62
6.2 Process . . . 62
6.2.1 Pilot test . . . 63
6.2.2 Personas . . . 64
6.3 Results . . . 64
6.3.1 Tree and Timeline . . . 64
6.3.2 Chat . . . 67
6.3.3 Thread Reconstruction . . . 72
6.3.4 Sociogram . . . 75

7 Conclusions **79**
7.1 Future work . . . 80
7.1.1 Tree & Timeline . . . 80

7.1.2 Chat 81
7.1.3 Thread reconstruction 81
7.1.4 Sociogram 82

References **83**

Appendices **85**

A Tree & Timeline Evaluation Script **89**

B Tree & Timeline Questions Form **93**

C Tree & Timeline Evaluation Results **97**

D Chat Evaluation Script **99**

E Chat Evaluation Questions Form **101**

F Chat Evaluation Results **105**

G Thread Recreation Evaluation Script **107**

H Thread Recreation Questions Form **111**

I Thread Recreation Evaluation Results **115**

J Sociogram Evaluation Script **119**

K Sociogram Evaluation Questions Form **123**

L Sociogram Evaluation Results **127**

M Thread to validate reconstruction **129**

N Threads used for evaluations **133**
N.1 Messages (Graph & Timeline 1) 133
N.2 Messages (Graph & Timeline 2) 134
N.3 Messages (Thread Recreation) 135
N.4 Messages (Chat) 137
N.5 Messages (Sociogram 1 - Without sociogram) 138
N.6 Messages (Sociogram 2 - With sociogram) 139

List of Figures

2.1 Dr. John Snow's map to show cholera clusters. London 1885. 7
2.2 Minard's representation of Napoleon Bonaparte's Russia crusade casualties. 8
2.3 Example: Finding the element that distinguish from the remaining - red ball within blue balls [FWS+12] 11
2.4 Example by Escher: Scene where it's hard to distinguish background from elements and even elements between themselves. 11
2.5 Shedroff's Diagram on information and knowledge acquisition process. . . . 13

3.1 Email thread with a tree visualization proposed by [RGM+] 22
3.2 Email thread with arc visualization proposed by [KWS04] 22

5.1 Conventional graphical representation of a tree structure. 36
5.2 First tree representation concept 38
5.3 Final result of tree view prototype, at the left, and messages content, at the right. 39
5.4 Final result of tree view prototype, at the left, and messages content, at the right, with a message selected. 40
5.5 Timeline prototype example 42
5.6 Example of the see more button. Gmail web client. 49
5.7 Example of the see more expanded button, with 1 message in the inline content. Gmail web client. 50
5.8 Sociogram. 58
5.9 Sociogram Expanded 58
5.10 Sociogram with node *Andrew* selected. 59

6.1 Users' opinion on difficulty to perform evaluations tasks with old and new visualization. 65
6.2 Design of reading view after improvements. 68
6.3 Users' opinion on difficluty to distinguish messages from self from messages from others. 69

6.4 Users' opinion on difficulty to send messages with this prototype versus standard email clients . . . 70
6.5 Users' opinion on chat prototype usefulness. . . . 70
6.6 Final version of the chat prototype. . . . 72
6.7 Visualization of the forwarded thread in an web email client . . . 73
6.8 Users' opinion on difficulty to extract information from the forwarded thread by using the web version and by using the reconstructed thread. . . . 74
6.9 Users' opinion on usefulness of thread reconstruction . . . 74
6.10 Users' opinion on difficulty to complete tasks when not using sociogram and when using it. . . . 76
6.11 Users' opinion on sociogram's usefulness. . . . 77

7.1 Design for tree and timeline prototype (main reading view) next iteration. 80
7.2 Design for chat's next iteration. . . . 81
7.3 Design for sociogram's next iteration. . . . 82

Chapter 1

Introduction

1.1 Context 1
1.2 Motivation 2
1.3 Goals 2
1.4 Dissertation structure 3

This dissertation entitled "Rethinking Email Information Visualization" was proposed by the company "Mailcube, Lda" and its outcome - prototypes that solve the addressed problems - intends to validate the proposed solutions, - partially integrating them with mailcube's application now - in order for a full integration of these prototypes and ideas with the project.

This document presents the state of the art on the areas of Information Visualization and Email and how they can be related to each other and on the other side present and document the prototypes produced during this dissertation along with the results of their evaluation and user validation.

1.1 Context

This dissertation emerges on a time where every single detail counts if a company wants to deliver a successful product to the customer. There is a growing concern about delivering the best possible product to the customer and each product must be different from its competitors, must have some highly valued differentiators and must facilitate the users' actions.

The area of Information Visualization intends to analyze how people consult information in a certain application or context and how that action can be optimized so that the user,

with a minor cognitive effort, can extract the maximum amount of knowledge. Size or position, for example, of every "grain" of information has a huge importance when design is user-oriented. This way of thinking already has some repercussions on some areas of study. Nevertheless the email, which is stopped in time since a few years ago with very few upgrades, hasn't been revolutionized (yet).

1.2 Motivation

With the mass increasing growth of the email use for many different objectives rather than it was originally created for, being for many people today more a habitat than a work tool [DB01], it was necessary for the email clients to adapt and embrace new different needs. But what really happens today is quite different [Gui11]. Email clients and the email environment are not target of a major structural update for a long time. This dissertation aims to improve users' email experience by adding Information Visualization techniques to change the paradigm on some features of the email or to enhance users' experience in other areas so that the email client can be prepared for the needs and the demands that users have nowadays.

Also, having studied the time that people lose handling email daily [JDW03] and each person having their own experience in mind, one can conclude that this theme is a serious threat to personal productivity and to an effective way to perform some tasks in an era where people are always on a run, as they need to access information anywhere and anytime.

The previous paragraphs introduce the two main themes that motivate this dissertation, which in Layman's terms are improving personal productivity while studying the problems associated to email so that disruptive features can be added to an email client that can aid users to solve these problems.

1.3 Goals

There are two main goals to achieve with the development of this dissertation which are as follows:

- **Identify areas that need improvement and relate them with information visualization** : study email usage gaps along with Information Visualization, relate both areas and identify improvement opportunities;
- **Choose a subset of improvements suitable for development within this dissertation's timeframe and improve** : after the first phase is concluded, choose

a subset of these identified opportunities, idealize, implement and validate a solution to solve each problem in this subset, gathering users' feedback and improve the prototypes based on it.

1.4 Dissertation structure

This document is divided into seven main chapters. First the scope, goals and what to expect from the work is presented.

Then, chapters 2 and 3 are dedicated to overview the state of the art in email and information visualization. They contain a description of what is Information Visualization and the results of the investigation process on the problems of email's usage in modern days.

Chapters 4, 5 6 describe the practical outcome of this dissertation. The choice of which prototypes to implement - and consequently which problems to solve - will be presented and justified, along with a detailed explanation of each prototype and its validation process and results.

Finally, chapter 7 presents the conclusions that were achieved with this dissertation, along with the future work.

Chapter 2

Information visualization

2.1 Definition 6
2.2 Fundamentals 9
2.2.1 Visual properties 9
2.2.2 Information exploring process 12
2.3 Conclusions 16

A good scheme is worth more than a great speech

Napoleon Bonaparte

Vision had always a preponderant role and was treated very respectfully. One good example is ancient Egypt where it was thought that the world was born from the *Quadxa* (also known as *sacred eye* or *divine eye*), which was considered a sacred symbol that was the source of knowledge and prosperity [Cos11]. Another good example of this importance that vision had since the "beginning of times" (seen in the same source) is that, in greek mythology, there was a prince - Argos - who had his body covered with eyes that were not all closed at the same time and became a symbol of permanent awareness. A more concrete and popular example is the phrase introduced by Saint Thomas in the bible and that's commonly used nowadays *«Ver para crer»* which translated to English is something like "See to believe".

Allied to this vision's importance is image as a mean to transmit, process and acquire information. As Costa states in [Cos11] image has a fundamental role in showing and presenting information that a speech cannot transmit. In fact the french expression *«Je ne sais quoi»* refers to that beauty that cannot be put into words, which has such a dimension that has to be experienced and is not attainable by a simple textual description.

Visual representations of data - whether this data is a set of statistical data about a company's performance or the description of a beautiful landscape in the Alps - are present in every day situations because they help people understand the world around them. It's under this premise that the area of Information Visualization emerged. It is an area that's growing in importance, entered the ACM classification list in its last iteration (2012) [ACM12], within the scientific environment and that can assume a decisive role on the success of newer products, distinguishing them from its competitors.

This chapter aims to define Information Visualization, to illustrate its goals, which are its fundamental techniques and practices and how they can improve user's experience and help with the cognitive process of information acquisition.

2.1 Definition

Information Visualization is defined on [CMS99] as "the use of computer-supported, interactive visual representations of data to amplify cognition.". This means that we are before a science that applies a set of techniques to information and organizes it in certain data structures so that the cognitive process of information acquisition can be improved and the user's cognitive effort is smaller but at the same time can extract more useful information from a data set.

Information Visualization, fundamentally, aims to aid the user exploring data sets (in any form - data tables, texts, or any other format that stores information) and extracting useful information [Kei02] through its manipulation whether applying techniques or displaying the data in different structures at the same time that minimizes the cognitive effort done within this process.

One of the greatest examples of giving a visual representation to a set of data is Dr. John Snow's dot map in 1885, which can be seen in figure 2.1. This map represents the major clusters of cholera in London. This intended to show that cholera was being spread through ingestion and not by inhalation, as it was believed at the time. With this representation it was obvious for the authorities that the placed where more deaths were occurring was near the Broad Street's pump.

Figure 2.1: Dr. John Snow's map to show cholera clusters. London 1885. [1]

Another good example is Minard's representation (on 1896) in figure 2.2, on the casualties of Napoleon Bonaparte's Russia crusade. This image from the end of the 19th century represents the decrease in the number of soldiers (that died in battle) in Napoleon Bonaparte's Russia campaign since their departure, having their fight in Russia as a middle point and then their return to the original position.

[1] Source: https://irevolution.files.wordpress.com/2009/08/cholera-snow-map.jpg. Last accessed 14-June-2015.

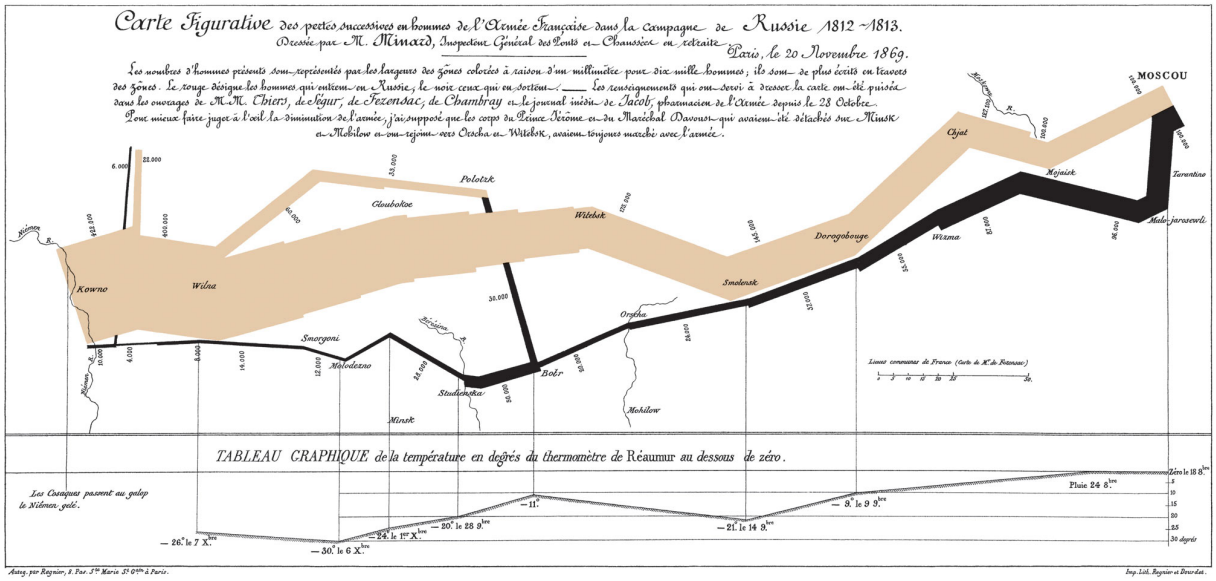


Figure 2.2: Minard's representation of Napoleon Bonaparte's Russia crusade casualties. [2]

These examples show two situations where graphic visualizations of statistical information can be much easily perceived by the brain.

Information Visualization usage results aren't easily measurable [FWS+12], because there's no unit to measure the effort that the user has to make to extract some kind of knowledge out of some set of information. This is a complex process where one cannot say that it is going to apply a certain technique in order to achieve a concrete goal - like getting a result to an equation or discover how many bodies exist in a room through computer vision, for example. The results are more abstract in a sense that the correct application of Information Visualization techniques should help the user on gaining insights on some set of information or exploring from different points of view the same data and, as said before, it is hard to define success within this context because it is not an objective goal, it's a subjective one.

Nevertheless, Tufte [TGM83] (and as [Cos11] also states) indicates a set of ideas that, if applied, shall increase the probability of success of an Information Visualization prototype. These ideas are as follows:

- Present information undistorted relatively to its original state;
- Lead users' focus on the content and not on the methodology, design or process to display it;
- Keep representations' coherence independently of data's volume;

[2] Source: https://eagereyes.org/wp-content/uploads/2012/08/Minard-Napoleon-Scan.png. Last accessed 14-June-2015.

- Encourage (or at least make possible for) the users to compare distinct parts of the information;
- Present the information using mechanisms that softens the visualization of data, providing distinct degrees of detail;
- Integrate visual representation of information along with other representations (textual or statistical).

Ultimately, success and usefulness of the application of Information Visualization is measured by the opinion of the user on how the knowledge acquisition process was effectively improved and facilitated after the implementation of these visualization techniques, within the specified context.

With this, one can then conclude that the best scenarios where Information Visualization can add value, in the user's perception, are the ones when there's information to be explored - going from a top level overview to a detailed visualization, leading to a better understanding of the set as a whole - or this information can be hierarchically organized to show the relation between elements, for example.

2.2 Fundamentals

After a detailed definition of Information Visualization has been presented, the next step is to introduce its main techniques and general rules to approach a certain problem.

2.2.1 Visual properties

According to [FWS+12] vision is the main mean of transport of information to the brain, because it has the largest bandwidth for the information to flow - around 100Mb/s, as stated on [War04]. There are several theories that study the vision and the way brain processes the information it gathers but the two main ones studied and whose principles were applied, due to the fact that they study, mainly, the timing and effectiveness of vision decoding and understanding information, were :

1. **Preattentive processing theory** [Tre85] - which states at a lower level of abstraction what are the most effective visual shapes / features can be effectively processed by the brain. This, in layman's terms, means that it's studied how an object can be highlighted from its environment when users have their first look, the moment before the user focus his attention in the analyzed object [Hea96], which is around 200 milliseconds. This process assumes another degree of importance in visual analysis

of data due to the fact that allows for a quicker and effective comprehension of data and the more relevant elements. Ware [War04] says that there are four main characteristics of an object that trigger this preattentive attention, which are its color, shape, movement and spatial location. Figure 2.3 shows how an object is clearly distinguishable from the remaining ones due to its different color. Then the attentive process goes into action in order to facilitate information highlighting during a more careful reading.

2. **Gestalt theory** [Kof35] - A theory that goes back around 100 years, explains how the brain decodes and interprets a visual object on a closer (after the preattentive vision takes place) look. Gestalt theory states that the perception is a whole and that cannot be separated in parcels, as Viennese philosopher Von Ehrenfels showed in [Ehr90] - where he states that, if 12 individuals listened to a single note of a melody and summed up their experience, they wouldn't have the same experience as an individual that listened to the whole melody. Gestalt theory defines three basic concepts with them being the field - which is the scene where the several phenomena will occur and that should be neutral to those phenomena -, structure - which is the aggregation of all the elements that compose the scene in an ordered and related way such that each element is inseparable from the rest in order for the scene to maintain its meaning - and shape - which is something like a glue that puts field and structure together, i.e. the way how the set of perceived elements is organized through a set of factors and attributes related to the sensations that users will feel when looking at it, like color or shape of the object itself. It is also stated that some properties like color and shape can, within some limits, be processed faster by the brain rather than using another visual properties. A good example of what can be done by not separating these three concepts between each other can be seen on figure 2.4. Here it's hard to distinguish background from the content of the scene itself and even distinguish objects between themselves.

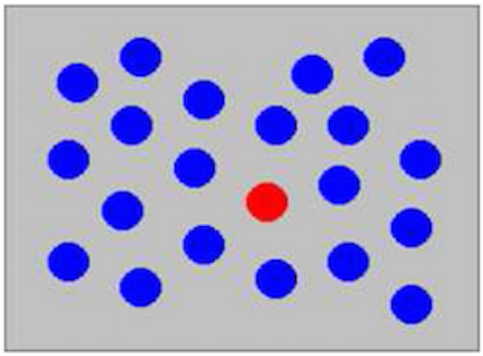


Figure 2.3: Example: Finding the element that distinguish from the remaining - red ball within blue balls [FWS+12]

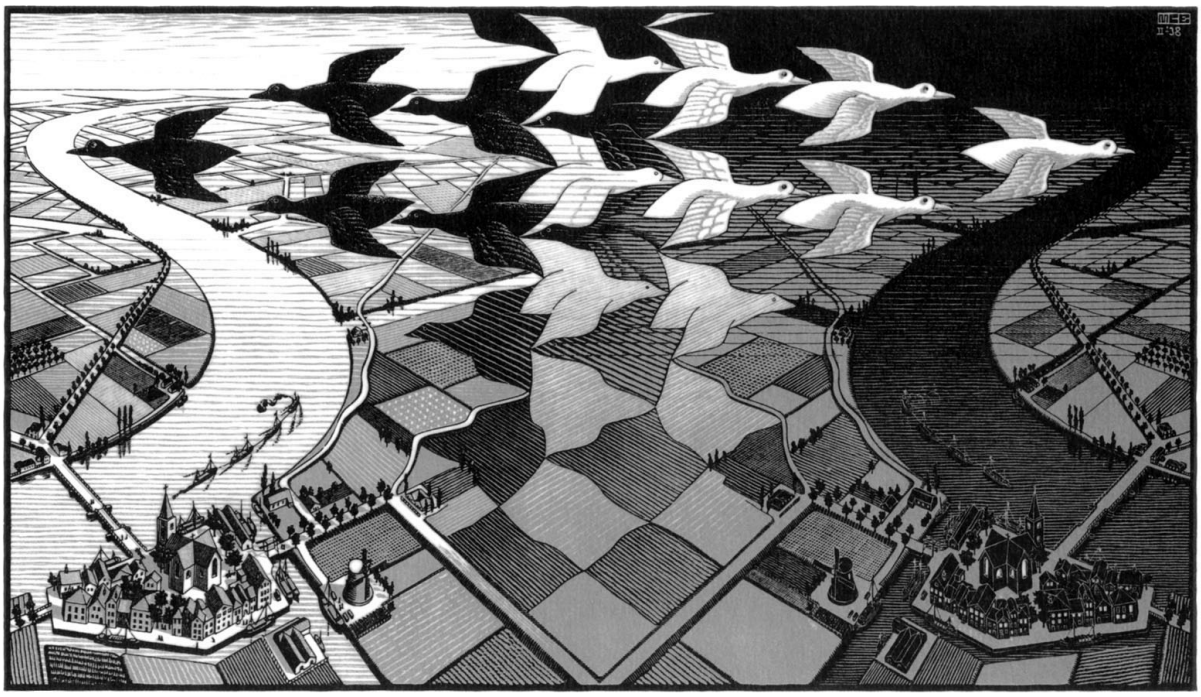


Figure 2.4: Example by Escher: Scene where it's hard to distinguish background from elements and even elements between themselves. [3]

Also, according to what is stated on [War04], there are some principles that, if followed and applied correctly, can help the brain making some assumptions right away and make the interpretation of information easier. These principles are:

1. **Proximity** - If objects are close to each other they tend to be associated as similar;
2. **Similarity** - Similar objects usually are together so they can be understood as identical;

[3] Source: http://britton.disted.camosun.bc.ca/escher/day_and_night.jpg. Last accessed 15-June-2015.

3. **Continuity** - Elements that are visually linked between them tend to be grouped together or assume they're somehow related;

4. **Symmetry** - Two items that are visually similar are likely to be perceived as a whole and not two different items;

5. **Closure** - A closed contour is perceived as an object;

6. **Relative Size** - When there's a bigger object acting as a background, smaller components of that object's pattern tend to be perceived as smaller objects;

These principles, even though in a very abstract and generic way help understanding how the cognitive process works on a set of objects, combined with techniques like the ones that are going to be described next, give a powerful starting point to analyze a problem.

2.2.2 Information exploring process

Peter Simlinger, director of the International Institute for Information Design, once said that *«Data transformed into high-quality information empower people to attain goals»*. This means the raw data that's collected from countless sources only has value when this data has to have a context and a meaning, so that it can then be analyzed and transformed into useful information and then knowledge. Shedroff diagram (figure 2.5 helps understanding this process. The meaning of each one of these phases, according to Shedroff [She01] is explained next:

- **Data** - Contextless and raw elements that act as a starting point for the analysis. An example is that the number 2407943 might be a date or a batch identifier [Cos11];

- **Information** - Phase where data has some kind of metadata associated to it, meaning that it has some kind of context and its meaning can be extracted;

- **Knowledge** - Phase where the user knows what the information's about and has some kind of interest in it or knows what to do with it. This is what distinguishes information from knowledge. This knowledge is personal, due to the fact that it's associated with people's experiences with the data;

- **Wisdom** - Wisdom is the last step on information acquisition process. This means that the user knows what the data's about, has interest in it, knows what to use it for and it's capable of identifying patterns and relate different pieces of information in order to generate more information (in a wider sense).

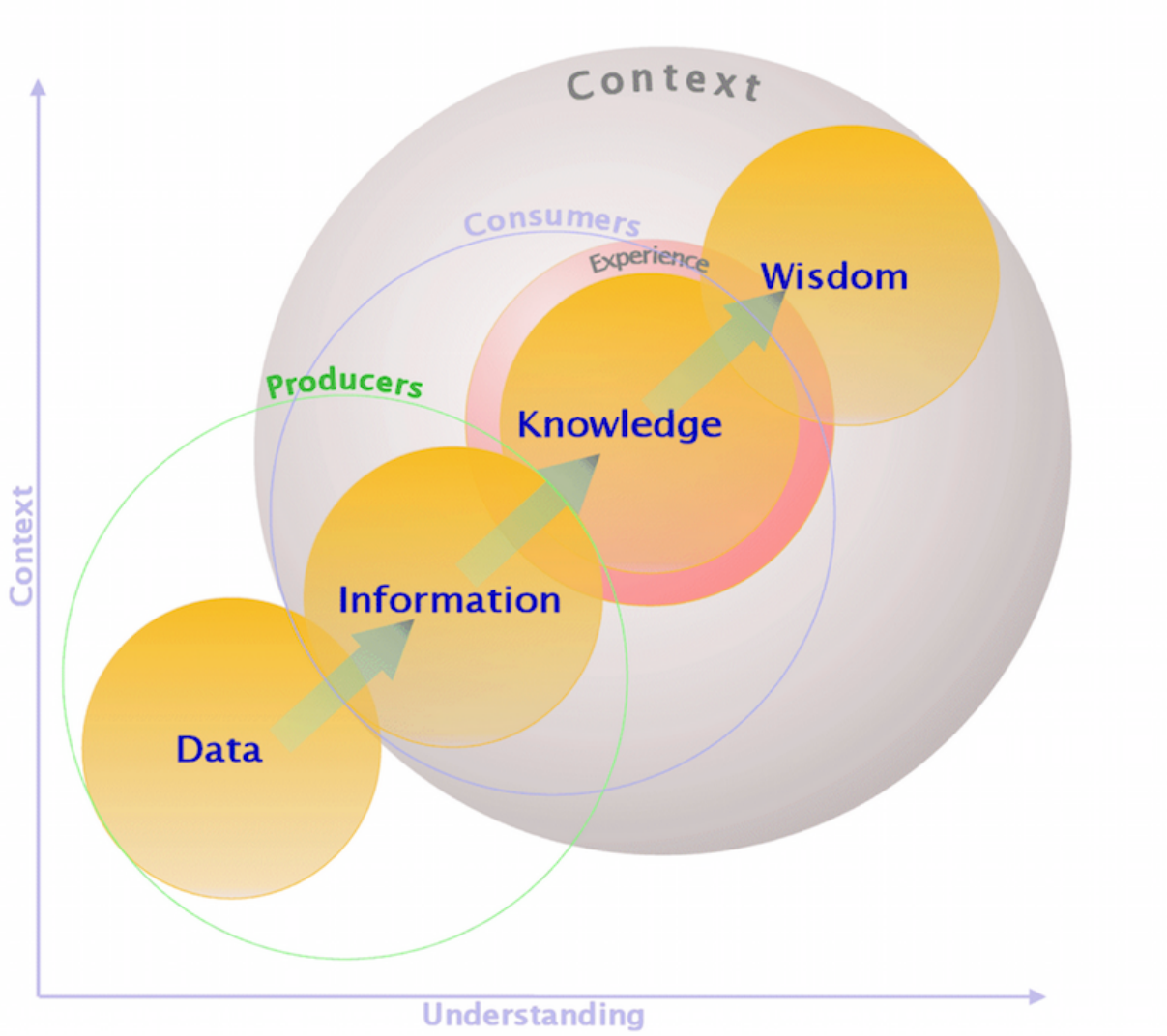


Figure 2.5: Shedroff's Diagram on information and knowledge acquisition process.

There are other theories regarding this information acquisition process in an abstract way like Colin Ware's [War04], Card et al.'s [CMS99] for example but their idea is similar - Data is acquired raw and then, using certain mechanisms (might be or not graphic) information is gradually perceived.

On a more concrete approach regarding visual tools to explore data, on [Shn] the authors use the expression "Overview first, zoom and filter, then details-on-demand" as a golden rule to follow when applying Information Visualization techniques on a certain system. For this, they suggest a set of seven actions to perform over a dataset, being them:

1. **Overview** - when presenting a dataset to the user, the first step is to give a general overview of this data so that the user can get a first perception of the universe that he's analysing;

2. **Zoom** - When presented with the overview the user should have the possibility to zoom in a portion of the set to analyze it properly or get a better insight of that particular "area" of the data set, to know if it has the potential to answer to his questions;

3. **Filter** - The user should also be able to filter dynamically the data available with some criteria reducing that way the size of the dataset presented and also matching the previously referred criteria;

4. **Details-on-demand** - After choosing what items are the most relevant within the specified context and that can provide answers to the questions on the table, the user must be able to view a detailed view of an item or a restricted set of items that are interesting in the user's point of view;

5. **Relate** - Relating items, showing their potential relationship or selecting items that have common attributes, for example;

6. **History** - Keep track of the user's actions so that he can either undo a certain action or set of actions or understand which was the followed path until achieve a certain point in the information exploration and acquisition. Most of the times is not only important the results one can get but also how he got to those results;

7. **Extract** - After having selected the set of information that the user considers useful, he should be able to extract this data into some format (external file or printing, for example) or either the query used to get that set (related to the previous point on the list, History);

Not all of them have to be applied in the same context because they might not be needed in that context, but this whole package of actions can help improving the perception that the user has over a dataset and therefore extract useful information quicker.

In the same paper, the authors also define seven data types (structures in which the data is or may be organized) and helpful techniques to each structure to help making this information understandable. They are as follows:

1. **1-dimensional** - This describes a sequence of text, organized as string. Main issues and respective solutions include finding a piece of relevant information in the middle of the string, which can be highlighted with different colors. Also, how to apply the methods mentioned above (overview, filter, etc) is an issue;

2. **2-dimensional** - This dimension represents planar data which only can be explored within 2 dimensions - like maps, for example. Main problems addressed in this data

structure is to explore multiple layers of 2-dimensions or paths/type of relationship between two items;

3. **3-dimensional** - Dimension that modulates most of the real-world objects, having these a volume (3 dimensions) and relations between each other as well, which may include occlusion, an object being inside other, among other examples;

4. **Temporal** - Temporal structures, as the name itself indicates, are used to display evolution through time of a dataset belonging to a certain constant domain - a patient's health records, or data about a project management, for example. The biggest problems on these data types are, for example, to get the start and end date of an item if there are overlaped items.

5. **Multi-dimensional** - Multi-dimensional datasets are composed, normally by objects which have N attributes and then it's generated a N-dimensional space in which objects are points on that space. Visualization techniques can be turning that into a 2 or 3 dimensional dataset, being the user able to change the dimensions that are displayed in real time being that way a very interactive feature. Nevertheless navigate in these structures can become confusing. Nevertheless, implement the techniques mentioned above is almost straightforward, as long as the developer understands the context;

6. **Tree** - A tree is a structure which has a root node and then, each node has children. A node can only have one parent but multiple children. This kind of structure is very usefull to represent hierarchies or nested information, for example. Indented outlines of different colors for each level of the tree can be useful techniques to apply to this structure;

7. **Network** - This network type is very similar to a graph structure, where nodes exist which are connected to each other via edges. This is a very powerful structure which can be used to show relationships between many elements with multiple connections between them. There is a lot of literature around these structures and how to handle its visualization such as [HMM00], for example.

These data structures embrace a wide set of possible uses and, having each one of them detailed on the paper mentioned above, it's a good starting point to organize the data set of a problem into a structure which has some standard techniques associated, as soon as the dimension of it is analyzed.

2.3 Conclusions

There are some conclusions that can be easily taken from this chapter, namely:

1. The main goal of Information Visualization is to enhance the information acquisition over a dataset process reducing at the same time the cognitive effort done by the user;
2. Information Visualization can be a valuable asset improving the usability and the user experience with an application where its principles are applied;
3. It's not easy to measure the success of the application of Information Visualization within a certain context rather than the users' opinions;

Chapter 3

Email analysis

3.1 Modern usages 18
3.1.1 Conversational threads 18
3.1.2 Team/Project management 19
3.1.3 File transfer and version management 19
3.1.4 To do list 19
3.2 Email usage gaps 20
3.2.1 Threads 20
3.2.2 Information separation and organization 22
3.2.3 Lack of context 23
3.2.4 Process automation 23
3.2.5 Social Email 24
3.2.6 Email archiving 24
3.3 Solving email gaps with Information Visualization 25

The first part of this chapter aims to present the modern usage of the email in a bipartite way. In the beginning the step we take is to analyze the new tasks that users nowadays associate to the email but that weren't planned on its original specification. The next one is to identify gaps in the email usage and in the modern email clients, whether they're associated to the original use of the email or to its newer roles. Most importantly it should be clear why email clients are failing to deal with these gaps and how much room for improvement there is to the area of email clients.

The second part of this chapter is, after this identification of new uses and gaps that can be filled with a new approach in the email environment, and following this last topic, a mapping between all of this information about email and Information Visualization will

be presented and detailed in such a way that it going to be clear in the value that can be added by applying this field of studies to the problems described.

3.1 Modern usages

This section intends to analyze, as defined by [WS96], email overload, meaning that's going to be described a set of functionalities that nowadays users associate to email despite they were not planned in the original scope of email. This email overload tends to let the user disoriented in his email environment turning what should be a standard task in a nightmare of data to analyze, filter and process. These new usages often arise from the embeddedness that email provides, because one user can put together a different set of actions into a single place, even though the practical usage is not the best, this embeddedness is worthy to the users.

3.1.1 Conversational threads

Firstly designed as an asynchronous mean of simple communication by exchanging single messages, email quickly started being used with a conversational purpose, generating that way a big volume of related emails (sequential replies between the intervenients) in a message-answer sequence, for example as the study presented on [BDHS03] proves.

This functionality, back in the days, brought several problems like identifying which emails belonged to the same conversation for example, but as stated on [FBGS06], several advances have been done in this area throughout the years. Nevertheless, there is still some work to do within this scope, namely the lack of context in some situations and a user-friendlier interface to give an overview of a thread and relation between its several messages. If a user is exchanging emails frequently in a short period of time or has multiple replies to the same email between two or more intervenients - can be a chat between a couple or a discussion between the members of a software development team, for example - , there's a high probability that it's a conversational thread. Size of these threads can evolve very quickly and if the user is absent from his email environment for some time, this thread can rise in size which will translate in many emails to catch up as soon as the user goes online. Also, the lack of context can make the process even more difficult. A user may think that a conversation is done and send it to the trash, and when he receives a new email of thread he's contextless - this problematic will be discussed ahead. But, with some tweaks in this functionality, this process can be softened and be more pleasant to the user. These tweaks and problems will be further detailed on the email usage gaps section.

3.1.2 Team/Project management

Following the last subsection, nowadays email is being used to communicate between work teams, assigning responsibilities, spreading messages among the team and many other management related tasks. This kind of communication, effective for informal purposes but weak for more formal data - like responsibility assignment for example-, is treated as it was a regular email, which can bring more trouble to the manager because he has to collect information from the mass of emails that are received daily and compile it all, as stated in [SM06].This micro/small management has a lot of space to improve. If instead of just a text where the manager says that element A has task B to execute and after a while element A says that the task is complete, there's a more formal way to assign these tasks, maybe integrating them with other services (as some To-do tools partially do now) or even better, integrate this sub-type of email directly into the email client, being the user able to export it to use that data after in another context.

3.1.3 File transfer and version management

One more functionality that's deeply massified is the file transfer / sharing via email as stated on [DB01]. This file sharing can have several purposes like a simple file sending or, for example, a sequence of document reviews with each intervenient changing the file as he needs and then sends his version to the remaining receivers. This has several consequences, with some being as follows:

1. The fact that there's **no automatic merging** between versions. This process has to be done manually and, the bigger the file is, more painful it is to do this merging, because many details need to be taken in account so that the merge is done without losing any important detail;

2. **Controlling versions is very hard**, because there's no official versioning, so the changes' log between versions can be done in, at least, two ways - the author says what he has changed in the email that has the correspondent version attached or someone reviews the previous version and the newer one side by side to identify the differences. Both options, being purely manually executed have a significant amount of space to mistakes meaning that one detail might go unnoticed changing at the same time drastically the context.

3.1.4 To do list

Since a user starts using email, an unread email is a "shiny dot" in the user's inbox, which caughts the user's attention. That lead to the start of a phenomenon that is the user

letting an email as unread or, after reading it marking it as unread again, so that the "shiny dot" remains there so that the user can get reminded of something related to that email (collecting more information about the subject before answering or a task associated to that email). This has been studied in several papers, such as [WS96, KWS04, DB01]. This issue is targeted by to do applications and services which can be integrated with email services like Any.do [1], for example, but are not an integrating part of email clients by themselves, as stated on [Moo].

3.2 Email usage gaps

As it was stated on [DW05] and also as it was said previously on this document the use of the email environment has evolved a lot throughout time and that evolution started to show some gaps / deficiencies in this system, which haven't been properly documented or related between them. This next chapter aims to identify some of those gaps and, in some cases point to a possible solution. Nevertheless, some of these problems are going to be deeply studied next in the the present chapter, like said in the introductory chapter and better explained on chapter 4 as well.

3.2.1 Threads

Email threads are a set of email messages related between them by a reply-to sequence, where an email is always a response to a previous one of the same sequence, as defined in [KK03]. The first email is called the root of the thread and when an email is response to another, that one is called the parent of the reply. These threads can have multiple origins, for example a conversation between two people discussing an assignment or a team discussing the state of a project. In any case, threads are characterized for having many messages and, usually, a big number of participants. This leads to a very large set of messages distributed most of the times with not linear relation between each other. This has been a target many studies and attempts of solutions by numerous authors, like [Sam04, WS96, KK03, KWS04, KWS04, RGM+, VGD] for example. Nevertheless, in the most used email clients, thread visualization and interaction still lacks some serious improvements, as it's still similar to what it was a decade ago. This visualization includes the information from the sender and the receiver(s), its time stamp, the content and some citations from the parent(s). This leads to several problems, them being:

1. **Keep track of the participants** - It's hard to distinguish within so much messages when a new participant is added to the conversation or if a participant is temporarily

[1] http://www.any.do/

removed from the conversation. There are some situations where the "Reply" or "Reply to all" are misused and that can lead to some awkward situations because the email interface is almost plain when dealing with the users' participation in threads. Another example is, when the user wants to maintain a parallel conversation within the same thread to discuss a personal problem (picking the software development team example again, when a developer wants to ask a question about a new functionality to another developer without flooding the remaining elements of the team with the personal questions but that is, at the same time, about the same topic) and then get back to the general conversation.;

2. **Keep track of relevant informations** - As text in email messages is mostly plain - only has formatting -, it is hard to distinguish which is the important information to retain, even most if it's an email with a big content or if it has inline replies. So there should be a way to highlight these important informations, making sure that the receivers really noticed that a certain point really is important and not just hoping for that;

3. **Keep track of tasks that emerge** - This topic is closely related to the last one, being more specific, just because the assignment of a task is indeed an important information. Nevertheless, it's a more specific subject and, in that way, should be possible to declare tasks in an email message so that the remaining participant(s) could check that in a prioritized way and even have the possibility to reply to the email assuming the responsibility of a task, in a formal way;

4. **Keep track of relationships between messages** - There are two motives to reply to a message - the user either wants to answer to some specific message and his reply refers to the parent's content or the user just wants to continue contribute to the conversation. In any case, when a thread reaches a certain point where tracking difficulty increases exponentially, making the task of accompanying the evolution of the thread and its information hard;

5. **Keep track of files' versions** - As said in the previous section, file sharing is one of the new features that users associate to email, making use of attachments. This, besides some solid alternatives to share files (Like Dropbox [2] or Google Drive [3] are two of the most popular file sharing systems. Also, version control systems are an alternative like GitHub [4] for example), email is one of the user's top choices [DB01],

[2] www.dropbox.com
[3] drive.google.com
[4] github.com

contributing as well to this email overload, adding one more variable to the thread chaos.

There is already some literature and some investigation, as it was mentioned before, on this topic, and some ideas on how to improve this scenario. One of them, for example is a thread tree with a timeline, as it can be seen on figure 3.1.

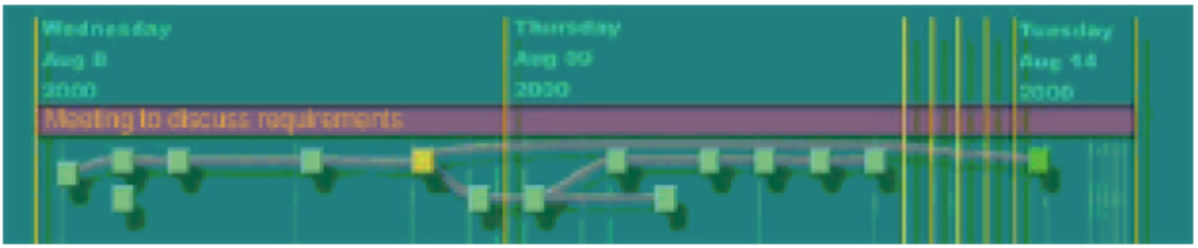


Figure 3.1: Email thread with a tree visualization proposed by [RGM+]

Another approach are the thread arcs, as observed on figure 3.2.

Figure 3.2: Email thread with arc visualization proposed by [KWS04]

3.2.2 Information separation and organization

Email's archives have been growing and growing as a consequence of the increasing of the daily number of email messages - as it can be seen on the comparison of a user study done in 1996 on [WS96] and another one done in 2006 on [FBGS06] - and the fact of the users tend to not delete emails frequently. With this, a need to get a fast and clean way to organize email and searching for information - in 2006 the average email box had over 28 000 messages - in a fast, clean and simple way.

Since the early times of email there's the possibility to create folders so that the user can distribute the messages through those and in that way get his email box more organized - with emails separated by the users' wishes. But, the use of folders has not been properly effective by the users because, as the volume of email messages rises, the user

either puts his efforts into maintaining himself an up-to-date inbox or organizing it, letting the messages accumulate over time. This phenomenon was called by [FBGS06] as *failed folders* - which are folders with less than 3 emails or that aren't used for a long period of time. On this study, one can conclude that, even though there are some patterns of inbox management and these patterns sometimes include occasional cleaning or organization, this is not a standard or common process. So, most of the inboxes are a frustrated attempt of organization because, as stated previously, the users have to put a lot of time on this organization and they prefer to lose this time getting up to date with the email flow.

So, inbox's organization is not optimized, which can lead to problems of productivity when looking for emails - most of the times filtering by substrings or sorting by dates/receivers/senders is not enough. Email clients shall give a better support for email organization after the user has dealt with it in a way that access to that information later in not a painful process and, at the same time, the user doesn't have the feel of working on an unorganized desk where the information is all over the place, and different types of information are mixed with each other.

3.2.3 Lack of context

On [DW05] it is said that there are some pieces of information that only have value within its context. This is one of the problems that most affects cognition because the user either makes an effort to understand what's the context of a certain message - by searching possibly related messages which, following the last topic is also hard - or remains without knowing what it refers.

This can happen in multiple scenarios like when a user receives an answer to a thread that he had already sent to the trash and has no background of the conversation or if a user receives an email regarding a parallel conversation in which he's involved but he is not up to date with it. Like it was mentioned before, this can happen in a huge variety of scenarios but today's email clients do not give enough contextual support (like suggesting similar conversations, or retrieving messages from a thread from the trash temporarily when a message from that thread is received so the user can have the whole picture, even though that information is discarded after again) to the user to avoid these situations.

3.2.4 Process automation

As said multiple times in the aforementioned references, time is always a problem when managing email. Time is never enough to do everything that one needs to do email related, being some of these actions are repetitive - like replying the same way to different emails,

send / receive reminders or categorize emails. This points to a strong need for process automation in the email environment.

There are some cases where this need for process automation is more obvious, like when a user is expecting an answer to an important email, when a user has a task associated to an email that he hasn't fulfilled yet or when a user has to categorize received/sent emails. There is already some literature on this topic - automating processes that are triggered by an email like [DLK06] or [DKFK05].

This automation can be on several forms, like an automatic reply (which already exists, although not on its full potential), automatic email classification by area or importance, for example (which also already exists as the study on [DKFK05] but is not spread or mature as it should and it's still retracting users). Also spontaneous messages (like said before, to remind a receiver that the sender is waiting for an answer to a previous email) are examples of this automatic processes implementation. Although there some steps towards this direction (from the identified gaps this is the one who has a more mature work on the market, but there still space to improve).

3.2.5 Social Email

In the email system the only way to retrieve information about a message or a thread today is either from the metadata or from the content. Nevertheless there is implicit information that's associated to an email that is in the user's head or in the user's email context, in a way that information can be gathered from the email's usage patterns, like for example, how urgent it is to receive an answer to a certain email or how delicate it is the relation with the client who's the email's receiver.

As explained on [Gui11], email is evolving to a more social state, where information shall be deducted from the user's actions and the focus on email interaction and environment should be more about the people rather than the messages. As the preferred information broadcast from social networks to their users (LinkedIn and Facebook are examples, with notifications being sent by email) this "social email" is enhanced because more information can be extracted from this and so the user experience in its email client environment can be improved taking advantage of this (besides the conclusions that can be taken from the email usage patterns) in such a way that these relations and this implicit information are explicitly displayed in the email client to help acquiring information.

3.2.6 Email archiving

This topic is related to several of the remaining email gaps here described, which is email archiving. Most of the users do not delete their emails, they just let them be in the

inbox (or distributed among folders) because eventually they may need them. These two activities (store the messages and eventually pick them) is called Archiving and retrieving respectively by [Sam04]. This is not a problem by itself, but as said before, it is a problem because it affects other areas, like the information searching and organization and the lack of context. This email archiving is not enough to be considered a Big Data problem (because per user, data doesn't reach the Terabyte mile and the information is not being gathered on real time, for example), but is indeed a problem of dealing with a big amount of data, and indexing it in such a way that relating information or searching it is done in a smooth way and not a long and repetitive process until the results are achieved.

3.3 Solving email gaps with Information Visualization

From all the problems mentioned in the previous section, almost all of them have visualization issues - in some cases the issue is that there's no visualization at all, like the lack of context problem. These problems are going to be the main focus of this dissertation, although if there's enough time, the problems that aren't directly related with information visualization are also going to be studied.

So the strategy, when solving these problems and relating them to Information Visualization, is going to be:

1. Understand what are the users' needs and the actions that are attached to these needs in the scope of a certain problem;
2. Mapping the dimension of the problem to one of the dimensions proposed on chapter 2.2;
3. Associate what are the actions also proposed on chapter 2.2 that can be applied to the previous point;
4. Develop prototypes that support the techniques and actions that were studied before and, having also in mind the principles of chapter 2.1 to enforce even more the application Information Visualization principles.

A strategy overview for each one of the problems that are going to be studied is going to be given on chapter 4. Of course that during the implementation process these strategies can suffer some changes because, one of the peculiarities of Information Visualization is that it's not a linear science and therefore some adjustments might have to be done to the solution when going from theory to practice.

Chapter 4

Problems and Approach

4.1 Problems addressed 27
4.1.1 Thread visualization 28
4.1.2 Individual message visualization and comparison 28
4.1.3 Thread participants analysis 28
4.2 Approach 28
4.3 Technologies 29
4.3.1 Visualization 29
4.3.2 Application 30

This chapter is going to present in detail three main topics regarding the practical execution of this dissertation, being them:

- **Problems addressed** - from the set of problems identified previously, here are going to be detailed which are the ones that effectively were studied and solved within the scope of this dissertation and a justification of why those were the chosen ones;
- **Approach** - Which was the followed approach to solve the problem here described and also some technical and logistical information about the scope of the project;
- **Technologies** - Which were the used technologies why they were chosen.

4.1 Problems addressed

Based on the set of problems described previously, as this dissertation's timeframe did not allow to take care of them all and also not all of them are related to Information

Visualization, a subset, presented next, was chosen according to the company's expectations and the author's opinion on what's more important to improve the email experience. Each problem will be shortly described - a more detailed description can be found in the previous chapter - , and an overview of the idea that was used to solve that problem - a complete description of the solution can be found in the next chapter.

4.1.1 Thread visualization

The first problem that was addressed is visualization of a thread, as described earlier. More specifically, the problems of understanding relations between messages, the display of the whole thread and exploration of its contents will be studied.

The solution consisted on implement three different visualizations over a thread namely a graph-like, a timeline and a chat.

4.1.2 Individual message visualization and comparison

This problem is related to the individual content of an email and it aims to take advantage of its text - both actual message and the inline content *baggage* that a message has. The main idea was to use inline content to recreate a thread, either a full recreation (for example if a message is forwarded to the user from another source, after a conversation was born and grown) or just filling the gaps (the user sent an email to the trash that was part of the visualized thread).

4.1.3 Thread participants analysis

Last problematic to be addressed is to analyze the relations between thread's participants and also their individual contribution to the thread. This is going to take in account the reply-to properties of each message, sender and receiver(s) - not CC or BCC but rather all the subjects that are in the *To* field.

4.2 Approach

As mentioned before, this dissertation was born within the scope of a new email client, developed by mailcube, which is still currently under development. So this work was not stand alone but rather was integrated into a bigger project which is, as previously stated, under development and the beta version is planned to be released soon.

The product wasn't released to the market in the useful timeframe of this dissertation, so, the prototypes developed and that are presented here were not fully integrated with it

and were not validated by the whole set of users and with the latest design work. With this setback the approach used was to develop more simplistic (regarding the *looks*) prototypes but more focused in the functionalities, maturing and validating them.

This resulted in the fact that the final outcome of this dissertation isn't yet ready to be integrated with the final version of the application but rather the concepts and the ideas are mature enough and validated by the users and only missing the visual tweak to make them suitable with the remaining application. As the figures presented in the next chapter state, the core design process was related to think what kind of colors use (more agressive or more neutral) or where to fit the prototypes (integrated with a standard reading view or apart from it, being an alternative to the standard), for example. This approach also allowed to surpass some technical difficulties due to the graphical customization of the prototypes that used OS X's native technology.

The prototypes were developed and when they were mature enough to be evaluated by the user, evaluation sessions were conducted - using usability tests. There was only time for a round of evaluations per prototype but it was gathered a lot of feedback - most of it validated the ideas proposed - and the improvements based on the users' suggestions were implemented to the final version and also were used to build the future work's design. The goal of these evaluations was to validate mostly the concept and interaction to know if the prototypes were in the right path. This was achieved with success.

The practical execution followed an agile approach, with iterations of two weeks, where the tasks were distributed among these iterations. Each one had associated problems to solve, within the scope of the previous problems' section. Tasks of development, evaluation and document writing were contemplated and distributed among the sprints.

4.3 Technologies

The technologies were restricted by the company, because the product already has a target platform. The target platform of the email client and in consequence the platform used to implement these prototypes is Mac OS X. As suggested on [Appb] it was used the XCode IDE, and the development environment - Objective-C and, on top of it, the Cocoa framework [Appa]- was based on the suggestions of the reference mentioned, with a few nuances which are going to be detailed in the next sections.

4.3.1 Visualization

As referred throughout this document, the main outcome of this dissertation are visualization prototypes which are integrated into an under-development Mac OS X application.

In this environment, there are some libraries and frameworks that vary in complexity and scope of functionalities that they provide, being the core technology OpenGL [Inc].

Nevertheless, its learning curve is big and complexity is high. This fact would lead to spending a big amount of time and effort getting into these technologies and learning how to use them properly, having to deal with performance and complex scene management and consequently spending less time working on the prototypes themselves.

An alternative was chosen, due to several facts that are going to be specified further, which was to implement these visualization prototypes in web technologies and then integrate them in the Mac OS X app via WebViews [Unk14] - OS X's native views that are capable of loading and displaying web pages. On top of this, the amount of available libraries (like D3 [1] or Raphaël[2]) also adds potential and interaction possibilities that can improve the user experience handling with these prototypes. This way, it is achieved the versatility of the graphic display and management, which together with the easiness and familiarity of the author with this technology and its performance (due to the asynchronous requests) is a solid alternative to the native technologies. Also, the customization that is achievable with style sheets allows these prototypes to look like native elements which does not harm the general style of the application.

Having stated the previous points, a choice was made that was to integrate web technologies (HTML, CSS and mostly Javascript and related libraries like the ones aforementioned) with the native environment in such a way that Javascript would act like the view in the model-view-controller pattern and the Mac platform would fill the remaining roles. With this, it's possible to achieve high degrees of customization that web technologies offer by default and, at the same time it's possible to integrate it with the native environment, offering a highly customizable user experience being also easy to implement, which is a big plus when the goal is to develop prototypes to validate concepts. This choice, lead to the fact that more time could be spent improving the prototypes and their details and not solving technical problems or seeking technical, achieving this way a much more mature set of prototype, which are validated and usable.

4.3.2 Application

As stated in the previous section, WebViews will act as the main component in the view in the model-view-controller pattern, which is the core architecture of the Mac OS X development environment, in some of the prototypes. The remaining elements would use the native environment, with Objective-C as the main language and Cocoa as the supporting framework. Therefore this process was done by using native and core

[1] http://d3js.org/
[2] http://raphaeljs.com/

technologies of the Mac OS X environment. It was not used Swift because the components
2 that were already implemented in the application were using Objective-C and, by coherence and integration easiness, these prototypes followed the same path.

Chapter 5

Prototypes

5.1 Thread visualization 35
5.1.1 Integration within the email client 35
5.1.2 Tree graph 35
Concept 36
Principles and relation to Information Visualization 37
Implementation 40
Main features 41
5.1.3 Timeline 41
Concept 41
Principles and relation to Information Visualization 42
Implementation 43
Main features 45
5.1.4 Chat 45
Concept 45
Principles and relation to Information Visualization 46
Implementation 47
Main features 47
5.2 Email comparison and visualization 47
5.2.1 Identifying inline messages 48
5.2.2 Email clients/service identification 50
5.2.3 Principles and relation to Information Visualization 55
5.3 Sociogram 55
5.3.1 Concept 56
5.3.2 Principles and relation to Information Visualization 56
5.3.3 Implementation 59
5.3.4 Main features 60

As it was mentioned in the previous chapter, the concrete outcome of this dissertation will be the prototypes that solve the addressed problems. This chapter aims to document the developed prototypes along with their technical and logical details.

During this dissertation five prototypes were implemented five prototypes - two that act as a complement to a standard reading view of an email thread, providing additional information about its content, one that acts as an alternative to the same view, displaying information in a slightly different way from what's standard in email clients in a specific type of conversation, one that recreates the messages of a thread (either all or the missing ones) from an email inline content and finally the last (but not least) prototype that displays information retrieved from the thread - relations between intervenients of the thread.

A transversal fact to all the prototypes is their main color (either most predominant - timeline - or the color used to highlight certain items - tree graph, for example). The chosen color was a tone of orange due, mainly, to two facts which are:

- It's a color that's easily perceived by the human brain's preattentive vision. This means that it's a color that catches user's attention and highlights itself from the remaining (more neutral or more commonly used in the context) colors;
- From the set of colors that fill the previous requisite this orange (#FF9A00) is the one that suits better with the application - is similar to the application's main color and does not have a specific or associated meaning like red is normally associated to a negative answer, for example.

Another fact common to the majority of the prototypes is the use of the circle to represent elements (a message in the tree graph or an individual in the sociogram, for example). As Birkhoff [Bir33] says, the circle is the perfect shape as it has the biggest number of symmetry axis and number of admissible rotation angles among all the shapes.

Also, these prototypes take advantage of both preattentive vision - by using color and size to highlight certain element within a set of similar elements, which is more detailed into each prototype, when applicable - and Gestalt theory - exploring intensely the relation between field, structure and shape to highlight the elements so that the user can focus on them and the information they intend to transmit and not on the *environment* where they're displayed - presented in chapter 2.

5.1 Thread visualization

The following section's goals are to describe the implementation and the logic behind the prototypes developed to solve the thread visualization problem described in s 3 and 4. A relevant note in these prototypes, except when stated otherwise, is that messages within each thread are organized in a descending order, temporarily speaking. This uses the approach "newer first" used in Apple Mail email client, for example.

5.1.1 Integration within the email client

As previously stated, the goal of developing these prototypes is to validate the ideas here proposed in order to integrate them in a disruptive email client that embraces this new features and that changes the paradigm of email visualization with them.

Regarding the two first visualization prototypes - a tree graph and a timeline -, the integration of the prototypes was to try not just simply pasting new features into the email client. It was done by blending the standard visualization (boxes placed sequentially with email information inside each one) with the new visualizations in such a way that the user does not have to choose between one and the other(s). Instead these new visualizations shall be a (not intrusive) helper for the user to explore and retrieve information. With this the user experience is enhanced as he has more tools to explore information and to retrieve it easily but they're integrated into the user's normal workflow and workspace, not being a strange body. The author believes that previous attempts of such visualization proposals like [Sam04] did not succeed because they were introduced as alternatives to the standard visualization and not as an auxiliary tool or extension of the email reading view, that's fully integrated and that acts as a helper.

This integration required a detailed application design process so that, even though there are new features, they're complementary and not an alternative, that was just stitched to the email reading area. The way these prototypes were integrated and how they fit and interact with the application will be described next.

5.1.2 Tree graph

The first prototype implemented to visualize email in a different way was a message graph (also denominated message graph) visualization. Next are going to be described the concept, principles and arguments behind this feature as well as the main technical features implemented.

Concept

A thread of emails (or conversation) is a sequence of messages which are connected via a reply-to relation, meaning that an email is the root of the sequence and the following ones are replies either to the root or to any other email that meanwhile joined the thread as a reply to another one. This means that a message always has a parent message, to which it was a reply (except from the first message of the thread, named the root) and can have an infinite number of replies, creating a tree structure.

The prototype developed intended to solve the problem of understanding the relation between messages in such a way that the user can understand their flow, understand if there are parallel conversations, understand which are the most relevant emails - the more replies an email has the more relevant it might be in the scope of thread - or any other conclusions that the user might infer from the graphical representation. Nevertheless, this tree representation was just conceptual, because the implementation was done in a slightly different way from the conventional tree representation.

The conventional representation of a tree structure is as figure 5.1 indicates. Here the approach followed was slightly different and it's explained next.

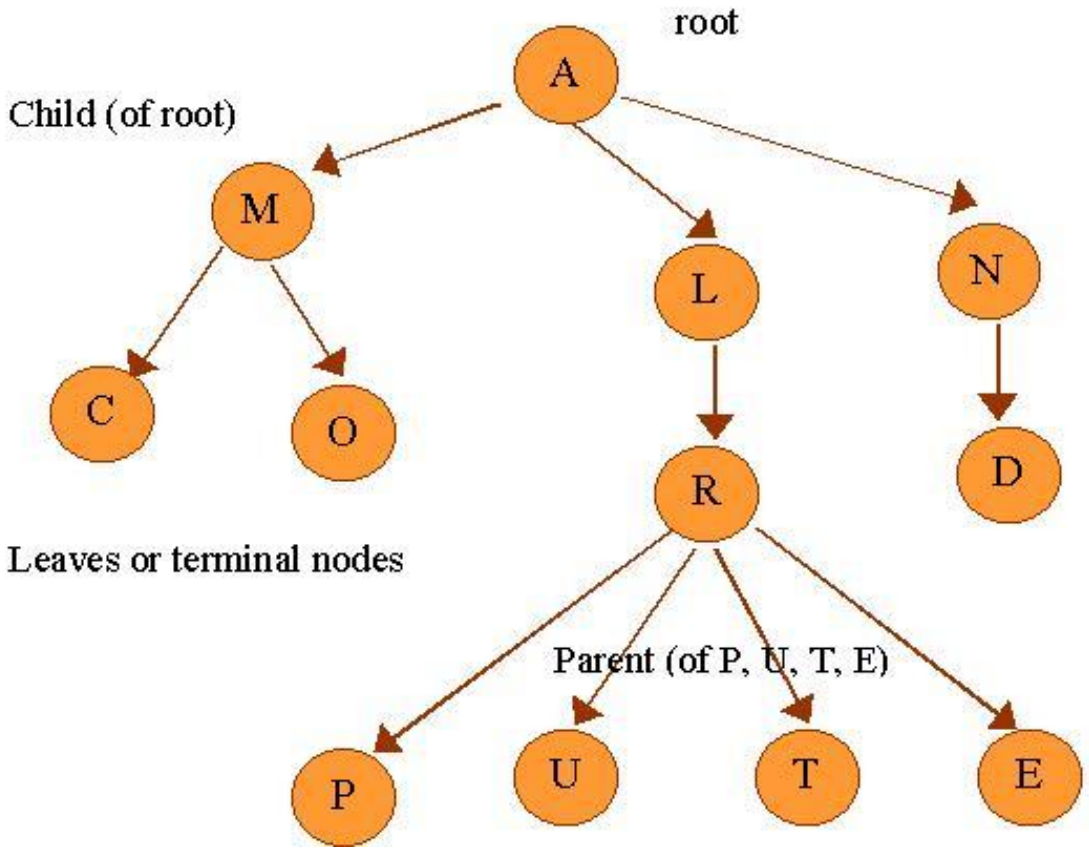


Figure 5.1: Conventional graphical representation of a tree structure. [1]

Principles and relation to Information Visualization

Before explaining the context, evolution and logic behind the several phases of the prototype, it's going to be explained the terminology and symbology used. It is as follows:

- **Node** - A node in this context represents a single message that belongs to the thread. A node can be highlighted, normal or de-highlighted depending on the status of the selection or state of the thread;

- **Connection** - A connection between two nodes represents a relation between them. In this case is a reply-to relation meaning that one node (message) is a reply to another one (parent message). The connection has one of two types: either connection to the parent (message to which the selected message is a reply) - dashed line - or a connection to the children (messages that were a reply to the selected message) - full line;

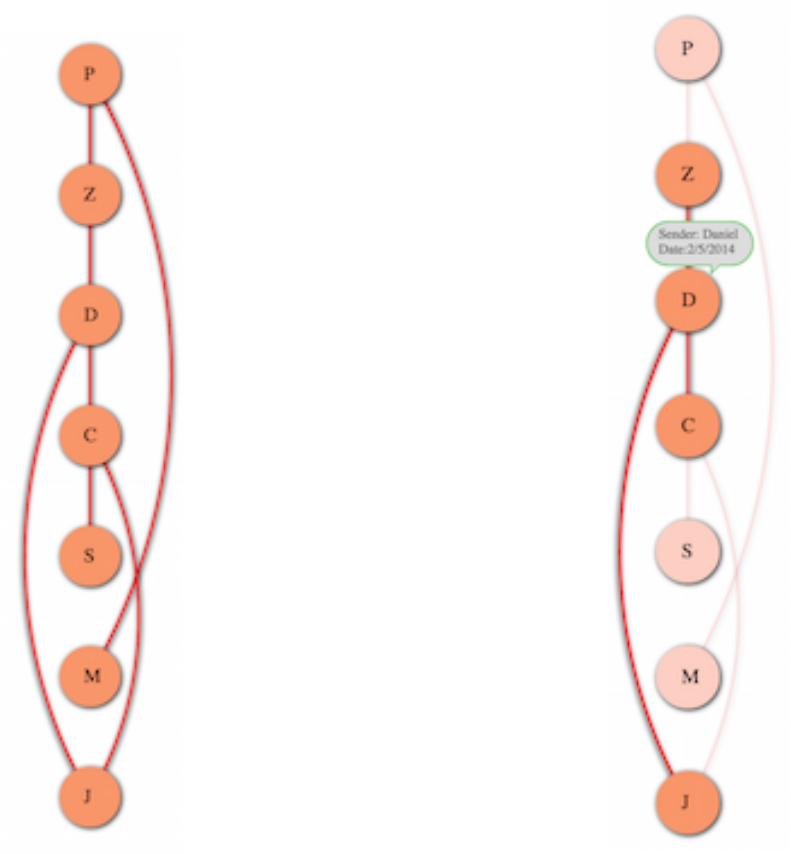
As stated before, the idea behind this implementation is to help the user perceive the relation between messages. Nevertheless, and once again as stated before, this visualization was complementary to the standard one, meaning that the information exploration process includes this standard reading view of an email thread.

This idea had a few phases and, in the first one, it was supposed to be an alternative view the standard one. A sketch of this idea can be found on figure 5.2.a) It is a vertical tree where the oldest node (root message) is the one on the top and the connections - which are all always visible - are represented through straight lines - when the parent of a reply is the one right before itself - and curved lines - when the parent of a reply isn't the one right before itself. For a better understanding of the relation between messages it had a highlighting mechanism of the direct relations of a message (its parent and its replies). this highlighting mechanism can be seen on figure 5.2.b).

After this iteration, and realising internally that this approach would fall in the same identified mistake in similar solution proposals - provide an alternative instead of a complement and information overload in the sensde that all the connections were visible at all time -, a few improvements were done, namely the integration with the reading area and the visualization itself.

The major differences were that the visualization was next to the reading area and side by side with the standard visualization - being a node associated to the "box" with the email content and metadata - and the edges between nodes (that weren't always all visible but rather were displayed depending on the selected node, showing direct relationships) were changed in such a way that the looks of the connections to the parent and the replies

[1] Source: http://sourceware.org/psim/manual/tree.gif. Last accessed 10-March-2015.



(a) First concept without any selection

(b) First concept with node "D" (and direct relations) highlighted

Figure 5.2: First tree representation concept

are distinguishable and more easily perceived which node is the parent and which are the replies.

This way, and as the node is always associated (logically and visually) to the email "box", it acts as a complementary information, not having to choose between visualizations. Also, as the information is automatically triggered with all of the actions and selections that are applied to other elements within the reading view so it would be natural for that element to be there and not being a strange one.

The final result was as it can be seen in figure 5.3

When an item is selected, the relevant items get a visual update, where the selected item is shown with a different color to be highlighted from the remaining ones - using the orange color mentioned previously - and the connections between the relevant nodes (parent and replies, when applicable) are drawn. Figure 5.4 shows how the tree graph view looks when an item is selected.

This prototype follows most of the principles of Information Visualization presented previously, as follows the explanation next:

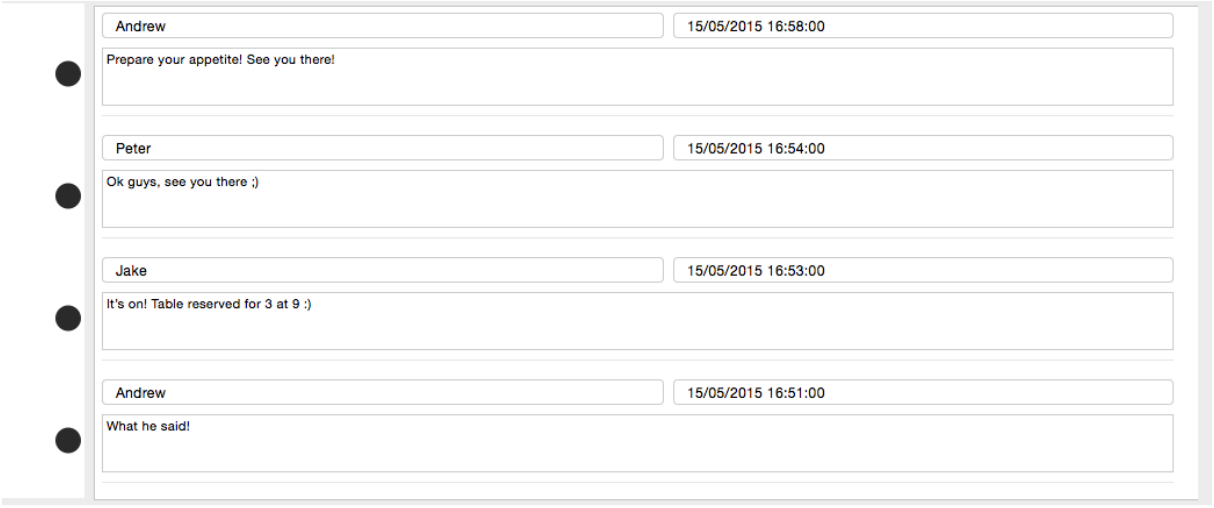


Figure 5.3: Final result of tree view prototype, at the left, and messages content, at the right.

1. **Overview** - The first impression that the user has with this prototype is that there's a node for each message. This allows to get an overview of the number of nodes. Further exploration will give an overview on the reply pattern;

2. **Zoom** - Zoom in not applicable in this context;

3. **Filter** - Filter is not applicable in this context because the elements that are shown are related to the messages of the thread and, as the thread is not "filterable", the prototype isn't "filterable" either;

4. **Details-on-demand** - When the user clicks a node, its details (connections) are shown and the message is highlighted in the message view. Also, when the user hovers a node with the mouse, a tooltip is shown with some additional information (send date and sender) of the message;

5. **Relate** - When an item is clicked its direct relations are shown, allowing that way the user to relate a message with others. Also, this selection is replicated in other views (reading view with messages list and timeline), by offering a way to relate items not only regarding the prototype information but also with information/items of other areas of the "bigger picture";

6. **History** - Being the possible actions to select (click on an item selects it, click on the selected item/empty space de-selects it and click on another item updates selection) or scroll, keep track of history is pretty straightforward;

7. **Extract** - Not applicable within this context.

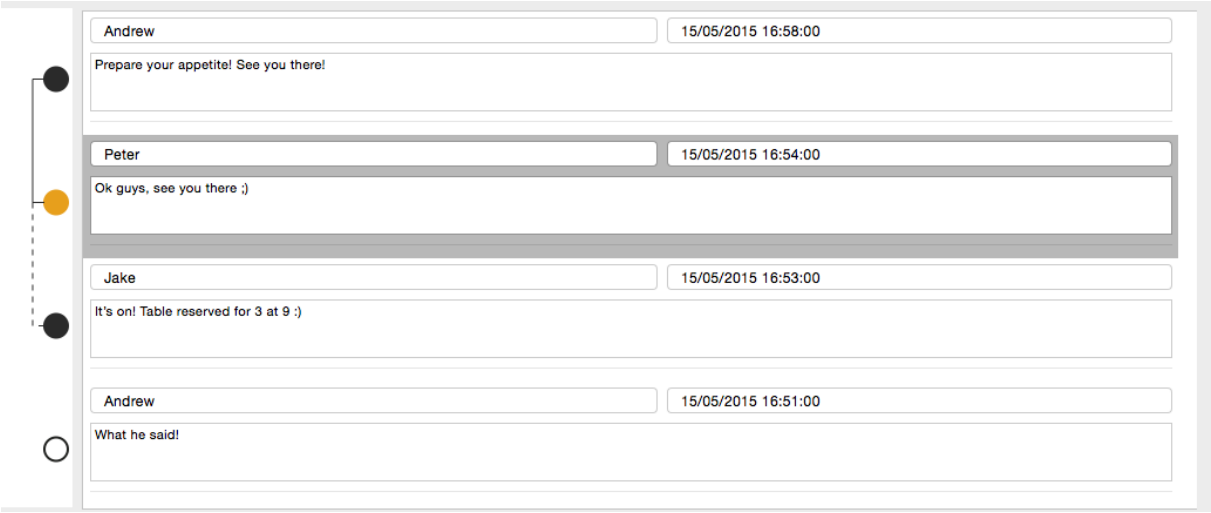


Figure 5.4: Final result of tree view prototype, at the left, and messages content, at the right, with a message selected.

Regarding the data type, also referred in 2.2, it fits in the category *Tree*. It was previously explained, a thread is organized in a tree-like structure, where the first node (root of the thread) can have an infinite number of replies. Each one of these replies can also have an infinite number of replies. These relationships are defined via the reply-to field of an email message.

Implementation

This feature was implemented using webviews (reasons were mentioned in chapter 4). To implement it, a set of alternatives were analyzed and the best one was chosen in order to maximize performance. The alternatives analyzed were:

- **D3.js** [2] - The first option was to use the Javascript library D3.js. It is recognized as one of the best and most complete Javascrip visual libraries that are available today. Nevertheless, its loading time (aprox 3-4 seconds) - due to the Force-directed layout [FR91]- and high degree of complexity were too much for this purpose. Therefore, this library was discarded;

- **Raphaël** [3] - Raphaël's library is mostly a wrapper over SVG elements. It was discarded because, by managing directly SVG elements, the application could have more control over what's going on and without losing performance with unnecessary library actions;

[2] http://d3js.org/
[3] http://raphaeljs.com/

- **Pure SVG drawing** [4] - This option requires to handle all the drawing and interaction by hand. Nevertheless, as they're not very complex, this option was chosen due to performance and it allowed more control over what's and how's draw. This option was the one chosen to implement the prototype.

Main features

The main technical features of this prototype are:

- **Automatic highlighting** - When a node's selected, automatically the relevant nodes (parent and replies) are highlighted. Regarding the non-relevant nodes (the ones which don't have a direct relation with the selected node) are on the other hand given less relevance. Also the selected email is also highlighted on other views - email list and timeline - in order for the user to have a global overview and not local.

- **Distinguish between parent and reply(ies)** - The direct relationships of a message are shown by drawing vertical connections from the node to the parent and replies (if applicable). Nevertheless these relations -parent/replies- are visually distinct in order to facilitate the understanding which nodes are what;

- **Synchronized with email list scroll** - As the prototype is connected to the reading view but they're different elements, scroll between them was synchronized. This allows the user to navigate or to perform selections in any of them, that the other goes along smoothly and transparently for the user.

5.1.3 Timeline

The second prototype implemented to help retrieve information from an email was a timeline where the emails are distributed chronologically according to their send date. Next are going to be presented the concept and fundamental principles on this feature.

Concept

The fundamental concept of this prototype is that email messages within a thread always have a chronological order which can be sorted by different criteria - such as send date or reading date. In this case the send date of a message will be used to do this sorting. Chronological order can help the user locate himself in the thread, and locate messages in time in a more efficient way than just scroll sequentially until the date that the user thought the message was sent for several reasons. One of them is because it shows message

[4] http://www.w3schools.com/svg/

distribution along time supporting, for example, some notions like "a while ago" or "in the last few days", to improve search for example. Another one is, for example, because helps the user location the periods that had effectively message exchange within the thread.

Principles and relation to Information Visualization

Firstly, it will be explained the elements that compose the timeline here described:

- **Node** - A node in this context represents an email message and it's identified by a circle or a rectangle with rounded corners if it's a cluster (events close to each other that are grouped when zoom does not allow to view them all);
- **Timeline** - Timeline is the area where the nodes are displayed and it's composed by a set of labels at the bottom with timestamps to identify the time range, vertical lines to help separate these timestamps and a thicker line to identify which is the current time (if current time belongs to the range shown in the timeline) .

An example of these elements can be seen figure 5.5

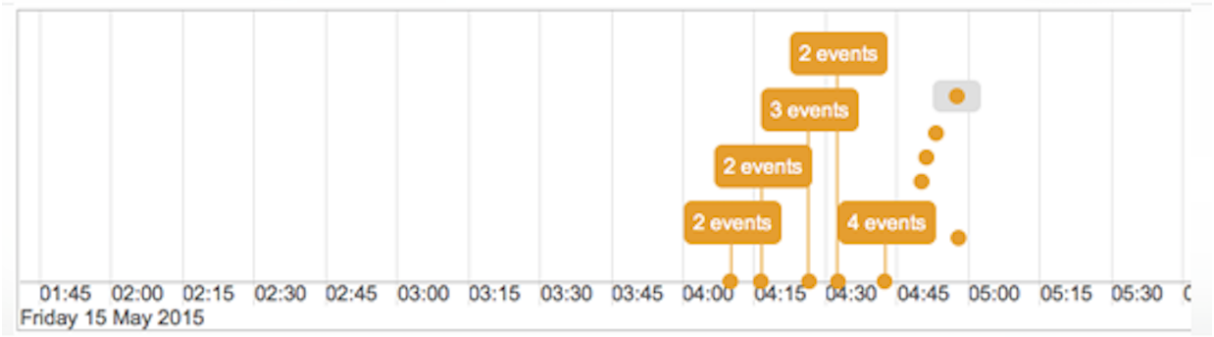


Figure 5.5: Timeline prototype example

This prototype follows most of the principles of Information Visualization presented previously, as follows the explanation next:

1. **Overview** - All the items are disposed inside the timeline, being the timespan defined to be between the date of the first and the last email of a conversation; When there's a big amount of items within the same period, a cluster (group) is created, indicating how many items are in that period so that the user can easily perceive the amount of overlapped items;
2. **Zoom** - Within this context, zoom is literally the action to explore the information given by the overview phase, where the user zooms in in a certain area to increase the granularity of time (going from a monthly view to a weekly view, for example),

unclustering at the same time if there's a cluster on the zoomed zone or the user can zoom out to get once again the bigger picture of a certain area;

3. **Filter** - Filter is not literally applicable within this context but rather is blended with the remaining options; From the start is filtered with the start and end date of the conversational thread and, each time the user zooms in/out the start and end dates of the visible interval are updated, filtering that way the data that the user's viewing;

4. **Details-on-demand** - When a user clicks a message, the reading view scrolls to the selected item and highlights it (triggering also the scrolling and highlighting), having the user that way access to the details of the conversation;

5. **Relate** - The timeline prototype addresses this question by displaying messages that are close in time close visually, being the user able to relate them by their time attribute;

6. **History** - Being the possible actions to zoom (zoom in reverses a zoom out action and vice-versa) and select (an item can be selected and then deselected with a single click, being only at most an item selected at each time), which are reversible and easily tracked by the user;

7. **Extract** - Not applicable within this context.

Regarding the data type, also referred in chapter 2.2, it fits in the category *Temporal*. As the relevant content of each element of the prototype is its timestamp, in order to organize information in a timely manner and help the user locate himself, regarding the time, inside the thread.

Having stated the previous matches between Information Visualization principles and how they fit in this prototype, next some details on the timeline itself are going to be presented.

This timeline has 2 dimensions of exploration [ABYG07], a horizontal one and a vertical one.

The horizontal dimension of exploration intends to give an overview of the global time interval and explore this time interval. The vertical dimension intends to organize items which are close in time and allow the exploration of a single point in time.

Implementation

As it was mentioned before, some of the prototypes were implemented by mixing web technologies with the native technologies of Mac OS X environment, which is the case

of the timeline. By displaying it in a webview inside the application, it was possible to implement it using web technologies. Nevertheless this prototype wasn't developed from scratch and it was based on a library - **Chap Links Timeline** [5]. Several libraries were analysed before making a decision and the libraries analysed and why they weren't chosen is presented next:

- **Timeglider** [6] - Besides being paid (student version was available but within the context of this dissertation wasn't enough), it was very complex visually for the scope of this prototype;
- **Timeline JS** [7] - The main reason for not choosing this library was its way of displaying events as it needed too much information to display when the goal is not to have such a complex and powerful visualization of a singe item/event. Other negative point was the navigation, which wasn't possible using Mac OS X touchpad, for example;
- **Fancy Timeline** [8] - The navigation wasn't suited for the scope of the prototype as well as the display of the items and information's architecture that was too complex and detailed;
- **Wellcomelibrary Timeline** [9] - This library is, in all terms, way more complex than the needs of the prototype so it was discarded;
- **Chronoline.js** [10] - Chronoline was a good alternative but a little lack of customization power on the items and documentation blew the chances for this prototype;
- **vis.js** [11] - This library is the updated version of the one chosen. Nevertheless it still had non-implemented core features - like clustering items - at the time the decision had to be made so this version could not yet be used;

After having considered all of the reasons mentioned above Chap Links Timeline was chosen because its ideology and core is simplistic enough to not overload the user with information but, at the same time, already has a structure that allows a high degree of customization, having also a very complete documentation. It has some functionalities that are very important to the scope and that integrate Information Visualization principles, like clustering, which played also an important role in this decision.

[5] https://almende.github.io/chap-links-library/timeline.html
[6] http://timeglider.com/
[7] http://timeline.knightlab.com/
[8] http://juanma-aguero.github.io/fancy-timeline/
[9] https://github.com/wellcomelibrary/timeline
[10] http://stoicloofah.github.io/chronoline.js/
[11] http://visjs.org/

Main features

The main technical features of this prototype are:

- **Item disposition** - Items are disposed horizontally by their send date, incorporating this way the horizontal exploration process aforementioned;
- **Clustering** - When items are close in time and zoom and granularity aren't high enough to be possible to distinguish them clearly, items are grouped in clusters, supporting this way the vertical exploration process aforementioned;
- **Navigation** - Navigation in the timeline is done using Mac OS X native controls (zoom, click), giving the user a friendlier navigation environment;
- **Item highlighting** - It's possible to highlight an item in the timeline that triggers a reaction in the reading view (with the item being highlighted there as well) and in the tree graph view mentioned previously, highlighting the relationships of the selected message.

5.1.4 Chat

The last prototype that allows different visualization over a thread is a chat view, similar to Facebook Messenger [12] or Skype [13] , that is applied to conversations that have typically short messages and with a little timespan between them.

Concept

As referred previously there are some times that an email conversation is created almost synchronously with short messages in fast pace creating a conversation similar to a chat. This way, the idea here proposed inteds to specify a metric to identify these conversations as potential chats and display them this way.

The metric used here was to check if the average length of the email's content (ideally excluding signatures and other non-essential elements) is equal or less to 140 characters - this value was chosen because it's becoming a standard size to small messages given the massification of the Twitter [14] application, where maximum message size is 140 characters. This allows users to use a metric that's increasing in popularity and it's getting into their daily life more and more. If this requirement is fulfilled, the application should suggest the user to view this thread as a chat, not being intrusive or mandatory. The previous

[12] messenger.com
[13] http://www.skype.com/en/
[14] twitter.com

prototypes were intended to be a complement to the standard email reading visualization but, on the other hand, this approach is an alternative in the sense that displays the content of each message (same information as before) in a different manner. The user should always be allowed to change between visualizations being the application the less intrusive as possible, making only suggestions (the most accurate as possible as well).

The approach here used is very similar to other well-known chat applications like the ones mentioned before, for example, but that's applied to an area that until today hasn't been thought yet. This allows users to focus only the content of fast and short messages instead of being overload with unnecessary information (CC, BCC, full subject, etc) and also identifying pretty easily own messages from messages sent by other participants in the thread.

Principles and relation to Information Visualization

Firstly, it will be explained the elements that compose each line in the chat view:

- **Identifier of each intervenient** -Each intervenient in the thread will have an identifier. It a photo's available, then it will be shown. If not available, the first two letters of the contact's name will be shown. This allows for a quicker identification of the intervenients, as well as it's easy to distinguish them from eachother instead of just reading their email address. These identifiers are displayed like in most chats - identifier of other intervenients placed at the left side of the reading view and identifier of messages from self are placed at the right side of the reading view;

- **Timestamp** - Each message has a timestamp associated to it. Nevertheless this timestamp does not have a standard format within this prototype but rather varies with the distance in time since the message was sent until the present date. There are three different formats: messages from the current day - where it's only displayed the hours and minutes from when the message was sent, in the format **hh:mm**-, messages from the day before - where it's displayed in the form **Yesterday @ hh:mm** - and messages previous to those ones - where the dates are displayed in a more complete format: **dd/MM/yyy hh:mm**. This allows for a much easier relative time location of each message within the thread;

- **Content** - The content itself of the email message in a box properly identified. Size should adapt to text size;

This prototype does not follow Information Visualization's principles as strictly or in such an obvious way as the ones presented before and that are going to be presented ahead. This prototype does not have a set of actions attached to it that allows top to bottom

data exploration or relate items between themselves. This prototype takes advantage from preattentive vision to highlight and distinguish items based on certain attributes to allow a better distinction of messages bad on these same attributes.

An example of this perception is distinguish messages sent from self from messages sent from other intervenients due to the placement of the sender's identifier - either the picture or a label with the first two letters of his/her name. Other example is the distinction of the time a message was sent due to the different formats - and consequent length and information shown - in the dates of every message, is easily distinguishable that there are some distinct periods in time (today, yesterday, before yesterday).

Regarding the data type, this prototype used data that fits in the category of *2-dimensional* given the fact that the relevant attributes to analyze in each item of the chat are its sender and its send date, given the fact that the content is equally important regarding the sender or the send date.

Implementation

This prototype is the only one that was implemented using exclusively native technology (no webviews). It consists in a list of messages organized in ascending way , i.e. older at top, newer at bottom as this approach is more familiar to the users due to the fact that's how the other relevant chat applications display information.

Main features

The main technical features of this prototype are:

- **Auto-layout** -Automatically separates and identifies messages sent from self and from other users;
- **Easy sending** - At the bottom of the list with messages, there's an input area to write a response and send it. This allows for a faster response mechanism to the users;
- **Recognizing a chat** - Application will automatically recognize a chat-like thread and suggest this visualization to the user being the less intrusive as possible.

5.2 Email comparison and visualization

This feature aims to solve two different problems in email visualization, namely the extraction of useful information (although not in its full scope because it's a very large field of studies that involves text mining and visualization techniques) and lack of context.

Regarding the problem of information extraction from a message and comparison between messages the sub-problem here addressed is how to take advantage of the inline quoted text - hierarchical chain of messages with the (either full or partial) history of a certain thread - that comes in an email body. What happens today is that that quoted text is in fact content of an email but is completely disregarded. Nevertheless, in this quoted text (when it isn't erased by a certain user) there is the whole history of an email thread. That information might have a lot of value, to perceive the whole history of messages from a thread when the user is new to the conversation (either it was added in the context of the thread or the message with the history of messages has been forwarded to him). Despite this value, the visualization of this inline older answers is very poor. It is done only by indenting messages per levels and adding a type of header to the message, identifying the date it was sent and, most of the times, the sender and/or the receiver. An example of this visualization can be seen on figure 6.7.

The approached here used and that lead to the solution that's going to be described next is to extract these old messages from the text in such a way that the thread can be virtually reconstructed (even though a user only received a single message, it has the previous replies inline) graphically. This way, and integrating this with the thread visualizations that were proposed and described before and also that'll be presented next, the user can perceive much easily the history of the single message it received and the context it is inserted. Another case of the usefulness of this feature is when a user deletes a portion of a thread and then another message is received and that information can be temporarily restored.

Next, it's going to be described how these messages were able to be extracted and the prototype was built.

5.2.1 Identifying inline messages

When a user replies to an email message, it automatically adds the previous messages (and its inline contents, recursively) to its own inline content. Usually, each email client (either online or application) has a ***see more*** (or similar) button, to view the inline history, as it can be seen on figures 5.6 and 5.7. Each time that an email is added to the inline content, it is also added a header - usually a single line - to that message identifying the timestamp of that email and sometimes its sender and / or receiver. This header varies along the different email clients and services. The process used to identify each message, was as follows:

- Break inline content in lines, so that each one can be evaluated individually;

- Parse each line and check if a certain line matches the pattern used by any of the email clients/services inline message header;
- If any of the header's regular expressions matched a certain line, it is known that next is an email message content which inside can have more emails;
- After a line is matched, the next lines are assumed to be content of email until a new line is matched with an expression or the parser runs out of lines;
- Depending on which format a header is, it is possible to extract from it its send date and sender. This is saved in a data structure along the message content.

Figure 5.6: Example of the see more button. Gmail web client.

To identify which was the header used by the email services / clients, it was conducted an empirical study, which consisted in, first, identifying the most used email clients and services and then, by replying to emails from each of them and then observe which was it inline message header. The thread generated by this empirical test can be checked on M. Having this, regular expressions were created for each of the headers (because all of them were different among themselves) to match and validate the expressions. After that regular expressions for the date used in each header and a more general expression to validate and extract the email from a header, when it's present.

Figure 5.7: Example of the see more expanded button, with 1 message in the inline content. Gmail web client.

Having regular expressions to match the header itself, the date and email within it, everything is set to extract the meta-data from the inline email and its content. By parsing all of the inlined text, it's possible to extract all of the emails present.

5.2.2 Email clients/service identification

The most used email clients and services in 2014 and nowadays were enumerated in some articles like [Aut, Lit] and those were the email clients and services used in this empirical study. The regular expressions that validate each email client/service inline content header is as follows:

- **Yahoo**

```
On\s[A-Z][a-z]{5,8},\s[A-Z][a-z]{2,8}\s[0-9]{1,2},\s.{11,13},.*<.*@.*>\swrote:
```

- **Yahoo (nameless)**

```
On\s[A-Z][a-z]{5,8},\s[A-Z][a-z]{2,8}\s[0-9]{1,2},\s.{11,13},.*@.*\swrote:
```

- **Webmail**

```
Em\s[0-9]{2}.[0-9]{2}.[0-9]{4}.[0-9]{2}:[0-9]{2},.*@.*:
```

- **Outlook**

```
On\s[A-Z][a-z]{1,2},\s[A-Z][a-z]{1,2}\s[0-9]{1,2},\s[0-9]{4}\sat\s
    .*,\s.*<.*@.*>\swrote:
```

- **Outlook (nameless)**

```
On\s[A-Z][a-z]{1,2},\s[A-Z][a-z]{1,2}\s[0-9]{1,2},\s[0-9]{4}\sat\s
    .*,\s.*@.*\swrote:
```

- **Mailbox**

```
On\s[A-Z][a-z]{1,2},\s[A-Z][a-z]{2}\s[0-9]{2},\s[0-9]{4}\sat\s
    [0-9]{2}:[0-9]{2}\s[A-P]M,.*<.*@.*>\swrote:
```

- **Mailbox (nameless)**

```
On\s[A-Z][a-z]{1,2},\s[A-Z][a-z]{2}\s[0-9]{2},\s[0-9]{4}\sat\s
    [0-9]{2}:[0-9]{2}\s[A-P]M,.*@.*\swrote:
```

- **Apple mail**

```
On\s[0-9]{1,2}\s[A-Z][a-z]{1,2}\s[0-9]{4},\sat\s
    [0-9]{2}:[0-9]{2},.*<.*@.*>\swrote:
```

- **Apple mail (nameless)**

```
On\s[0-9]{1,2}\s[A-Z][a-z]{1,2}\s[0-9]{4},\sat\s
    [0-9]{2}:[0-9]{2},.*@.*\swrote:
```

- **Gmail online**

```
[0-9]{4}-[0-9]{2}-[0-9]{2}\s[0-9]{2}:[0-9]{2}\sGMT(\+|\-)
    [0-9]{2}:[0-9]{2}\s.*<.*@.*>:
```

- **Gmail online (nameless)**

```
[0-9]{4}-[0-9]{2}-[0-9]{2}\s[0-9]{2}:[0-9]{2}\sGMT(\+|\-)
    [0-9]{2}:[0-9]{2}\s.*@.*:
```

- **Thunderbird**

```
On\s[0-9]{2}\/[0-9]{2}\/[0-9]{2}\s[0-9]{2}:[0-9]{2}, .*\swrote:
```

- **iCloud**

```
On [A-Z][a-z]{1,2} [0-9]{1,2}, [0-9]{4}, at [0-9]{2}:[0-9]{2} (AM|
    PM),.*<.*@.*> wrote:
```

- **iCloud (nameless)**

```
On [A-Z][a-z]{1,2} [0-9]{1,2}, [0-9]{4}, at [0-9]{2}:[0-9]{2} (AM|
    PM), .*@.* wrote:
```

- **AOL**

```
-+Original Message-+\nFrom:.*<.*@.*>\nTo:.*<.*@.*>\nSent: [A-Z][a-
    z]{1,2}, [A-Z][a-z]{1,2} [0-9][0-9], [0-9]{4}
    [0-9]{1,2}:[0-9]{1,2} (am|pm)\nSubject:.*
```

- **AOL (nameless)**

```
-+Original Message-+\nFrom:.*@.*\nTo:.*@.*\nSent: [A-Z][a-z]{1,2},
     [A-Z][a-z]{1,2} [0-9][0-9], [0-9]{4} [0-9]{1,2}:[0-9]{1,2} (am
    |pm)\nSubject:.*
```

- **GMX**

```
Sent: [A-Z][a-z]{5,8}, [A-Z][a-z]{2,8} [0-9][0-9], [0-9]{4} at
    [0-9]{1,2}:[0-9]{1,2} (AM|PM)\nFrom:.*<.*@.*>\nTo:.*.*@.*\
    nSubject:.*
```

- **GMX (nameless)**

```
Sent: [A-Z][a-z]{5,8}, [A-Z][a-z]{2,8} [0-9][0-9], [0-9]{4} at
    [0-9]{1,2}:[0-9]{1,2} (AM|PM)\nFrom:.*@.*\nTo:.*@.*\nSubject:.*
```

- **Inbox**

```
On [A-Z][a-z]{1,2}, [A-Z][a-z]{1,2} [0-9]{1,2}, [0-9]{4} at
    [0-9]{1,2}:[0-9]{1,2} (AM|PM) .*<.*@.*> wrote:
```

- **Inbox (nameless)**

```
On [A-Z][a-z]{1,2}, [A-Z][a-z]{1,2} [0-9]{1,2}, [0-9]{4} at
    [0-9]{1,2}:[0-9]{1,2} (AM|PM) .*@.* wrote:
```

Having all of these regular expressions, to validate the date it's just needed to extract the date part from the regular expression of each email system/service. To validate and extract the email from the header, the following regular expression is used:

```
(?:[a-z0-9!#\$\%&'*+/=?^_`{|}~-]+(?:\.[a-z0-9!#\$\%&'*+/=?^_`{|}~-]+)*|\"
    (?:[\\x01-\\x08\\x0b\\x0c\\x0e-\\x1f\\x21\\x23-\\x5b\\x5d-\\x7f]|\\[\\
    x01-\\x09\\x0b\\x0c\\x0e-\\x7f])*\")@((?:(?:[a-z0-9](?:[a-z0-9-]*[a-z0
    -9])?\.)+[a-z0-9](?:[a-z0-9-]*[a-z0-9])
    ?|\[(?:(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.)
    {3}(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?|[a-z0-9-]*[a-z0-9]:(?:[\\x01
    -\\x08\\x0b\\x0c\\x0e-\\x1f\\x21-\\x5a\\x53-\\x7f]|\\[\\x01-\\x09\\x0b\\
    x0c\\x0e-\\x7f])+)\\]))+
```

This expression validates an email address within any text. In this context, it is used to extract the email address from the header of the inline text.

Most of the clients/services have a nameless variation which is more general than the nameful one, that it is used when the name of the sender is not specified in the inline header but only its email. The name can be missing for a big variety of reasons - the sender didn't configure the option to send its name, or the contact is not yet present in the user's contact list - , but when it is, more information can be extracted. That is why both scenarios are covered.

Regarding the date extraction, the same approach was used: identify the date format of each email service/client and then try to validate each format with the current string.

If any of those formats returns a valid date, then there was a match. With this, this prototype is able to extract dates from every inline header. There is just a special case that, due to the header format -where the hours are given in the format of 12 hours instead of 24, but there's no indication if it's AM or PM and therefore that information is lost - the information extracted is not 100% correct because the email service itself loses information when it's creating the inline header. All the identified date formats for these email services/clients are listed below:

- ```
  E dd, yyyy 'at' hh:mm a
  ```

- ```
  dd/MM/yyyy hh:mm
  ```

- ```
  d/M/Y hh:mm
  ```

- ```
  E, M dd, yyyy hh:mm a
  ```

- ```
  dd.mm.yyyy hh:mm
  ```

- ```
  E, M dd, yyyy 'at' hh:mm a Z
  ```

- ```
  dd M yyyy, 'at' hh:mm a
  ```

- ```
  dd M yyyy, 'at' hh:mm a
  ```

- ```
  yyyy-M-dd hh:mm Z
  ```
  ```

```
E, M dd, hh:mm a
```

```
E, M dd, yyyy 'at' hh:mm a
```

```
M dd, yyyy, 'at' hh:mm a
```

```
dd M yyyy, 'at' hh:mm
```

An explanation for the meaning of each of the symbols (p.e. *dd* or *yyyy*) within this context might be found on [Ell15].

As it is easily observable, all headers are different between them having some more radical differences than the others - for example AOL and GMX headers have more than one line, opposing to all the others. To support more email services, it will just be needed to observe its header format and include it in the process.

5.2.3 Principles and relation to Information Visualization

Regarding the information exploration process, it cannot be applied to this prototype given the fact that it's a purely logical feature. Rather it produces information that it's going to be visualized, for example, using other prototypes. After a thread's reconstructed, its outcome is going to be sent to the reading view of the application, to be visualized with its standard features.

Only the data type can be defined here. The data type of the information here received can be classified as *1-dimensional*. This is due to the fact of the content that's received is a unique string, divided into several lines, but in fact it's a single string. Then, this string is splitted and each part is analyzed individually, as explained previously.

5.3 Sociogram

The following section intends to present the last implemented prototype within the scope of this dissertation: a sociogram, as it's referenced in the literature [Han98], that displays relations between intervenients of a thread. Its fundamental aspects and details are going to be described in the next sections.

5.3.1 Concept

The main problem here addressed is the micro social network that is born from a thread where users are connected to each other by their message exchange, having a visual representation. What is aimed to be represented is this social network, the participants and their relation and contribution throughout time - who replied to who, who sent the bigger amount of messages, among other aspects. This representation, besides the mapping to Information Visualization properties and actions that is going to be mentioned ahead, as sociogram. The core of this representation is in fact similar to a network graph representation where nodes are the actors - participants - and the edges mean some kind of relation between them - in this case this relation is a message exchange between the two ends of a certain edge.

5.3.2 Principles and relation to Information Visualization

Before explaining the context, evolution and logic behind this prototype, it's going to be explained the terminology and symbology used. It is as follows:

- **Node** - A node in this context represents a participant of the thread - anyone who sent at least a message and/or was a receiver in, at least, a message. Node size depends on the amount of messages the user sent to the thread. Node's size increases with the number of sent messages to the thread. Each node has a label in the form **"N (A)"** where N stands for the name of the user and A stands for the amount of sent messages the user sent;

- **Link** - A connection between two nodes represents a relation between them. It means that there was at least one reply from one of the participants (represented by an end node) to a message sent by the other participant (other end node). It might be the case that both users replied to each other. Edge's width also grows with this number of exchanged messages between. Each link has a label in the form **"A"** where A stands for the amount of exchanged messages between the two users.

Regarding the size of the nodes and the edges, they are proportionally mapped from a relative scale - minimum and maximum number of sent messages to the thread among all the users and the minimum and maximum number of exchanged messages by every pair of nodes - to an absolute scale (which is going to be explained next). This allows the graph to have an expected and relative aspect - as it is known *a priori* the minimum and maximum sizes for the nodes' size and edges' width - that's independent from the size of the thread. For example, a node with radius 10 might mean that a user sent 10 messages

to a thread with a certain number of messages but might also mean that the user sent 50 messages, in another thread with a bigger number of messages.

Node's radius varies between 5 and 25 pixels. This allows that the smaller node is easily identifiable and still be noticed by the user with small effort. Also, the bigger nodes aren't too big, allowing that a graph with a big number of nodes isn't overpopulated. This values were achieved empirically and had user's approval in testing sessions, as it can be seen in chapter 6.

Regarding node's radius, edge's width vary between 2 and 10 pixels, values that were achieved in a similar process as the one explained in the previous paragraph.

The chosen color for this diagram were not random. The main orange for the node's color was already explained before. Regarding the link's color, it's the color that's directly complementary to the orange - a tone of blue #0D55A6 - , achieving that way a high contrast, being both elements easily distinguishable from each other and, at the same time, visually pleasant. Another reason for the links to have a different color is because this prototype is going to be displayed in a more reduced space than the remaining ones, with the elements being smaller than the tree ones, for example and therefore a higher degree of contrast was required.

- **Overview** - When the graph is firstly displayed, it's zoomed out to a level where the graph is condensed and it's possible to perceive which are the bigger nodes and which are the larger links, for example. Yet, the labels of nodes and links are hidden. An example of this graph can be seen in figure 5.8;

- **Zoom** - Within this context zoom action is literally to zoom in/out a specific area. When a certain level of zoom in is reached, i.e. when the graph is enoughly expanded, labels are automatically shown permanently. When the graph is zoomed out after the same certain level, labels are automatically hidden, being only available on demand. An example of a zoomed in graph can be seen in figure 5.9;

- **Filter** - In this chapter filter is a bit mixed with the relate action in the sense that it's possible to filter (not hiding, but on the opposite side highlighting) related nodes to the selected one. This way, when a user select a node, its direct relations are highlighted (by showing the label, if hidden, and by drawing a stroke on the node) and at the same time relating that node to the ones that have some kind of relationship with it, being possible to navigate in this chain of relations (Node A is related to Node B that is related ... to node N). A result of performing such an action - click an item to filter and show its relations can be seen in figure 5.10;

- **Details-on-demand** - Every node and link have a label, as mentioned before. When

a user click a node, its details are shown (direct relations and label). Also, when the graph is zoomed in, all the labels are shown;

- **Relate** - See previous point *Filter*;
- **History** - Being the possible actions to zoom (zoom in reverses a zoom out action and vice-versa) and select (an item can be selected with a click in the node and deselected with a click in another item or in the empty space), which are reversible and easily tracked by the user;
- **Extract** - Not applicable within this context.

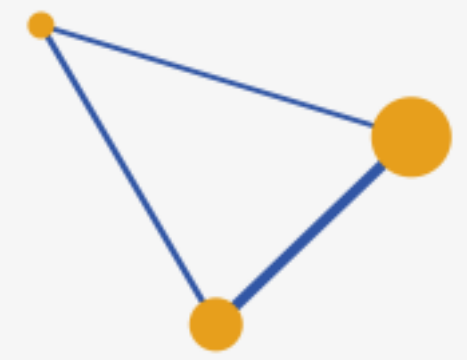


Figure 5.8: Sociogram.

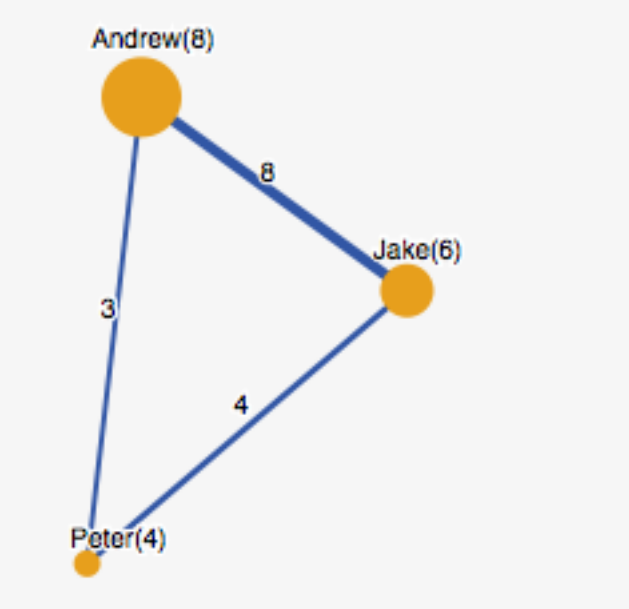


Figure 5.9: Sociogram Expanded

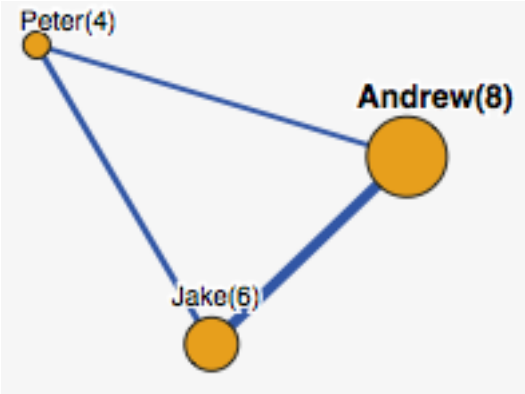


Figure 5.10: Sociogram with node *Andrew* selected.

Regarding the data type used on this prototype, it can be classified as a *Network*. It fits in this category because the elements create a network, where nodes represent each individual and links represent some kind of relation between the users - one user at the end of the link replied to a message from the user at the other end of the link (or vice-versa).

5.3.3 Implementation

To implement this feature a set of alternatives were analyzed and the best one was chosen in order to maximize performance. The main alternatives analyzed were:

- **Native OS X Graphics** - As it was mentioned before for other situations, native implementation of this feature was discarded due to the high learning curve and technological complexity overhead that this implementation would cause. This lead to choose OS X's webviews and implement this feature using web technologies;
- **Pure SVG drawing** [15] - This option requires to handle all the drawing and interaction by hand. As the goal is to draw an interactive network graph, this option would require a lot of work done by hand when there are better options that facilitate and smoothen this process. Therefore, this option was also discarded;
- **Raphaël** [16] - Raphaël's library is mostly a wrapper over SVG elements and their manipulation. Therefore does not automate or facilitate the hardest part of this task that is to draw the graph itself, so it was also discarded
- **D3.js** [17] - The chosen option was to use the Javascript library D3.js. It is recognized as one of the best and most complete Javascrip visual libraries that are available

[15] http://www.w3schools.com/svg/
[16] http://raphaeljs.com/
[17] http://d3js.org/

today. Despite its loading time (approximately 3-4 seconds) - due to the Force-directed layout [FR91]- it was the chosen option because it handles the graph layout pretty smoothly and wraps several actions and interactions in a pretty clever and easy-to-code way. Also it's highly customizable and there are many examples over the web that go along with a pretty powerful and detailed documentation that allow a fast learning process.

5.3.4 Main features

The main technical features of this prototype are:

- **Auto-layout** - Taking care of the force directed layout it displays the graph in an optimized layout, minimizing link and node overlapping;
- **Relationship highlighting** - When a node is selected, it's highlighted from the remaining nodes by showing its label in bold and a stroke around the node. Also, the nodes that are related to the selected one are also highlighted - with a similar stroke and its label (but with regular formatting, not in bold) - being easily perceived the name and amount of messages sent by the direct relations;
- **Details always visible with a certain level of zoom** - When a certain level of zoom in is achieved, node's and link's label are visible all the time. When the graph is zoomed out again above the defined threshold labels are automatically hidden again and are only displayed on demand;
- **Automatic node and link size** - Given the relative amount of messages in the thread, sent by each user and exchanged by every pair of users, it's then scaled to a standarized range of values and elements' size is assigned accordingly to this absolute scale in relation to the relative amount of messages previously described.

Chapter 6

Prototype Validation and Results

6.1 Methodology 62
6.2 Process 62
6.2.1 Pilot test 63
6.2.2 Personas 64
6.3 Results 64
6.3.1 Tree and Timeline 64
Observations 66
Improvements after evaluation 66
6.3.2 Chat 67
Improvements after evaluation 71
6.3.3 Thread Reconstruction 72
Observations 75
Improvements after evaluation 75
6.3.4 Sociogram 75
Observations 77
Improvements after evaluation 77

This chapter is going to describe the process used to test and validate the prototypes here proposed along with the achieved results. This process fits in the category of Usability Testing [Unk15b, NK93], where users were brought to the process to test the prototypes and provide feedback that's not extractable from analytical data.

The next sections will describe in detail the methodology and process used as well as the results extracted from the testing process.

6.1 Methodology

The approach here followed was to perform usability tests. Their goal is to give a set of tasks within a test scenario to a group of users (who will perform them individually) and they should complete them in order to evaluate the features or the interface covered by the test plan, and further evaluate their performance according to several metrics. According to Nielsen [NK93, Nie12] the optimal number of users to participate in these tests is five (5) - given the trade off of what it costs to bring another user into the test scenario and the benefits that would come from it.

Each prototype had one round of tests and then with the user's feedback the prototypes were improved and lead to the final version presented in this document or in future work's design, if it was technologically complex for the remaining time. It was not possible to follow the initially planned approach of merging the prototypes into the application, send them to production getting feedback from real users and then iteratively improve them, due to the fact that the product where this dissertation's inserted is not yet on the market. With this in mind, the followed approach was to implement the prototypes to a more stable version and then evaluate them without a complete environment but rather a prototype of an email reading view that was functional. Due to this, the prototypes were evaluated stand-alone and not in the context of a whole app. The reading view's design wasn't top notch and the content - as the test scenarios were controlled - was suited to that visualization so that the fact that this reading view design wasn't a final version wouldn't affect the evaluation of the prototypes.

The experimenter - person that conducts the evaluations - was the author of this document.

6.2 Process

To evaluate the prototypes presented previously, usability tests were conducted. These tests were executed in two different points in time, where in the first one were evaluated the Tree & Timeline, Chat and Thread Recreation prototypes and in the second only the sociogram was evaluated. This was due to the fact that not all of the prototypes had enough maturity to be evaluated at the same time and at the same time and, by evaluating some of them sooner, there was more time to improve these prototypes with the given user feedback.

The tests were formative [Rub08] - meaning that the test was not performed on top of a final version but rather on an under-development one - that had as a goal to measure the usefulness of the concept and its pros and cons on the user's point of view.

The chosen method to perform these tests was the Concurrent Think Aloud method [NK93] where the user is invited to say what's on their minds and discuss with the experimenter. This allows the experimenter to take notes and insights not only on the user's actions but also his doubts and opinions on the evaluated matter. In this phase of testing, where all the input is valuable to improve and, more important than this, to validate the usefulness and value of these proposals this was the most adequate method.

All the evaluations had a prepared thread scenario. All of these threads can be seen in appendix N where in some threads the messages are identified with a header in the form **"A B (C)"**, where A is the id of the message, B is the sender and C is the message to which the message with id A is a reply-to. When this identifier is not present, the messages follow a reply-to-last pattern and there's no need for identifiers. Also, all the evaluations had a script with a brief explanation of what was the scope and goal of the exercise and the exercise itself. These scripts can be found in appendices A, D, G and J.

6.2.1 Pilot test

Prior to the real tests, it was performed a pilot test (as it's suggested by [Unk] and Nielsen [NK93]) before each one to validate the test, clarify if there were no typos or errors on the script, if the questions were clear and that would go straight to the point and if the exercise itself and its goals were clear and easily perceivable.

This pilot test was executed by a user who has less technical knowledge than the the target users that performed the tests and that are intended to use these prototypes when they're available in the product. The main outcomes of this pilot test were the identification of some typos in the scripts and the introductory speech, besides having also a better notion on each test's duration. Regarding the typos they were rapidly fixed. Regarding the introductory speech, prior to the pilot, it was a "light" speech in the sense that it was assumed that the users were familiarized with all the concepts - what's a thread, what's a reply pattern within an email thread, among other small details - addressed in the exercises. After, it was necessary to re-evaluate this speech in order to make clear some aspects that are crucial to perform the exercises prior to their execution and not during the exercise itself. Nevertheless, the feedback was really positive, pointing that the exercise fulfilled the pre-established goals, it wasn't boring for the users to perform it - putting together the exercise itself and the interaction with the experimenter - and the goals were also understandable by the user.

6.2.2 Personas

The tested subjects were all male university students, currently in the first or second year of their masters in software engineering. with ages between 21 and 28 years old. The tests were performed for this target audience due to the fact that they all have an above the average skills with IT products (derived from their field of studies) and, as the product where the prototypes are intended to be integrated core aim are power users (or expert user as [NK93] calls them), this was the perfect target audience, as suggested on [Unk15a]. Besides their technical knowledge, their easiness of getting used to knew concepts and use them right away and characteristical critical spirit were a decisive factor when choosing them as the target audience for the tests.

All the test subjects had a similar profile and background so that the results could be consistent and therefore relevant for this dissertation and for the company (as these test subjects were similar to the target audience of mailcube's application).

6.3 Results

Now there are going to be presented the results of the evaluation sessions of the prototypes. These results intend to measure time that users took to execute the intended tasks and their accuracy - which is important to measure if products indeed help users perceiving information correctly - and easiness to use in comparison to standard methods, besides more subjective and implicit information. For each prototype there'll be presented its evaluation results and an analysis to these same results.

6.3.1 Tree and Timeline

The tree graph and the timeline prototypes were evaluated together as they intend to help the user achieve the same goals within the thread and they're placed in the same view. The evaluation questions were directed for both prototypes, as both were needed to answer the questions correctly/efficiently. The script given to the users, the questions' form and the full spreadsheet of answers can be found in appendices A, B and C respectively. The main ideas of the evaluation was to understand if the prototypes would help users identify the distribution of messages along time, relations between messages or reply patterns.

There were two tests to evaluate this prototype - one of them to answer the form without the prototypes and another with them (threads' content was different but their context was similar) -, then evaluate the users' answers and compare the results. The results were very positive and validated the proposed concept. The percentage of correct answers using the prototypes was 100% to all users. On the other hand, the percentage

of correct answers of one of the users using the version without the prototypes dropped to 70% (with the remaining ones conserving 100%). In this second part, the user got a question wrong and only guessed another because he was biased by the previous exercise (answer the question with the new prototypes) but it was a guess, not a certain answer. Therefore it was only counted as 10% correct.

One of the questions asked was the easiness to answer to the questionnaire with or without the prototypes. The answers were pretty clear, as figure 6.1 shows: all the users thought that answer to the questions with the prototypes is easier than without. This validates the idea that the prototypes indeed help solving the addressed problems as the subjective opinion of the users and the objective results - 100% of correct answers with the prototypes versus only 70% without the prototypes -point in this direction this.

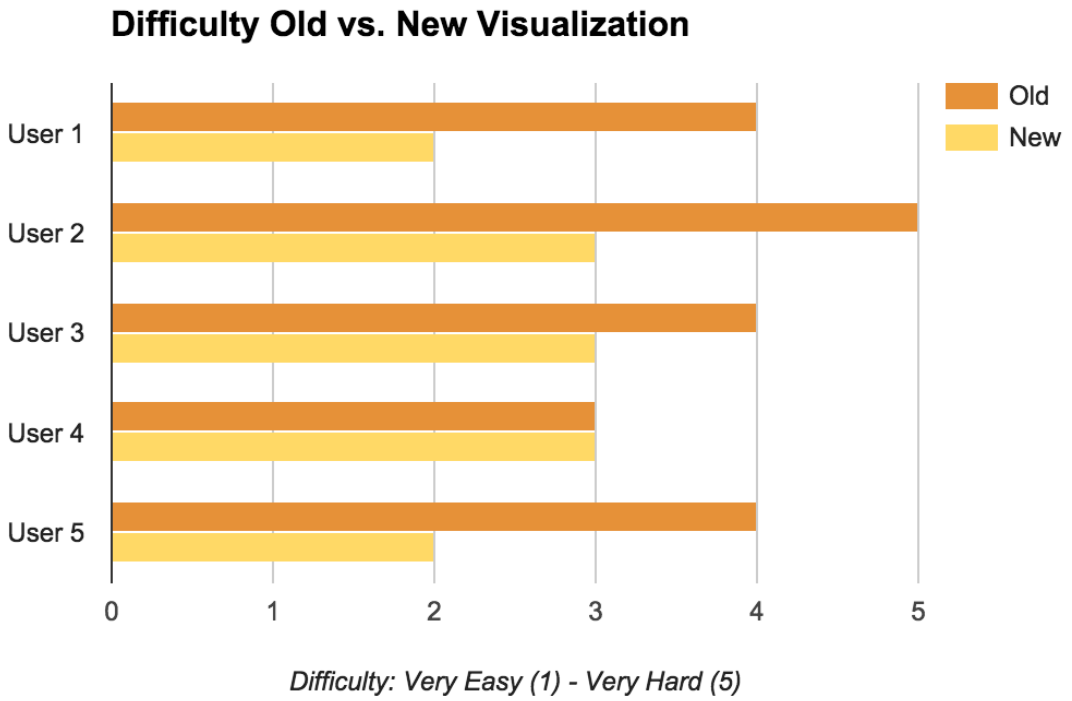


Figure 6.1: Users' opinion on difficulty to perform evaluations tasks with old and new visualization.

Regarding the time taken to complete the tasks, all of the users took more time when answering using the prototypes than without using them. This was due to the fact that, as the concepts were new, the users while were performing the tests were asking the experimenter some questions regarding the features, how to use them, asking for a reminder about how to do a certain action which lead to a longer test duration. The average duration of a test with the prototypes was 8 minutes and 22 seconds. On the other hand the average duration of a test without the prototypes was 5 minutes and 37 seconds.

Observations

The last question asked in the questionnaire was for the users to make relevant observations and suggestions regarding this view. There will be presented both the comments submitted by the users and notes taken by the experimenter from the comments users made out loud.

- *« Scroll should be synchronized in reading and tree view »;*
- *« Tree View is more useful in big threads »;*
- *« Click on cluster should dismantle the cluster and not just "zoom in" » ;*
- *« Merge tree and timeline, by displaying relations between nodes in time; »;*
- *« Tree view could display hierarchy/indentation »;*

Improvements after evaluation

This section intends to present the improvements made from the first round of tests within the application based on the users' suggestions and empiric observations.

Regarding the list of the emails - it's not directly related to the prototypes but it belongs to the view where the prototypes were inserted and therefore improvements were made as well based on the gathered feedback - all the text fields were marked as uneditable, so that the user wasn't able to alter data. This was a detail that the author was unaware of and that was only noticed by the users. Also the size of the font of the content of the message was lowered because users thought that it was confusing to distinguish sender, from date from content as they all looked similar. Also, date format was also updated from a format that, for the users, was too long and hard to perceive at glance, to a shorter one, just displaying the essential information.

Regarding the timeline some of the suggestions were implemented some were not due to a variety of reasons, that will be explained ahead. The first improvement was the maximum and minimum level of zoom that is allowed in the timeline. The maximum level of zoom was until two days before and after the dates of the initial and end dates of the thread. According to the users, this level of zoom out was too high, being redefined to one day instead of two. This way its goal is still fulfilled - there's a margin so that the first and last events don't be hidden (fully or partially) -, but it's not excessive. Also the minimum level of zoom in was updated given the fact that it was possible to zoom in until the microsecond. This level of zoom in was updated to allow only exploration until the seconds level. The last improvement was related to the item selection. Users complained about the fact this feature was graphically dislocated from the app, being ugly and a bit unperceivable. With this in mind, it was updated to fit the color scheme of the reading

view, being less intrusive but more easily noticed. Nevertheless there were two suggestions that were not implemented. The first one was to merge the connections and the relations from the tree graph with the timeline. This was not implemented due to the fact that the author believes that it would be a technologically epic task which wouldn't be suitable for this dissertation timeframe. Although the idea is promising, due to time limitations and technical complexity, it was not possible to integrate this suggestion right away but it will be in the *icebox* of future work associated to this prototype it will be referenced as future work and hopefully will be tested and validated. The second unimplemented suggestion was to "uncluster a cluster" with a single click instead of just zoom in a little bit, also due to time limitations.

Regarding the tree graph users' main and biggest concern was related to the scroll synchrony between the standard view and the tree graph. The problem was that when the user scrolled the view with the messages list, the view with the tree graph did not accompanied it properly and the visual effect that was intended to achieve with it, that was to associate each node to an element with the message content, is not accomplished and the prototype becomes somewhat useless. After the evaluation the bug was fixed and at the moment is properly and fully functional. The other relevant suggestion - made by 3 out of 5 users - was to indent the messages or the nodes to show the messages' hierarchy, when applied. Nevertheless, the author believes, based on the observation of the users' behaviour, that this demand was due to the fact that the tree graph was distant from the list with the messages. As the next iteration of the design addresses this distance problematic, this suggestion was not implemented now. If, in a future evaluation after the new design is implemented, the suggestion remains then it'll be considered more seriously. Besides these suggestions, all of the users acknowledged this prototype, specially for bigger threads.

A note only to an observation from the users that performed the evaluation of the sociogram - which had nothing to do, in terms of exercises or script, with the tree and timeline prototypes but as the users had to explore the reading view, they interacted with these prototypes - when confronted with the bigger picture of the reading view, asked what those "elements" (tree and timeline) were and, after a brief explanation they described them as "very smart" and "could be useful in many scenarios".

The final result of the reading view, having these prototypes integrated and improved can be seen in figure 6.2.

6.3.2 Chat

The present section intends to describe and present the results of the chat prototype evaluation. This evaluation was conducted in the same time period as the tree & timeline

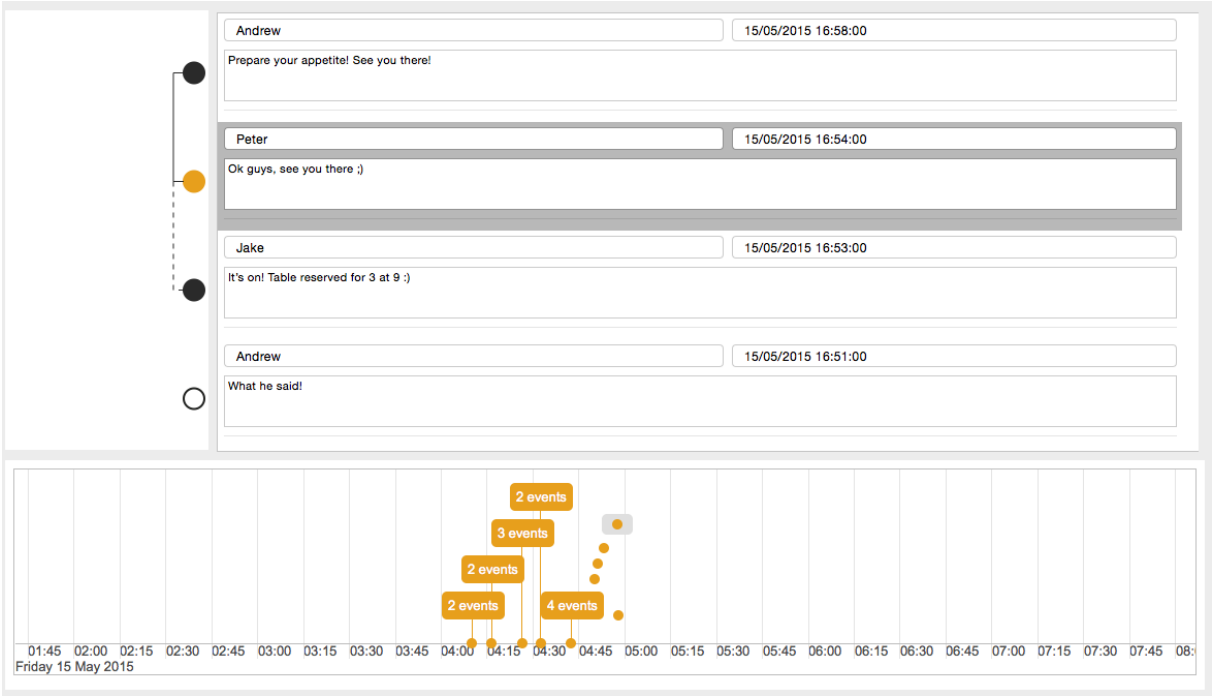


Figure 6.2: Design of reading view after improvements.

and thread reconstruction evaluation. This evaluation consisted in presenting the users with a pre-prepared thread and a form with questions related to this thread. The user had to answer the objective questions using the chat view and to the subjective ones using their personal experience with other email systems. The results, which are overall very positive and that validate the proposed concept, are going to be presented next. In appendices D, E and F can be found the script given to the users, the questions' form and the full spreadsheet with the complete set of answers, respectively. The exercise to evaluate the chat was composed only by a single thread and a set of questions (opposed to the two set of questions when evaluating tree and timeline, for example), where the goal was for the users to compare their experience using this chat versus their standard use of email clients.

The main ideas to validate with this exercise were if it was easier, for threads that fit the profile of a chat, to get an overview of the thread, identifying easily messages sent from others and from self, and also to send messages.

In the first place is relevant to say that all the users answered correctly (100% accuracy) to all of the objective questions using this prototype, identifying correctly the number of distinct participants and the number of messages from self. The average time that the users took to answer to answer to the questionnaire was 2 minutes and 51 seconds.

Regarding the question "Is it easier to distinguish your messages from other intervenient's messages with a chat Visualization rather than a standard Visualization?" the answers were pretty clear: all the users answered positively, as figure 6.3 indicates.

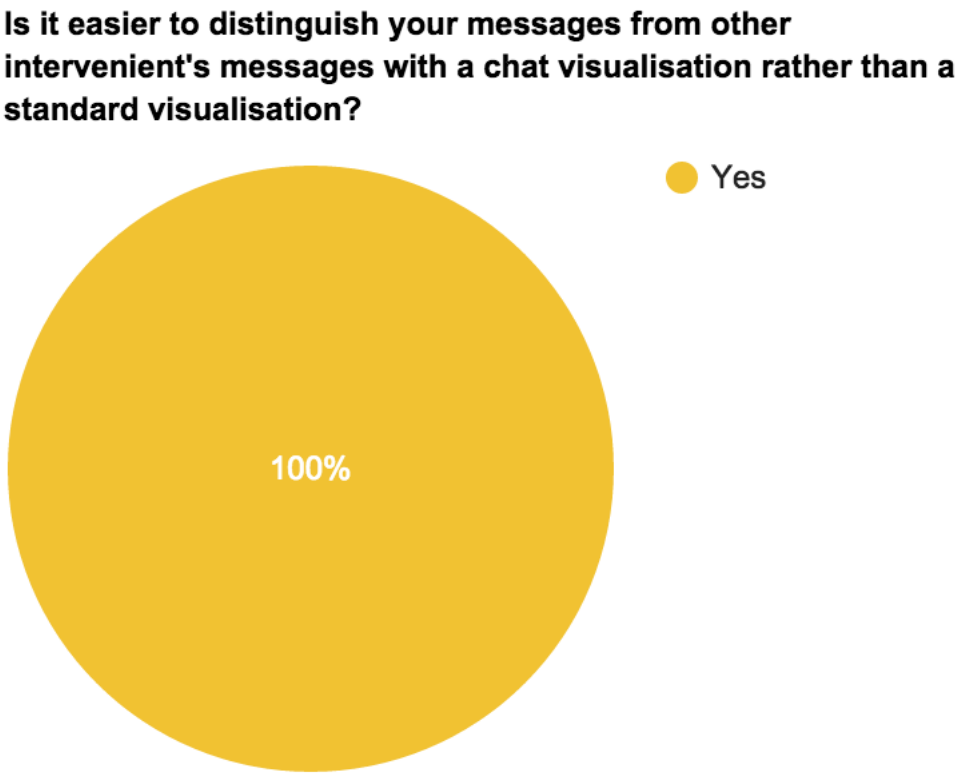


Figure 6.3: Users' opinion on difficluty to distinguish messages from self from messages from others.

Next, the users were asked if it was easier to send messages with this prototype rather than with the standard method to reply to emails in current email clients. Again, the answer was consensually positive, as it can be seen in figure 6.4

Regarding the usefulness of this prototype to visualize threads that fit the chat prototype, again all of the users answered positively, as it can be confirmed on figure 6.5.

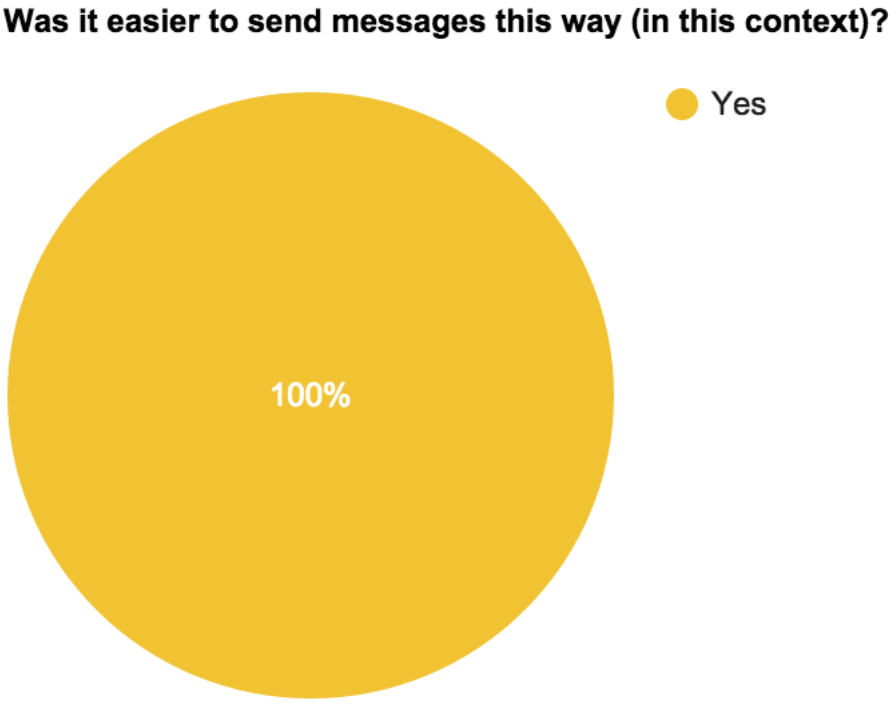


Figure 6.4: Users' opinion on difficulty to send messages with this prototype versus standard email clients

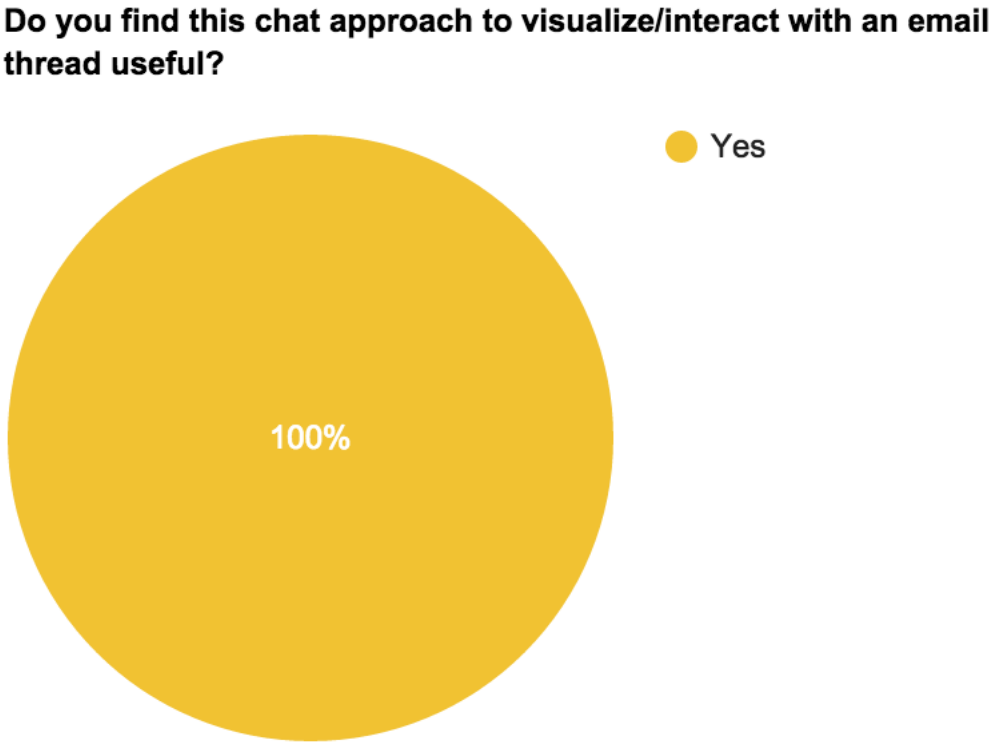


Figure 6.5: Users' opinion on chat prototype usefulness.

In the sequence of the last question the next question asked was "Why is this approach useful?". Here are presented parts of each user response. The full response can be seen, as previously stated, in annex F.

- *«Easy to glance through content, and identify participants.»;*
- *«Translates email messages exchange into a simple conversation into an easily digestible environment.»;*
- *«For short conversations where all participants are online at the same time, this method is more efficient due to less overhead»;*
- *«It streamlines the interface in a way that will facilitate the interactions between the users.»;*
- *«If the messages are short, the chat system is easier perception-wise than an e-mail thread.»;*

Observations

The last asked question in this questionnaire was for the users to make relevant observations or suggestion to improve this prototype. The most relevant content of this input, compiled with the experimenter's notes on what the users said out loud is presented next.

- *« Hide custom signatures in chat view, as it's unnecessary information »;*
- *« Adapt box size to message width (like Facebook or Skype) »;*
- *« Add label do distinguish separated periods of time »;*
- *« Time's missing on messages prior to "yesterday" »;*

Improvements after evaluation

After the evaluation, some minor improvements were implemented into the chat prototype. These were related to the date formats that were presented below the area where the content is displayed. Previously the hours and minutes were only shown in messages sent in the present day. Due to this, the date label from messages sent "Yesterday" or the days before were updated to show hours and minutes, besides only the date.

The other main improvement that can be done is regarding the box size. To update this feature, there's already a design proposal based on the users' feedback that will be presented in the Future Work section. It was not implemented right away due to time limitations, given the technical challenge that arises from here.

Besides these two suggestions, users really liked the prototype and validated it, as it can be seen from the very reduced number of critics or improvement suggestions.

The final result of the chat view, after the improvements, is as figure 6.6 shows.

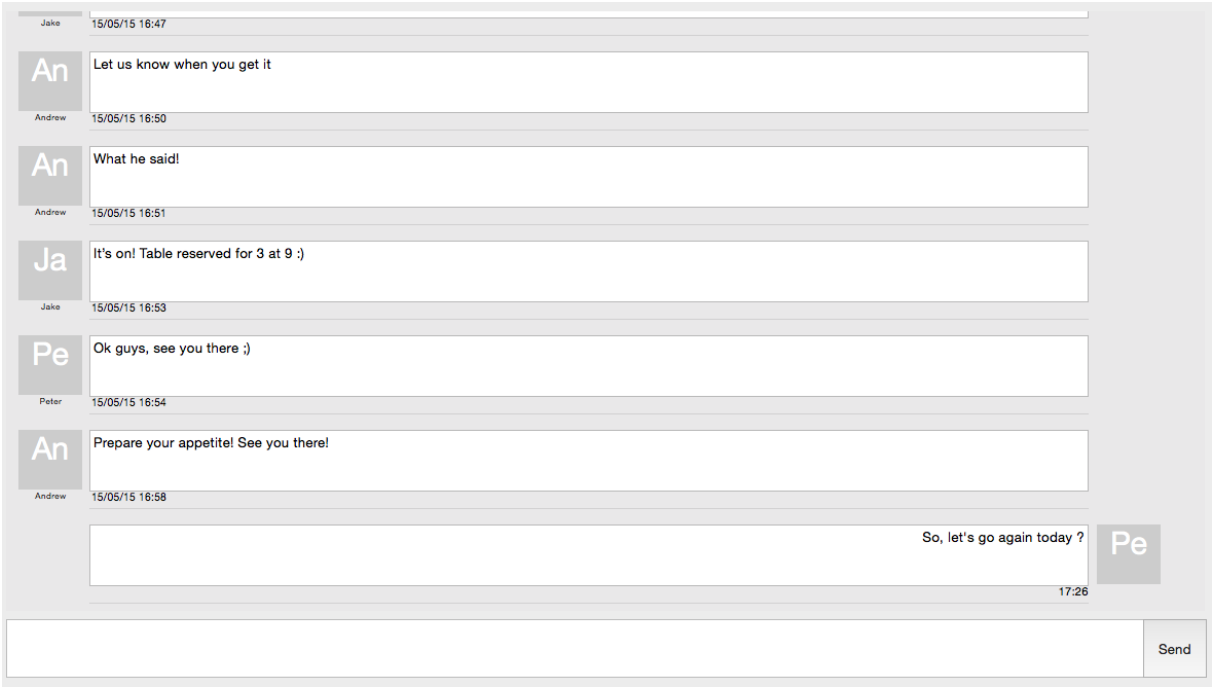


Figure 6.6: Final version of the chat prototype.

6.3.3 Thread Reconstruction

Here it's going to be described the evaluation test that was performed to the thread reconstruction feature. This feature was also tested, as previously stated, along the tree & timeline and chat prototypes. This exercise was the last one amongst the three exercises mentioned and so, when it was time to perform this one, users were already familiarized with the scope and the previously presented tools that were at their disposition. In appendices G, H and I can be found the script given to the users, the questions' form and the spreadsheet with the complete set of results.

The test consisted of two parts, where the users had to answer a questionnaire about a certain thread using, in the first part, a modern web email [1] client and then, in the second part, the same reconstructed thread by the implemented prototype within the application. The goal of this exercise is to confront the user with the daily situation of a thread that was "generated" by two users and then forwarded to a third intervenient, which had to extract information from the inline content, formatted as it's shown in figure 6.7 and

[1] Gmail - gmail.com

with a reconstructed thread where the user also had to extract information but that's represented like a regular thread, like the user's been involved since the beginning.

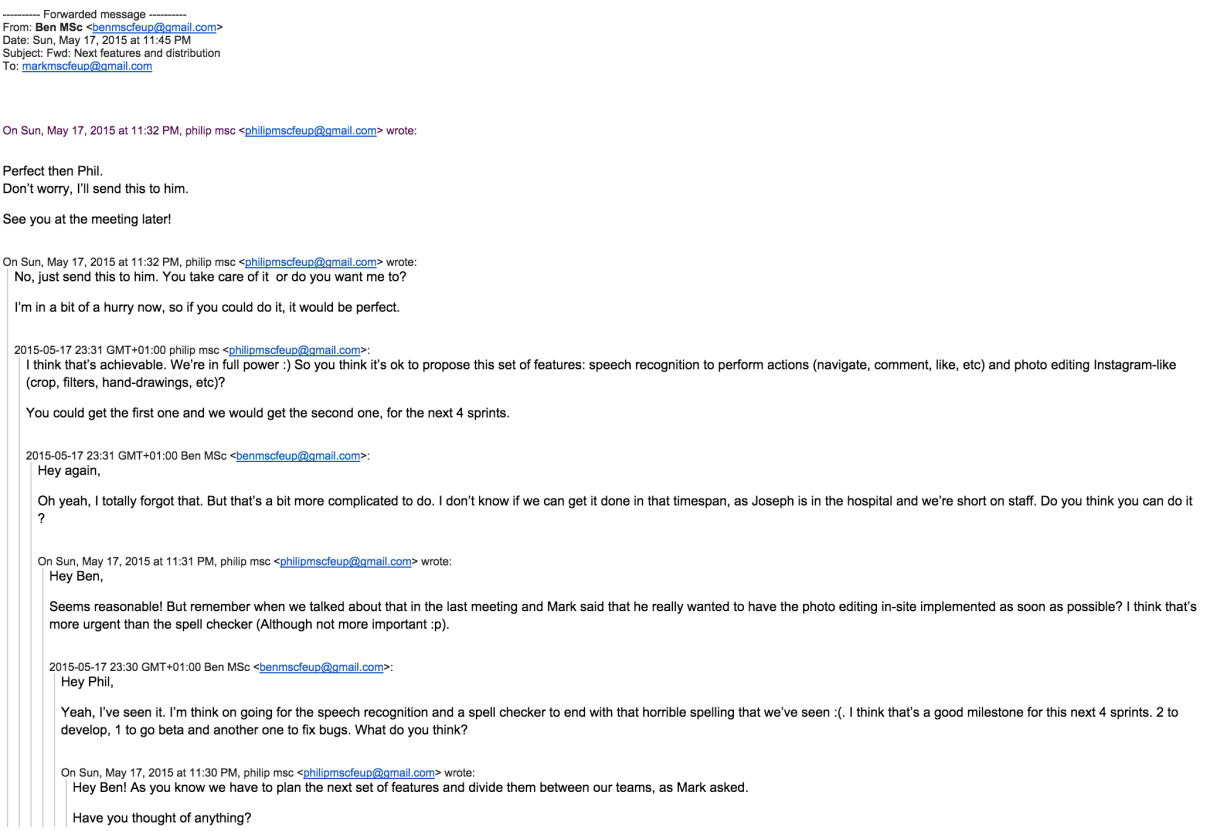


Figure 6.7: Visualization of the forwarded thread in an web email client

Again, the overall results were extremely positive. Users were faster answering to the questions by using the reconstructed version than using the web one - 3 minutes and 15 seconds versus 4 minutes and 35 seconds.

Regarding the accuracy rate, all users had 100% of correct answers when using. On the other hand, when using the web version, one of the users only guessed correctly 75% of the answers and another user guessed 88% - this means that one of the answers was only partially correct. Regarding the question "Identify if it was a long discussion (regarding time, not number of messages)", user replied short, but indicated (voluntarily) that it lasted 3 min, when it lasted 2 -, which points to the direction that the reconstructed thread helps users achieving more consistent and correct results.

When users were asked to indicate the difficulty of executing the tasks with both the web and reconstructed versions, their answers were very clear: answer to the questionnaire with the reconstructed thread is easier than answering to it using the inline content of an email message, as it's graphically translated by figure 6.8.

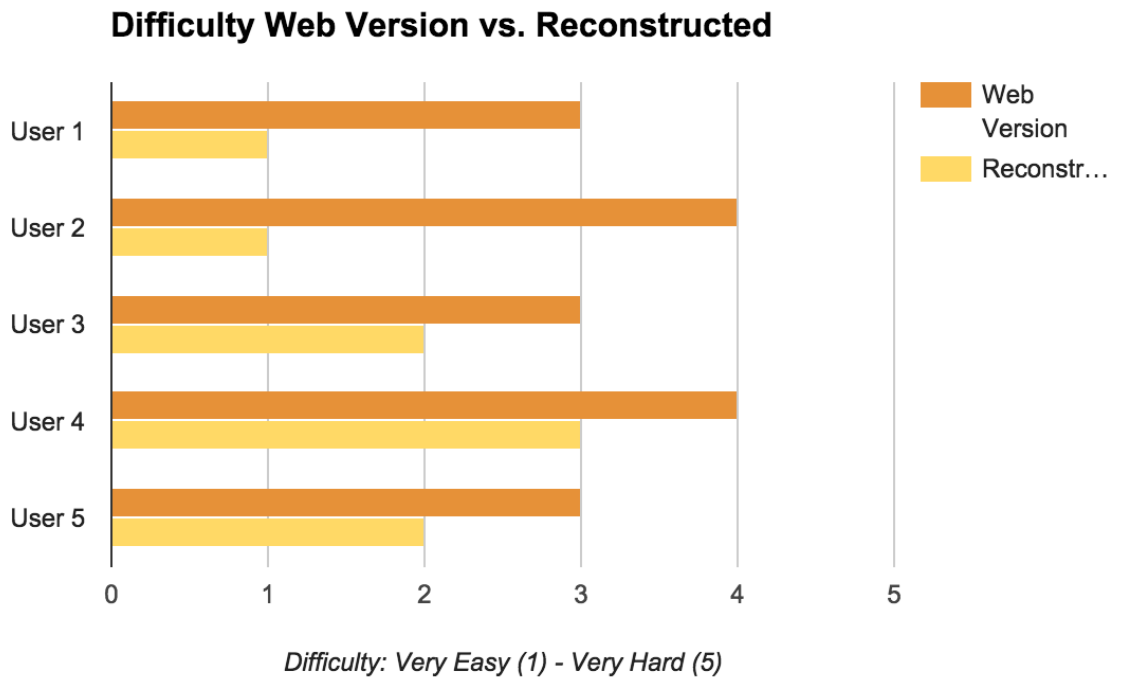


Figure 6.8: Users' opinion on difficulty to extract information from the forwarded thread by using the web version and by using the reconstructed thread.

Finally, regarding the usefulness of this prototype, all the users agreed that it's indeed useful, as figure 6.9 indicates.

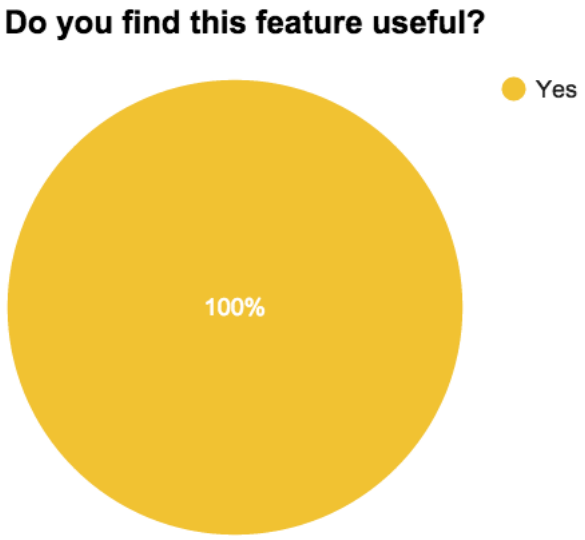


Figure 6.9: Users' opinion on usefulness of thread reconstruction

Observations

Regarding the last question, where users were asked to make comments about the prototype or suggest some improvements. The quotes shown ahead are the most relevant (full comments can be seen in appendix I).

- *« The access to different views really helps when going for an overview analysis like this.»;*
- *« With this style (web version) it's much harder to read and identify certain things.»;*

It's relevant to mention that most of the users thought this feature *"really clever"* and that solves a problem that the users were not aware of, because this type of feature is untouched within the email environment since its conception.

Improvements after evaluation

As this prototype was not a Visualization one per se, but rather a more logical feature - parse text and extract information - there weren't a lot of possible suggestions. The only worth mentioning was from a user that suggested that older messages should be displayed at the top and not at the bottom. Nevertheless this was a strategic decision made by mailcube that applies to the product. Also, the reason behind this decision is mentioned and explained previously in this document. With this in mind, the experimenter explained to the user why this suggestion and decision were made and then he understood it.

6.3.4 Sociogram

This section intends to present the achieved results from sociogram's evaluation. This feature was tested individually due strictly to the fact that this prototype got ready to be evaluated later than the others. This evaluation was composed by two parts. In the first users had to reply to a form based on the reading view aforementioned (see figure 6.2, without sociogram's aid. Then, in the second part users had to reply to the same form, about a different (but similar) thread, but using sociogram instead of the reading view. In appendices J, K and L can be found the script given to the users, the questions' form and the spreadsheet with the complete set of results.

The first relevant fact to enhance is that, in the first part of the evaluation (without the sociogram), 60% of the users got at least one answer wrong whilst in the second part, using the sociogram, all of the users got all of their answers right, which indicates that, using the sociogram, users extract *better* information.

Regarding the time taken to complete the tasks, users took in average significantly less time to perform them using the sociogram - 2 minutes and 44 seconds - than without

using it - 4 minutes and 34 seconds. This, allied to the percentage of correct answers, that it was much easier for the users to extract correct information from the thread using the sociogram than without using it. Also, all of the users thought this prototype useful within the context of an email thread.

The answers to the question "Do you think this feature should be a complement to standard thread visualization or an alternative?", answers were mixed, and the solution achieved that'll be implemented in the next phase of this prototype is to make this sociogram togglable - which means that the sociogram will be presented as a complement, but that's not always visible, meeting users suggestions and observations.

Regarding the difficulty to complete the tasks, all of the users thought it was easier to complete them using sociogram except for one, that thought that was easier to perform the tasks without the sociogram, as figure 6.10. When asked why he thought it was harder to execute the exercise using sociogram he said that it was due to the fact that he had to parse all of the new concepts and using them right away. Nevertheless he also said that he had more confidence on his answers using the sociogram. This fact revealed to be correct as he got 2 out of 6 questions wrong when not using the sociogram.

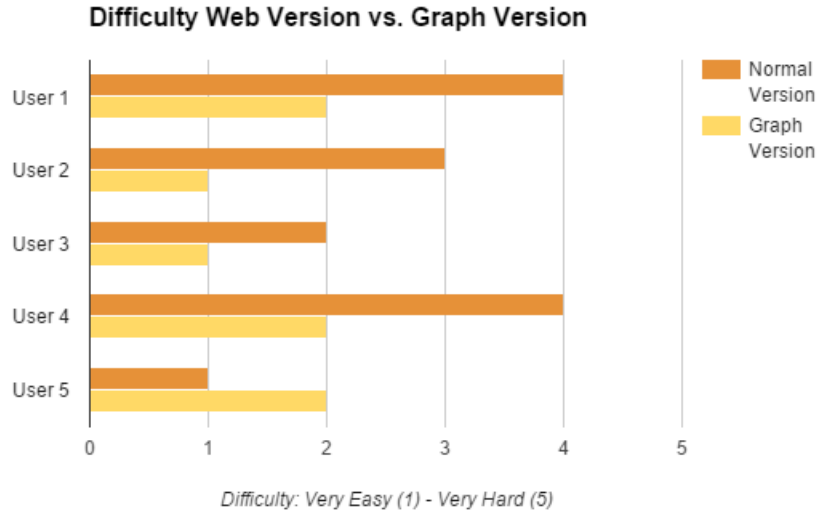


Figure 6.10: Users' opinion on difficulty to complete tasks when not using sociogram and when using it.

Lastly, it's relevant to notice that all of the users thought this prototype was useful and that fits its purpose, as figure 6.11 demonstrates.

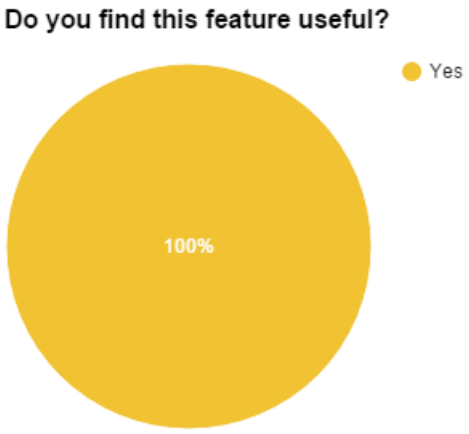


Figure 6.11: Users' opinion on sociogram's usefulness.

Observations

Last question of the form was asking the users for some relevant observations or suggestions about the tested prototype. The more relevant observations/suggestions are presented next.

- *« Show more information in the connections »;*
- *« Feature should be used as a complement when there aren't much users. When there's a bigger number of users (10/15), should be an alternative. »*

Improvements after evaluation

The last version of the prototype, for now is very similar to the evaluated version. After the evaluation, the biggest difference is regarding the colors of the elements. They were changed to use the aforementioned orange and its complementar color - mentioned in chapter 5.3.1 -, which allows for a higher degree of contrast and smoothens the effort performed by the user to extract information.

An example of the final state of the sociogram prototype can be seen in figure 5.8.

Chapter 7

Conclusions

7.1 Future work 80
7.1.1 Tree & Timeline 80
7.1.2 Chat 81
7.1.3 Thread reconstruction 81
7.1.4 Sociogram 82

The balance of the produced work in this dissertation is extremely positive. As the first chapters indicate, there's a huge gap between what the email ecosystem offers to the users and what they demand or need from it. This gap has a big range, that goes from artificial intelligence and data mining to information visualization, for example, where this dissertation fits. The work here developed was more directly related to information visualization but also had a component of text mining and design.

The gaps that were addressed with the five implemented prototypes were thread visualization, lack of context, message comparison, information extraction and inferring. After the development of these prototypes, they were validated with users and improved based on their feedback, leading to the documented version along with the suggestions - that weren't possible to implement at the time - mentioned in the future work section.

These prototypes took into account the principles suggested by the two studied vision theories to empower elements distinction from each other when needed- preattentive vision - and maximize item drawing into the space and making clear the line line that separates the elements and their organization from space where they were drawn - Gestalt theory.

The results from these evaluations were extremely positive as they point that these prototypes are feasible solutions for the addressed prototypes and are useful and usable in the daily email usage. It was also gathered a considerable amount of information regarding

the gaps that these prototypes still have - besides the big amount of improvements already performed during the process and after the evaluations -, which were incorporated in the design proposals that can be seen in the future work section.

7.1 Future work

The future work for the work here developed relies on implementing the next design - that includes a mature version of the suggested prototypes aligned with the design of mailcube's application and user's suggestions that aren't yet implemented at the moment - and once again test the result of this integration in order to fully validate this merge.

The design of the prototypes for the next stage is as it can be seen in the next figures.

Please note that the content of these prototypes is not real (might not be coherent) but rather its only purpose is to illustrate how the design will be.

7.1.1 Tree & Timeline

On figure 7.1 it's presented the reading view, that includes the timeline and the tree graph. There are two main improvements to apply here - besides the already implemented ones - which are to apply the new design, presented next, and try to merge the tree and timeline - show relations between messages in the timeline's nodes.

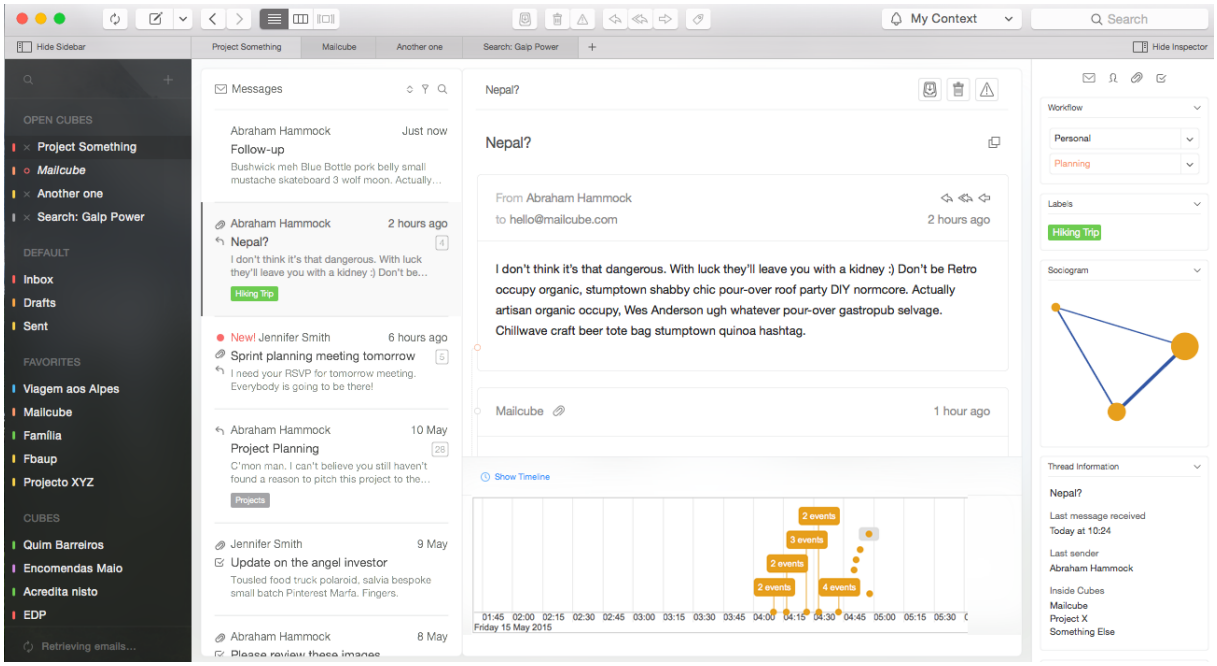


Figure 7.1: Design for tree and timeline prototype (main reading view) next iteration.

7.1.2 Chat

The next version of the chat prototype is as figure 7.2 shows. Main features to improve, besides the design, are the identification of who where the receivers of a sent message, and message box's size.

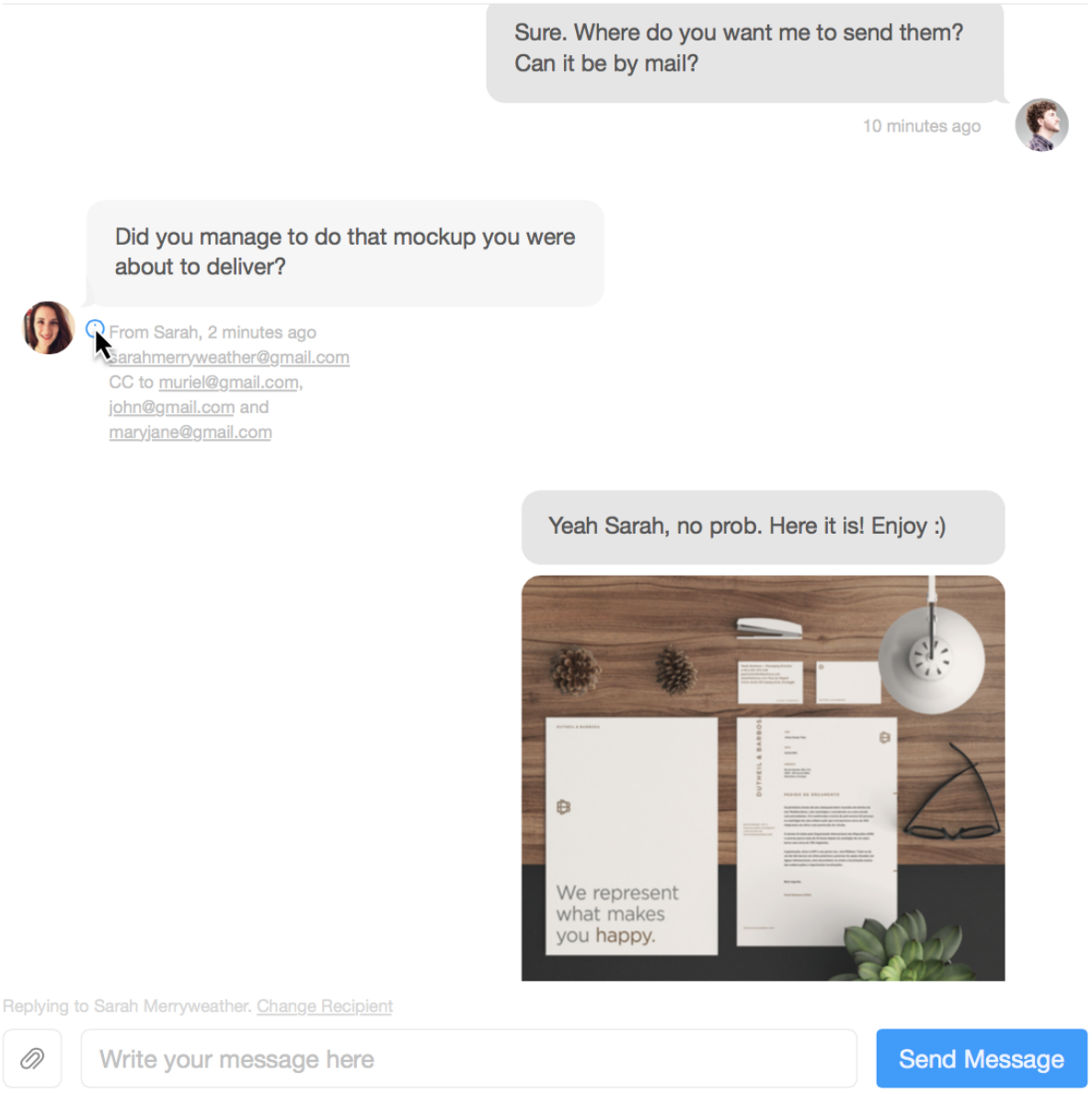


Figure 7.2: Design for chat's next iteration.

7.1.3 Thread reconstruction

Regarding the thread reconstruction prototype, the only improvement that can be done is to add support for more email services / clients, by doing some more empirical experimenting.

Besides that fact, it can be said that this prototype is ready to go to the market.

7.1.4 Sociogram

Lastly, regarding the sociogram, its design is complete. The next step is to merge the information inferred from the thread with information from social networks (namely LinkedIn [1]) in order to offer a bigger context to the user regarding relations with the other intervenients beyond the thread. Displayed information in links should also be revised, in order to offer more detailed information to the user rather than just the number of exchanged messages between the two end nodes of the link. Regarding the integration in the application (complement vs alternative), it's placed in the bar at the right of the reading view on figure 7.3, which is togglable (in the context of mailcube's application's called Inspector), and that way sociogram is shown as a complement to the reading view but it does not have to be visible all the time.

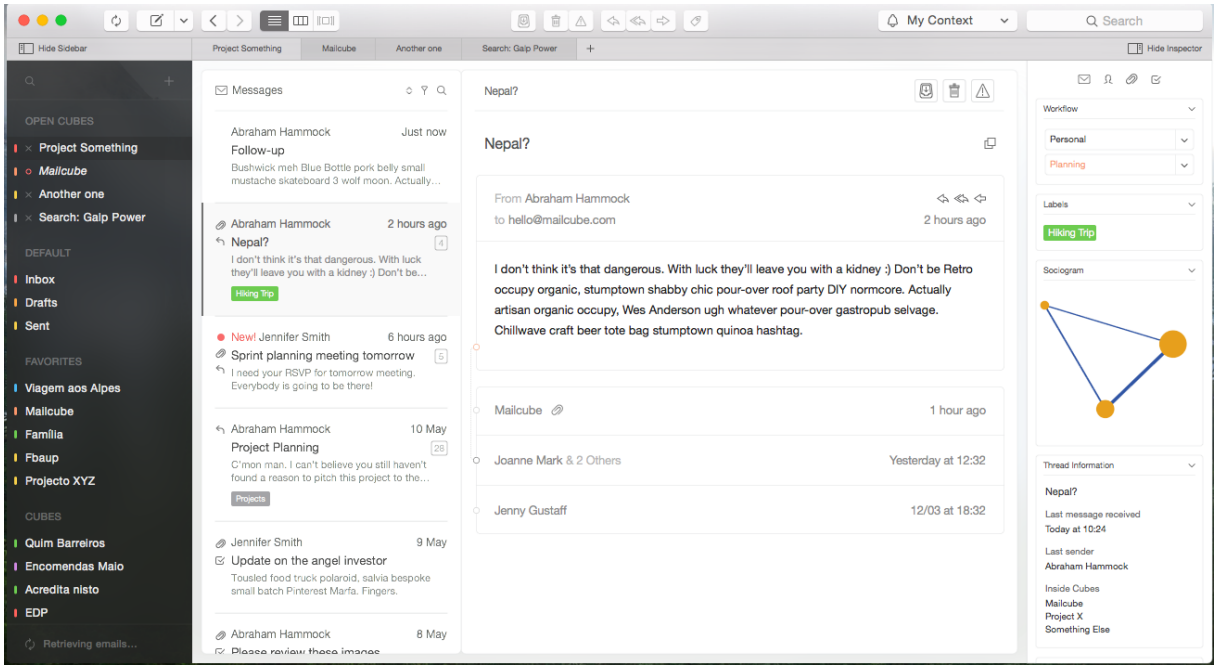


Figure 7.3: Design for sociogram's next iteration.

Based on the presented sketches and considerations, the prototypes should be improved and re-evaluated in order to validate their latest version. After this, they're ready to be taken to the "real" market and ultimately validated.

[1] linkedin.com

References

[ABYG07] Omar Alonso, Ricardo Baeza-Yates, and Michael Gertz, *Exploratory Search Using Timelines*, Proceeding of the 25th Annual SIGCHI Conference on Human Factors in Computing Systems (2007), no. 1, 23–26. Cited on p. 43.

[ACM12] ACM, *The 2012 ACM Computing Classification System — Association for Computing Machinery.*, 2012, http://www.acm.org/about/class/2012?pageIndex=1. Last accessed 16-January-2016. Cited on p. 6.

[Appa] Apple Inc, *Cocoa - os x technology overview - apple developer.*, https://developer.apple.com/technologies/mac/cocoa.html. Last accessed 25-January-2015. Cited on p. 29.

[Appb] ———, *OS X Yosemite for Developers - Apple Developer*, https://developer.apple.com/osx/. Last accessed at 25-January-2015. Cited on p. 29.

[Aut] Unknown Author, *Google Trends - Web Search interest - Worldwide, 2004 - present*, http://www.google.com/trends/explore?hl=en-US#q=Gmail,+Yahoo+Mail,+Hotmail,+AOL+Mail,+Outlook&cmpt=q, Last accessed 23-March-2015. Cited on p. 50.

[BDHS03] Victoria Bellotti, Nicolas Ducheneaut, Mark Howard, and Ian Smith, *Taking Email to Task: The Design and Evaluation of a Task Management Centered Email Tool*, Proceedings of the SIGCHI conference on Human factors in computing systems (CHI'03) (2003), no. 5, 345–352. Cited on p. 18.

[Bir33] George David Birkhoff, *Aesthetic measure*, Cambridge, Mass., 1933. Cited on p. 34.

[CMS99] Stuart K Card, Jock D Mackinlay, and Ben Shneiderman, *Readings in information visualization: using vision to think*, Morgan Kaufmann, 1999. Cited on pp. 6 and 13.

[Cos11] Emilia Dias Costa, *A Visualização de Informação como um método e um processo próprios do pensamento em Design*, Ph.D. thesis, University of Porto, 2011. Cited on pp. 5, 8, and 12.

[DB01] Nicolas Ducheneaut and Victoria Bellotti, *E-mail as habitat: an exploration of embedded personal information management*, Interactions **8** (2001), 30–38. Cited on pp. 2, 19, 20, and 21.

[DKFK05] Laura a. Dabbish, Robert E. Kraut, Susan Fussell, and Sara Kiesler, *Understanding Email Use: Predicting Action on a Message*, Proceedings of the 2005 Conference on Human Factors in Computing Systems (CHI) (2005), 691–700. Cited on p. 24.

[DLK06] Mark Dredze, Tessa Lau, and Nicholas Kushmerick, *Automatically classifying emails into activities*, 2006. Cited on p. 24.

[DW05] Nicolas Ducheneaut and Leon A Watts, *Opus : University of Bath Online Publication Store In search of coherence : A review of email research*, 11–48. Cited on pp. 20 and 23.

[Ehr90] Von Ehrenfels, *Gestalt Qualitiies*, 1890. Cited on p. 10.

[Ell15] EllisLab Inc, *Date Variable Formatting*, 2015, https://ellislab.com/expressionengine/user-guide/templates/date_variable_formatting.html, Last accessed 8-June-2015. Cited on p. 55.

[FBGS06] Danyel Fisher, a. J. Brush, Eric Gleave, and Marc a. Smith, *Revisiting Whittaker & Sidner's "email overload" ten years later*, Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work - CSCW '06 (2006), 309. Cited on pp. 18, 22, and 23.

[FR91] Tmj Fruchterman and Em Reingold, *Graph drawing by force directed placement*, Software: Practice and Experience **21** (1991), no. NOVEMBER, 1129–1164. Cited on pp. 40 and 60.

[FWS+12] Jean-daniel Fekete, Jarke Van Wijk, John Stasko, Chris North, Jean-daniel Fekete, Jarke Van Wijk, John Stasko, Chris North, and The Value, *The Value of Information Visualization*. Cited on pp. ix, 8, 9, and 11.

[Gui11] Nicolas Guillaume, *Next Generation Mail: Toward a Personal Social CRM*, 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology **3** (2011), 389–391. Cited on pp. 2 and 24.

[Han98] Robert a Hanneman, *Introduction to Social Network Methods*, Network **2** (1998), no. 3, 292–1990. Cited on p. 55.

[Hea96] Christopher G Healey, *Choosing effective colours for data visualization*, Visualization'96. Proceedings., IEEE, 1996, pp. 263–270. Cited on p. 9.

[HMM00] I. Herman, G. Melancon, and M.S. Marshall, *Graph visualization and navigation in information visualization: A\nsurvey*, IEEE Transactions on Visualization and Computer Graphics **6** (2000), 1–21. Cited on p. 15.

[Inc] Apple Inc, *Cocoa Drawing Guide*, https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/CocoaDrawingGuide/DrawingEnviron/DrawingEnviron.html#//apple_ref/doc/uid/TP40003290-CH202-BBCJDGHJ, Last Accessed 3-June-2015. Cited on p. 30.

[JDW03] By Thomas W Jackson, Ray Dawson, and Darren Wilson, *Understanding email interaction increases organizational productivity*, no. 8. Cited on p. 2.

[Kei02] Daniel a. Keim, *Information visualization and visual data mining*, IEEE Transactions on Visualization and Computer Graphics **8** (2002), no. 1, 1–8. Cited on p. 6.

[KK03] Bernard J Kerr and Bernard Kerr, *IBM Research Report THREAD ARCS : An Email Thread Visualization*. Cited on p. 20.

[Kof35] Kurt Koffa, *Principles of Gestalt Theory*, Routledge & Kegan Paul Ltd., London, 1935. Cited on p. 10.

[KWS04] Bernard Kerr, Eric Wilcox, and Rogers Street, *IBM Research Report Designing Remail : Reinventing the Email Client Through Innovation and Integration Designing Remail*. Cited on pp. ix, 20, and 22.

[Lit] Litmus, *Email Client Market Share*, https://emailclientmarketshare.com/, Last accessed 23-March-2015. Cited on p. 50.

[Moo] Pau B. Moody, *Reinventing Email*, Tech. report, IMB Research, Cambridge, USA. Cited on p. 20.

[Nie12] Jakob Nielsen, *How Many Test Users in a Usability Study?*, 2012, http://www.nngroup.com/articles/how-many-test-users/. Last accessed 8-June-2015. Cited on p. 62.

[NK93] Jakob Nielsen and Morgan Kaufmann, *Usability Engineering*, 340. Cited on pp. 61, 62, 63, and 64.

[RGM+] Steven L Rohall, Daniel Gruen, Paul Moody, Seymour Kellerman, and One Rogers Street, *Email Visualizations to Aid Communications*, IBM Research, 2–5. Cited on pp. ix, 20, and 22.

[Rub08] Chisnell Rubin, Jeff, Dana, *Handbook of Usability Testing, second edition*, 2008. Cited on p. 63.

[Sam04] Maryam Samiei, *EzMail: Using Information Visualization Techniques to Help Manage Email*, no. April. Cited on pp. 20, 25, and 35.

[She01] Nathan Shedroff, *An overview of understanding*, Information anxiety **2** (2001), 27–29. Cited on p. 12.

[Shn] Ben Shneiderman, *The eyes have it: A task by data type taxonomy for information visualizations*. Cited on p. 13.

[SM06] S. Soroczak and D.W. McDonald, *Collaborating Over Project Schedules*, Supporting the Social Side of Large Scale Software Development (2006), 43. Cited on p. 19.

[TGM83] Edward R Tufte and PR Graves-Morris, *The visual display of quantitative information*, vol. 2, Graphics press Cheshire, CT, 1983. Cited on p. 8.

[Tre85] Anne Treisman, *Preattentive processing in vision*, Comput. Vision Graph. Image Process. **31** (1985), no. 2, 156–177. Cited on p. 9.

[Unk] Unknown, *Running a Usability Test*. Cited on p. 63.

[Unk14] ———, *WebKit Framework Reference*, 2014. Cited on p. 30.

[Unk15a] ———, *Recruiting Usability Test Participants*, 2015, http://www.usability.gov/how-to-and-tools/methods/recruiting-usability-test-participants.html, Last accessed 8-June-2015. Cited on p. 64.

[Unk15b] ———, *Usability Testing*, 2015, http://www.usability.gov/how-to-and-tools/methods/usability-testing.html, Last accessed 8-June-2015. Cited on p. 61.

[VGD] Fernanda B Viégas, Scott Golder, and Judith Donath, *Visualizing Email Content : Portraying Relationships from Conversational Histories*. Cited on p. 20.

[War04] Colin Ware, *Information Visualization: Perception for Design*, Morgan Kaufmann Publishers Inc., San Francisco, CA, 2004. Cited on pp. 9, 10, 11, and 13.

[WS96] Steve Whittaker and Candace Sidner, *Email overload: exploring personal information management of email*, Proceedings of the SIGCHI conference on Human . . . (1996), 276–283. Cited on pp. 18, 20, and 22.

Appendices

Appendix A

Tree & Timeline Evaluation Script

Email Tree & Timeline Visualization

Pedro Daniel Cardoso Santos
ei10021@fe.up.pt

20th May 2015

1 Goal

The main goal of this exercise is to evaluate the usefulness, pros and cons of the newly developed tools to help visualize an email thread. These visualizations are a timeline and a tree graph. The purpose of the tree graph is to help understanding the relations between messages and the timeline to give a temporal overview of a thread.

Please note that this product isn't finished yet so this is not a final version but rather a first proof of concept to validate the ideas here developed.

2 Exercise

Next are going to be described the two exercises that will be used to evaluate this prototypes with your participation. A general consideration is that this application follows the principle of Apple Mail where the more recent emails are at the top of the list and are ordered in a descending way.

2.1 Part 1

Please open the application and click the button "View new" to view the thread that's going to be analyzed here **with** the new visualization prototypes. A window will open with the title being **Sprint Task Assignment** and has the content of the thread with subject mentioned before.

At the left side of the thread there is a view with some nodes representing an email and, when clicked, it's highlighted along with its replies (if available) and the parent (email to which the selected one was a reply, if available once again). At the right is displayed the list with all the emails of the thread. At the bottom is the timeline.

There are a few questions that can be answered in a form that can be found here: `http://goo.gl/forms/eb8upwGTk9`.

2.2 Part 2

Please open the application and click the button "View old" to view the thread that's going to be analyzed here **without** with the new visualization prototypes. A window will open with the title being **Sprint Task Assignment** and has the content of the thread with subject mentioned before.

There are a few questions that can be answered in a form that can be found here: `http://goo.gl/forms/eb8upwGTk9`.

3 Conclusion

Thank you for your collaboration! If you have any question, please feel free to contact me.

Appendix B

Tree & Timeline Questions Form

Edit this form

94 TREE & TIMELINE QUESTIONS FORM

Thread Tree & Timeline Evaluation

Form with questions related to the evaluation of thread's tree and timeline visualizations

* Required

Are you answering to this questionnaire using the "new" or "old" visualisation?

- New
- Old

Identify the most intense periods of discussion *

Identify the most important message (the one which has most replies) *

Sender, date and time

Identify the reply pattern *

- Reply to the last
- Reply to a specific email
- Other:

Identify every user first reply to task assignment message *

Ex: User 1 - Yes; User 2: yes but ...

How many messages belong to the thread?

Write just the number. Ex. 17

95

How easy was it to complete these tasks? *

- Very easy
- Easy
- Normal
- Hard
- Very hard

Relevant observations / Suggestions

Submit

Never submit passwords through Google Forms.

Appendix C

Tree & Timeline Evaluation Results

Timestamp	Identify the most intense periods of discussion	Identify the most important message (the one which has most replies)	Identify the reply pattern	Identify every user first reply to task assignment message	How easy was it to complete these tasks?	Relevant observations / Suggestions	Are you answering to this questionnaire using the "new" or "old" visualisation?	How many messages belong to the thread?	Time taken (min)	% correct answers
5/19/2015 23:58:02	Friday 15th	Too hard to identify	Reply to a specific email	Mark - okay Jonh - okay Peter - okay	Hard	It's very confusing and disorganized	Old	16	7	
5/19/2015 23:59:00	wednesday 13th	Joseph, 11th May at 10h01 min	Reply to a specific email	Taylor - okay Ken - okay Jacob - okay	Easy	It's simpler this way.	New	18	5	
5/20/2015 15:11:03	Wednesday 13	Joseph, Monday 11 May, 10h 01	Reply to a specific email	User 1 - Yes User 2 - Yes User 3 - Yes, half way there	Easy	Scroll not working well. Had to manually count the number of messages. Zoom control for timeline is weird. Message with most number of replies is not that easy to identify. Tree view is not that helpful as it is at the moment (identation of messages might be more useful).	New	18	4:45	100
5/20/2015 15:19:54	Friday 15	Bernard, Friday 15, 10h01	Reply to a specific email	User 1 - will be on vacation, so no User 2 - yes User 3 - yes	Hard	This interface has worse usability than most normal e-mail clients.	Old	16	2:50	100
5/20/2015 16:38:47	13 maio, à volta das 8 da manhã	Joseph, monday 11 may 2015 as 10h01m	Reply to a specific email	Taylor - yes ken - yes jacob - yes	Normal	A tree é bastante útil para agrupar o "flow" de emails que nao estao organizados temporariamente, mas revela-se pouco util para threads pequenas. Adoro a timeline porque permite ver facilmente a ordem cronológica dos emails. Como sugestao, um "merge" da tree com a vista em timeline seria ideal.	New	18	12:14	100
5/20/2015 16:45:55	11 may 2015	Bernard 11 may 2015 10h01 (influenciado pelo questionario anterior, new, dado que aqui requer muito mais esforço para interpretar)	Reply to a specific email	Mark - no john - ok peter - ok jake - ok	Very hard		Old	16	6:20	70
5/20/2015 17:35:57	11/05 @8am 13/05 morning 15/05 midnight	the original message	Reply to a specific email	taylor - yes Ken - yes Jacob - yes, with delay	Normal	Identifying the most important message wasn't that easy, everything else was. Would be useful if the tree visualisation showed hierarchy Would be useful to be able to invert the order - read from oldest to newest from top to bottom	New	18	6:55	100
5/20/2015 17:48:52	11/05 @10am 15/05 @1pm 15/05 evening (only 2 emails)	Original email Bernard, 11/05 @10	Reply to the last	Mark - can't, vacation John - ok Peter - ok Jake - ok	Hard	Missing subject makes identifying replies hard	Old	16	6:33	100
5/20/2015 18:29:57	monday 11 May 2015 @ 9:30 13 May 15 May	Joseph, 11 May 2015 10:01	Reply to a specific email	Taylor - Ok Ken - Ok Joseph - Question to jacob Jacob - 50% there	Normal		New	18	12:46	100
5/20/2015 18:35:57	11 May 2015 15 May 2015	Hard to ld. but I think it's the first one: Bernard 11 May 2015 10:01	Reply to a specific email	Mark: No John : OK Peter: OK Bernard: Question jake: Ok	Normal		Old	16	7:36	100
5/21/2015 15:33:20	Friday 13 May	Joseph, 11 May 2015 10:01	Reply to a specific email	Taylor - Ok Ken - Ok Joseph - Question to jacob Jacob - 50% there	Easy	Sync left hand side scroll with right hand side; Adapat box sizes if window size is changed; expand specific email on click; Expand whole cluster with just one click.	New	18	5:10	100
5/21/2015 15:23:20	Friday 15 May	Bernard, 11 May 2015	Reply to a specific email	Mark - No John - Yes Peter - Yes Jake - Yes	Hard		Old	16	4:47	100
								Avg Time New	8:22	
								Avg Time Old	5:37	

Appendix D

Chat Evaluation Script

Email Chat Visualization

Pedro Daniel Cardoso Santos
ei10021@fe.up.pt

20th May 2015

1 Goal

The main goal of this exercise is to evaluate the usefulness, pros and cons of an alternative visualization for an email conversational thread similar to a chat. Its purpose is to offer a quicker and simpler visualization over a thread when its profile looks like a chat - short messages with timespan between them being short as well and the relationship between messages typically is "reply to the last one" instead of "reply to a specific email to refer its content".

Please note that this product isn't finished yet so this is not a final version but rather a first proof of concept to validate the ideas here developed.

2 Exercise

Here is going to be described an exercise to evaluate the chat visualization for an email thread. This visualization should be applied to threads typically composed by short emails (were considered threads that have emails with average length equal or less to 140 characters).

Its scope is a discussion between a group of friends so that they can decide where they are going to have dinner tonight.

To view this conversation, please click the button "View chat" and a window with the content will be opened.

Please answer to the questions in the following link: `http://goo.gl/forms/5ARzl8avLv`.

3 Conclusion

Thank you for your collaboration! If you have any question, please feel free to contact me.

Appendix E

Chat Evaluation Questions Form

Edit this form

102 CHAT EVALUATION QUESTIONS FORM

Chat Evaluation

* Required

Identify the amount of intervenients *

Identify the amount of messages that you sent *

Is it easier to distinguish your messages from other intervenient's messages with a chat visualisation rather than a standard visualisation? *

- Yes
- No

Was it easier to send messages this way (in this context)? *

- Yes
- No

Do you find this chat approach to visualize/interact with an email thread useful? *

- Yes
- No

Why? *

Relevant observations/suggestions

Submit

Never submit passwords through Google Forms.

103

Appendix F

Chat Evaluation Results

Timestamp	Identify the amount of intervenients	Identify the amount of messages that you sent	Was it easier to send messages this way (in this context)?	Do you find this chat approach to visualize/interact with an email thread useful?	Why?	Relevant observations/suggestions	Is it easier to distinguish your messages from other intervenient's messages with a chat visualisation rather than a standard visualisation?	Time taken (min)	% correct answers
5/20/2015 0:10:21	3	4	Yes	Yes	Because, with the chat it's always easier to distinguish the messages i sent and the messages that other people sent.		Yes	4	
5/20/2015 15:30:55	3	4	Yes	Yes	For quick conversations allows for a faster, no-frills sending of replies. Easy to glance through content, and identify participants.		Yes	2:50	100
5/20/2015 16:55:52	3	4	Yes	Yes	Translates email messages exchange into a simple conversation into an easily digestible environment. Condensates necessary information into a small space, allowing to visualize all of the information in the first look		Yes	3:20	100
5/20/2015 17:53:31	3	4	Yes	Yes	For short conversations where all participants are online at the same time, this method is more efficient due to less overhead. Then again, in these situations we tend to use actual chat applications.	Missing time	Yes	3:02	100
5/20/2015 18:44:40	3	4	Yes	Yes	It streamlines the interface in a way that will facilitate the interactions between the users.	Signature support for the "new" interface and hide it in the "Chat"	Yes	3:19	100
5/21/2015 15:23:28	3	4	Yes	Yes	If the messages are short, the chat system is easier perception-wise than an e-mail thread. It's more compact and the information gets through more easily	Maybe give a hint UI-wise if a (relatively) big period of time passed (a line or something like that). That line could have the day or the hour centered horizontally (like other chat systems like FB or Skype). Adapt message box size to the size of the message (less width if message is shorter - again like FB or other. MINOR DETAIL)	Yes	1:45	100
							total seconds	856	171.2
							Average time	14:16	2:51

Appendix G

Thread Recreation Evaluation Script

Thread recreaction evaluation

Pedro Daniel Cardoso Santos
ei10021@fe.up.pt

20th May 2015

1 Goal

The main goal of this exercise is to evaluate the usefulness, pros and cons of a new feature for email clients that is to reconstruct a thread based on its inline content when its historic is not previously saved - when an email with a discussion is forwarded to a new intervenient so that he can acknowledge or when the user deletes messages of the thread, losing that way context of the thread's content, for example.

Please note that this product isn't finished yet so this is not a final version but rather a first proof of concept to validate the ideas here developed.

2 Exercise

Here is going to be described an exercise to evaluate the thread recreation feature.

This exercise has two parts: the first is to perform the tasks with the current technology and then perform a similar set of tasks using the new feature aforementioned.

2.1 Old way

First of all please login into a Gmail account with the following credentials: **markmscfeup@gmail.com** / **myhardpassword15** and open the email in the inbox. Its scope is a discussion between two project managers of the same company discussing which tasks should be assigned to each time that is then forwarded to the CEO so he can acknowledge it.

Then, please answer to the questions in the following link: `http://goo.gl/forms/p6caOq1sDc`.

2.2 Reconstructed thread

Please open the application and click the button "View reconstructed" to view the reconstructed thread. The thread is the same but it was reconstructed to

behave like a regular thread.

Then, please answer to the questions in the following link: `http://goo.gl/forms/p6caOq1sDc`.

3 Conclusion

Thank you for your collaboration! If you have any question, please feel free to contact me.

Appendix H

Thread Recreation Questions Form

Edit this form

112 THREAD RECREATION QUESTIONS FORM

Thread Recreation Evaluation

* Required

Are you answering to this questionnaire using the web version or the reconstructed thread? *

- ○ Web version
- ○ Reconstructed thread

Identify the intervenients *

Identify which were the decisions made *

Identify who mentioned the problem of implementing the spell checker *

Identify if it was a long discussion (regarding time, not number of messages) *

How easy was it to perform these tasks?

- ○ Very Easy
- ○ Easy
- ○ Normal
- ○ Hard
- ○ Very Hard

Relevant observations/suggestions

113

Submit

Never submit passwords through Google Forms.

Appendix I

Thread Recreation Evaluation Results

Timestamp	Identify the intervenients	Identify which were the decisions made	Identify who mentioned the problem of implementing the spell checker	Identify if it was a long discussion (regarding time, not number of messages)	Relevant observations/suggestions	Are you answering to this questionnaire using the web version or the reconstructed thread?	How easy was it to perform these tasks?	Time Taken (min)	% correct answers
5/20/2015 0:26:54	Ben and Phill	It's very hard to identify because the conversation looks really confusing this way.	Phill	Short		Web version	Normal	10	
5/20/2015 0:31:58	Ben and Phill	They were talking about spliting tasks and at the end, Ben was responsible about one thing and Phill was responsible about another thing.	Phill	Short		Reconstructed thread	Easy	4	
5/20/2015 15:37:21	Ben and Philip	Go with Speech-recognition and photo-editing for tasks for the sprint	Philip	no		Web version	Normal	3:39	100
5/20/2015 15:41:49	Philip and Ben	Go for speech-recognition and photo-editing	Philip	no	The access to different views really helps when going for an overview analysis like this. The chat view, however, ends up being much more generally helpful than the regular view, which seems to be caused more by a flaw of design of it than by the natural advantages of the chat view.	Reconstructed thread	Very Easy	1:35	100
5/20/2015 17:03:17	Ben e Philip	philip - photo editing instagram-like ben - speech recognition	philip	short		Web version	Hard	5:01	100
5/20/2015 17:06:31	philip e ben	ben - speech recognition philipp - photo editing	Philip	Short		Reconstructed thread	Very Easy	1:30	100
5/20/2015 18:03:06	Ben Philip	Send the thread to the ceo. Thread mentions in-site photo editing and new spellchecker, both for the next 4 sprints	Phil	Short - 3 minutes total	Two different styles of timestamps makes identifying times harder than it should Oldest at bottom is annoying. Oldest at top is much easier to read in this context.	Web version	Normal	4:06	88
5/20/2015 18:09:13	Ben Phil	Same as before, same thread	Phil	Short	Again, oldest at bottom is not as comfortable as could be Timeline stretched to width of conversation makes it seem that the conversation was long. Maybe use colors to color code days/hours/minutes	Reconstructed thread	Easy	4:03	100
5/20/2015 18:54:54	Phill Ben	speech recognition photo edit	Ben	2min	With this style it's much harder to read and identify certain things.	Web version	Hard	5:16	100
5/20/2015 18:58:44	Ben Phill	to develop speech recognition and photo ed.	Phil	Short	To preform this task there was no need for the timeline and side thing :D Because only the content was important.	Reconstructed thread	Normal	2:41	100
5/21/2015 15:32:59	Phillip Ben	speech recognition to perform actions (navigate, comment, like, etc) and photo editing Instagram-like (crop, filters, hand-drawings, etc)	Ben	no		Web version	Normal	4:53	75

Timestamp	Identify the intervenients	Identify which were the decisions made	Identify who mentioned the problem of implementing the spell checker	Identify if it was a long discussion (regarding time, not number of messages)	Relevant observations/suggestions	Are you answering to this questionnaire using the web version or the reconstructed thread?	How easy was it to perform these tasks?	Time Taken (min)	% correct answers
5/21/2015 16:01:42	Ben and Philip	Besides Phil had a suggestion, ben says not to take decisions and send it to Mark	Philip	No, 2 min	I would only know for sure if I would see both versions, but maybe it's more intuitive if most recent e-mails are presented at the bottom of the box, and not at the top. If you wanted to see previous e-mails, you would scroll up."	Reconstructed thread	Easy	6:21	100
							Avg Time Web	4:35	
							Avg Time Reconstructed	3:14	

Appendix J

Sociogram Evaluation Script

Sociogram applied to email

Pedro Daniel Cardoso Santos
ei10021@fe.up.pt

1st June 2015

1 Goal

The main goal of this exercise is to evaluate the usefulness, pros and cons of the newly developed tools to help extracting information from an email thread. In this case, the tool that's going to be evaluated here it's a sociogram - a network graph that displays social relations - applied to an email thread.

Please note that this product isn't finished yet so this is not a final version but rather a first proof of concept to validate the ideas here developed.

2 Exercise

Before describing the exercise itself, there are going to be presented some notions and concepts about the prototype. It's a network graph that shows relations among people in a thread. Each node represents an individual (size is related to the number of messages that the user sent to the thread) and each link represents the number of direct replies between the two users that are at the ends of the link. When a user replies to himself, these messages are not represented by links.

2.1 Part 1

Please open the application and click the button "View old" to view the thread that's going to be analyzed. A window will open with the title being **Equity sharing?** and has the content of the thread with subject mentioned before, without the sociogram.

There are a few questions that can be answered in a form that can be found here: `http://goo.gl/forms/TCvWFP68Bm`.

2.2 Part 2

Please close the previous window the application and click the button "View new" to view the thread that's going to be analyzed. A window will open with

the title being **Equity sharing?** and has the content of the thread with subject mentioned before, with the sociogram.

There are a few questions that can be answered in a form that can be found here: `http://goo.gl/forms/TCvWFP68Bm`.

3 Conclusion

Thank you for your collaboration! If you have any question, please feel free to contact me.

Appendix K

Sociogram Evaluation Questions Form

Edit this form

124 SOCIOGRAM EVALUATION QUESTIONS FORM

Sociogram Evaluation

* Required

Which version are you using to answer to this questionnaire? *

- Normal Version
- Sociogram

Identify the number of intervenients *

Identify who were the bigger contributors for the discussion *

Identify, among them, the participant who sent the bigger number of messages to the thread *

Did Charles interact with Mike ?

- Yes
- No

Was there anyone who was just “observing” the conversation? *

sending 1 message or less

- Yes
- No

If yes, then who?

Do you find this feature useful ?

- Yes
- No

Do you think this feature should be a complement to standard thread visualization or an alternative?

- Complement
- Alternative

How easy was it to perform these tasks? *

- Very Easy
- Easy
- Normal
- Hard
- Very Hard

Relevant observations / Suggestions

Submit

Never submit passwords through Google Forms.

Appendix L

Sociogram Evaluation Results

Timestamp	Which version are you using to answer to this questionnaire?	Identify the number of intervenients	Identify who were the bigger contributors for the discussion	Identify, among them, the participant who sent the bigger number of messages to the thread	Did Charles interact with Mike ?	Was there anyone who was just "observing" the conversation?	If yes, then who?	Do you find this feature useful ?	Do you think this feature should be a complement to standard thread visualization or an alternative?	How easy was it to perform these tasks?	Relevant observations / Suggestions	Time
6/1/2015 11:00:58	Normal Version	5	Charles and Jake	Charles	No	No				Hard	Hard task, specially if considering that we were only evaluating a thread with about 20 messages.	05:37
6/1/2015 11:06:45	Sociogram	4	Mike, Ken, Ben	Mike	Yes	No		Yes	Alternative	Easy	There is no way to determine if if Charles replied to Mike.	02:29
6/1/2015 12:03:16	Normal Version	4	Charles, Jake	Charles	Yes	Yes	Mike			Normal		03:40
6/1/2015 12:06:39	Sociogram	4	Mike	Mike	Yes	No		Yes	Alternative	Very Easy	Show more information in the connections (like the amount of emails that one person sent to the other)	01:31
6/1/2015 12:46:12	Normal Version	5	Charles, Peter and Jake	Charles	Yes	Yes	Mike			Easy		05:47
6/1/2015 12:54:50	Sociogram	4	Mike, Ken and Ben	Mike	Yes	No		Yes	Complement	Very Easy	Using different colors, distinguish sent from received messages	03:22
6/1/2015 14:53:09	Normal Version	5	charles	charles	Yes	Yes	Mike			Hard		04:30
6/1/2015 15:00:59	Sociogram	4	mike, ben, ken	mike	Yes	No		Yes	Complement	Easy	Feature should be used as a complement when there aren't much users. When there's a bigger number of users (10/15), should be an alternative.	03:38
6/1/2015 15:24:18	Normal Version	5	Charles	Charles	Yes	No				Very Easy		03:17
6/1/2015 15:31:05	Sociogram	4	Mike	Mike	Yes	No		Yes	Complement	Easy		02:40
											Avg Time Normal	04:34
											Avg Time Sociogram	02:44

Appendix M

Thread to validate reconstruction

This one was sent from iCloud mail
On Mar 23, 2015, at 05:48 PM, Pedro Santos <pedrodanielcsantos@gmail.com> wrote:

This one was sent from thunderbird
On 23/03/15 15:25, Pedro Santos wrote:
This one is sent from Yahoo mail.

On Monday, March 23, 2015 12:18 PM, Pedro Santos <ei10021@fe.up.pt> wrote:

This one is sent from webmail

Em 23.03.2015 12:17, pedrodanielcsantos@gmail.com escreveu:

This one is sent from Outlook

Sent by Outlook [7] for Android

On Mon, Mar 23, 2015 at 5:15 AM -0700, "Pedro Santos" <pedrodanielcsantos@gmail.com [8]> wrote:

This one is sent from mailbox

>> Sent from Mailbox [5]
>>
On Mon, Mar 23, 2015 at 12:14 PM, Pedro Santos <pedrodanielcsantos@gmail.com [6]> wrote:
>>
>>> This one is sent from Google Inbox
>>>
On Mon, Mar 23, 2015 at 3:35 PM Pedro Santos <pedrodanielcsantos@gmail.com> wrote:

>>>
>>>> This one is sent from apple mail.
>>>>
On 23 Mar 2015, at 12:12, Pedro Santos <pedrodanielcsantos@gmail.com [1]> wrote:
>>>>>
>>>>> This one is sent from Gmail online.
>>>>>
>>>>> Sent with MailTrack [2]

>>>>>
>>>>> Pedro Santos
>>>>>
>>>>> SKYPE: pedrodanielcsantos
>>>>> LINKED IN: http://www.linkedin.com/pub/pedro-santos/85/b83/588
>>>>> [3]
>>>>>
2015-03-23 12:12 GMT+00:00 Pedro Santos <pedrodanielcsantos@gmail.com:
>>>>>
>>>>>> Just to start
>
>
> Links:
> ------
> [1] mailto:pedrodanielcsantos@gmail.com
> [2]
>
>
https://mailtrack.io/install?source=signature&lang=en&referral=pedrodanielcsantos@gmail.com
&idSignature=22

> [3] http://www.linkedin.com/pub/pedro-santos/85/b83/588
> [4] mailto:pedrodanielcsantos@gmail.com
> [5] https://www.dropbox.com/mailbox
> [6] mailto:pedrodanielcsantos@gmail.com
> [7] http://taps.io/outlookmobile
> [8] mailto:pedrodanielcsantos@gmail.com

--
Pedro Santos

Appendix N

Threads used for evaluations

N.1 Messages (Graph & Timeline 1)

1 Bernard (-) Hello guys!
So, I planned this next sprint and according to the product's priorities and your availability I distributed the tasks in the following way:
@John, you take the login and authentication mechanism;
@Jake, you take care of the connection to the Facebook, Linkedin and Google+ APIs;
@Mark, you take care of the display of Jake's work;
@Peter finally you're responsible for the user's dashboard;
Is everyone ok with this?

2 Mark (1) Hey Bernard,
I think you forgot that I'll be on vacation this next week and therefore this sprint. So I won't be able to execute the assigned task.

3 John (1)
Ok for me! Let's work!

4 Peter (1)
I'm also ok with that.

5 Bernard (2) Damn, i forgot! Jake, can you please take care of Mark's tasks ?

6 Jake (1)
It works for me!

7 Jake (5)
Sorry, i didn't see this mail. Ok, it'll be hard, but i'll try to

8 Bernard (1)
Hey guys, sprint ends today! What's the status of your tasks ? Everythinh's done and ready? :)

9 John (8)

Everything's perfect! :)

10 Jake (8)

Connection to APIs fully done, visualization of the information still needs some tweaks. I would say 80

11 Peter (8)

100% done here! :)

12 Peter (10)

Jake, need some help ? I'm free, and I can help you do it untill the sprint review meeting!

13 Jake (12)

I'd really appreciate that!

14 Jake (13)

Thanks to Peter's help, all of my tasks are done!

15 Bernard (8)

Thank you all so much for the effort! Perfect sprint! :)

16 Mark (15)

Wow guys, looks like you almost don't need me! :p

N.2 Messages (Graph & Timeline 2)

1 Joseph

Hey guys!

As you all know we have to get things right this sprint as we are behind schedule. I'll be programming as well this sprint. So, for the tasks assigned to each one:
@Taylor you'll be responsible for the reservations section.
@Ken you'll be responsible for the payment mechanism (integrate with paypal).
@Jacob you're going to review the infrastructure and get it ready for deployment.
@Myself I'll be handling the frontend of Taylor and Ken's features.

Everyone's ok with this? Let's go guys, We're amost there!

2 Taylor (1)

Ok by me! Let's go!

3 Ken (1)

Sure thing! Already working on it!

4 Joseph (1)

Jacob, how about you man? It's already been 2 days since the task assignment and you haven't said anything! Please let me know!

5 Jacob (1)

Sorry for the delay boss, i forgot to answer! Of course I'm ok with that. Half way there!

6 Joseph (1)

Ok, we're on the right path now then! Thank you all.

7 Joseph (6)

Any updates on this ?

8 Taylor (7)

Done. Is there anything else I can do?

9 Ken (7)

Working on it. It's taking more time than I expected. But it'll be done in time.

10 Jacob (7)

Planning is done, I'm waiting for the material to get it up and running.

11 Joseph (6)

I forgot to tell you mine. My bad. It's going pretty well by me!

12 Joseph (8)

Taylor, way to go! :) Can you please take care of viewing and managing your history?

13 Taylor (12)

On it!

14 Joseph (11)

So, sprint ends today. What's the status on your tasks ? Mine were successfully completed! :)

15 Ken (14)

100% done here!

16 Taylor (14)

Both tasks done! Successful sprint is successful.

17 Jacob (14)

Everything's gonna be ready for today's sprint. Packages got later than I was expecting. But it's under control, we can consider it done!

18 Joseph (14)

I did it all as well. Congratulations to us, we are on time again! Great effort guys!

N.3 Messages (Thread Recreation)

Philip

Hey Ben! As you know we have to plan the next set of features and divide them between our teams, as Mark asked.

Have you thought of anything?

Ben

Hey Phil,

Yeah, I've seen it. I'm think on going for the speech recognition and a spell checker to end with that horrible spelling that we've seen :(. I think that's a good milestone for this next 4 sprints. 2 to develop, 1 to go beta and another one to fix bugs. What do you think?

Philip

Hey Ben,

Seems reasonable! But remember when we talked about that in the last meeting and Mark said that he really wanted to have the photo editing in-site implemented as soon as possible? I think that's more urgent than the spell checker (Although not more important :p).

Ben

Hey again,

Oh yeah, I totally forgot that. But that's a bit more complicated to do. I don't know if we can get it done in that timespan, as Joseph is in the hospital and we're short on staff. Do you think you can do it ?

Philip I think that's achievable. We're in full power :) So you think it's ok to propose this set of features: speech recognition to perform actions (navigate, comment, like, etc) and photo editing Instagram-like (crop, filters, hand-drawings, etc)?

You could get the first one and we would get the second one, for the next 4 sprints.

Ben Hey Phil,

I believe that's enough. And if you're ok with that, for me it's perfect. Do we have to validate this with Mark?

Philip

No, just send this to him. You take care of it or do you want me to?

I'm in a bit of a hurry now, so if you could do it, it would be perfect.

Ben

Perfect then Phil.

Don't worry, I'll send this to him.

See you at the meeting later!

Ben

Hey Mark!

Here's the discussion I had with Philip about the next features to implement and which team implements what. I hope you're ok with that! See you at the meeting later!

Best regards,

Ben

N.4 Messages (Chat)

Peter Hey guys! So, where's dinner tonight? And at what time?

Jake

I would say somewhere downtown? At around 8.

Andrew

I have a meeting untill 8.30. No can do!

Jake

Damn! So, 9?

Andrew

I think it works for me!

Jake

So, Peter?

Peter

9 is perfect! And where do we go?

Andrew

I really need to save money! Can we go to somewhere cheap?

Jake

Don't be silly Andrew! It's only for this time. And I'm in the mood for a good Francesinha :)

Peter

Damn you Jake! Now I'm in the mood as well. Andrew is settled then? Francesinha at 9?

Andrew

Ok, you convinced me! Done! At that that place on the main square?

Peter

Do you think we can get a table?

Jake

I know the guy! I can call and reserve a table.

Andrew

Let us know when you get it

Peter

What he said!

Jake

It's on! Table reserved for 3 at 9 :)

Peter

Ok guys, see you there ;)

Andrew

Prepare your appetite! See you there!

N.5 Messages (Sociogram 1 - Without sociogram)

Peter

Hey guys. As you know, we have to decide on the proposal to sell our stocks! 500k for 25

Andrew

I don't know man. It cust our valuation to one half.

Jake

For me it's on. We need the money more than anything now...

Charles

I agree with Andrew... We should ask for more money or less equity.

Mike

I also think that we should take more money than we are currently getting.

Peter

I forgot to tell you my opinion LOL. I think we should take it as is now man...

Andrew

So it's 2 to accept current offer and 3 to ask for more mone/less equity. Consensus?

Peter

@Charles, how much money do we have left?

Charles

Enough for 5 months.

Peter

If that's the case I think we should negotiate it a little further and try to get somethig more. 3-2, but majority says not to take it now :)

Charles

Glad you agree :) Only Jake's missing now.

Jake

You got a point man... But i don't know... Feels weird to throw 500k out the window.

Charles

You gotta believe us man. We're gonna get at least 750k!

Jake

Are you sure? With the same investor?

Charles

We have two of them in mind now. I've been dealing with the offers. And I think we're on the right path

Jake

Ok then. It's on, let's reject it for now and be millionaires! :)

Charles

Ok guys, to it's settled for now. We'll reject this one and then rock out the next investor :)

N.6 Messages (Sociogram 2 - With sociogram)

Ken

Guys, how are we on the investment round ? I haven't heard anything about this in a while or am I distracted?

Charles

SORRY!!! I forgot to update you guys on this. We have two offers on the table. 1 of them is 25k for 25% and the other is 50k with 10% royalties. Your opinion?

Mike

Hmmm... tricky... I think we will loose more money with the royalty thing. So I'd say the first one.

Charles

I'm with you Mikes!!! :)

Ken

meeeeeeeh. I like the royalty deal ... Ben???

Ben

I like it also (the royalty deal).

Mike

Are you sure? For each euro that we earn we have to give him 10 cents. There goes the profit!!! :(

Ken

Ok, you got me Mike! Let's go with the 25% then!

Mike

How about you Ben? We should achieve consensus...

Ben

I really like the royalty thing. Do you think it's that bad?

Mike

On the long run it will be disgraceful for us... And there's no turning back after. And the first guy really wants to help us improve and not just give the money.

Ben

Ok. 25k for 25% it is!

Mike

Hell yeah! :D It's on guys!

Mike

Money's already in! WOOP WOOP WOOP