VIBE: Visualização e Exploração de Blogs

Nelson Pinto Rocha Marques

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

Júri

Presidente: Professor Doutor Mário Rui Fonseca dos Santos Gomes
Orientador: Professor Doutor Manuel João Caneira Monteiro da Fonseca
Vogais: Professor Doutor Luís Manuel Pinto da Rocha Afonso Carriço

Novembro 2011
VIBE: Visualização e Exploração de Blogs

Nelson Pinto Rocha Marques

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

I would like to express my gratitude to everyone who contributed with their time and effort to help define and build the path that led to the writing of this dissertation. First and foremost, I would like to thank Professor Manuel João Fonseca, for his insight and invaluable guidance during each step of this work. His lectures contributed in no small part to my choice to proceed along this branch of Computer Science and the obtained results would not have been possible without his words of encouragement.

The effort of the users who gave their time to participate in the evaluation of our solution should also be remembered. A special thanks goes to Nelson Silva, Tiago Cardoso and Ricardo Dias for their feedback and expertise.

I would also like to thank all the colleagues that provided assistance throughout the degree, in particular Tiago Lourenço, for his support and friendship.

My final thanks go to my family, especially my parents, to whom I dedicate this dissertation. Everything I have attained until now I owe to their spirit of sacrifice and dedication.

Lisboa, November 14, 2011
Nelson Marques
Não sou nada. Nunca serei nada, não posso querer ser nada. À parte isso, tenho em mim todos os sonhos do mundo.

- Álvaro de Campos
Resumo

Os blogues são actualmente muito utilizados para publicar informação regularmente, tendo alcançado um lugar de destaque na transmissão de informação através da Internet. No entanto, a sua navegação é difícil e pouco usável, dada a sua rigidez estrutural, bem como a escassez de funções para ajudar o utilizador na procura de informação relacionada com os temas do seu interesse.

Embora tenham surgido algumas soluções para este problema, estas falham em alguns aspectos como a falta de uma visão geral sobre o conteúdo do blogue ou a dificuldade de acesso à última entrada. Outras soluções, apesar de colmatarem estas falhas, pecam por cortarem radicalmente com a estrutura à qual os utilizadores estão habituados.

Com este trabalho pretendemos proporcionar aos utilizadores uma solução para visualizar e explorar blogues individuais, focada na flexibilidade de navegação e em entradas de blog melhoradas, com o intuito de promover o reconhecimento do conteúdo através do suporte a miniaturas de imagens e marcação de entradas lidas.

A solução proposta dispõe de um mecanismo de visão geral e filtragem integrado, que tira partido da informação relevante associada a posts e comentários, de modo a ajustar os resultados visíveis às necessidades do utilizador, à medida que este explora o blogue.

Os resultados experimentais com utilizadores revelam que a nossa abordagem é adequada ao problema em questão. Em concreto, a nossa solução recebeu uma elevada classificação a nível de usabilidade, em comparação com duas soluções similares, na realização das mesmas tarefas, as quais foram também realizadas com menor dificuldade.
Abstract

Blogs are highly used nowadays to publish information on a regular basis and retain an important place as far as information transmission over the Internet is concerned. However, navigating them is difficult, mainly due to its rigid structure and scarce search support.

Even though there have been some developments dedicated to solving this problem, the present solutions fail in some aspects such as the lack of overview capability or difficult access to the last entry. Other solutions, although surpassing those flaws, do not present a familiar structure, leaving a gap between novel interfaces and the one used during the last years.

With this work we aim to bridge this gap by providing users with a solution to visualize and explore individual blogs, in an effective and efficient way, focused on flexibility of navigation and enhanced blog entries that promote the recognition of content with support for image thumbnails and read wear.

The proposed solution is equipped with an integrated overview and filtering mechanism that takes advantage of the most relevant metadata within posts and comments in order to adjust the visible results to the needs of the user, as he explores the blog.

Experimental usability tests with users demonstrate that our approach is an adequate solution to the previously described problem. In particular, our solution received higher usability scores as compared with two other similar tools used in the same tasks. Users also reported having less difficulty in completing relevant tasks when using our solution, while taking less time.
Palavras Chave
Keywords

Palavras Chave

Blogues
Exploração
Navegação
Visualização

Keywords

Blogs
Exploration
Navigation
Visualization
## Contents

1 Introduction ......................................................... 1  
   1.1 Problem Description ................................................ 1  
   1.2 Objectives ..................................................................... 2  
   1.3 Developed Solution ...................................................... 3  
   1.4 Contribution and Achieved Results ................................. 3  
   1.5 Dissertation Organization ............................................. 4  

2 Context and Related Work .............................................. 5  
   2.1 Context ........................................................................ 5  
      2.1.1 Blog Interconnection and Community ......................... 5  
      2.1.2 Subject Similarity .................................................... 7  
   2.2 Blog Exploration and Visualization .................................. 8  
      2.2.1 Usual Blog Format ................................................... 9  
      2.2.2 Blogger Views ........................................................ 11  
      2.2.3 iBlogVis .................................................................. 12  
      2.2.4 PostConnect ........................................................... 13  
      2.2.5 TRIB - Telescope for Responding comments for Internet Blogs .............. 15  
      2.2.6 The Focus-Metaphor approach .................................... 16  
      2.2.7 StarTree® Interface ................................................... 17  
      2.2.8 Taglines .................................................................. 18  
      2.2.9 Lifelines .................................................................... 19
2.2.10 VisGets .......................................................... 19
2.3 Aggregators ....................................................... 20
2.4 Exploration and Visualization Techniques ................. 23
   2.4.1 Faceted Exploration and Navigation ...................... 23
   2.4.2 Other Visualization Techniques ......................... 24
2.5 Comparative Table and Discussion ........................... 27
2.6 Summary .......................................................... 31

3 Blog Analysis and Interface Design ............................ 33
   3.1 Blog Analysis .................................................. 33
      3.1.1 Experiment ............................................... 33
      3.1.2 Results .................................................. 34
      3.1.3 Implications on the Solution ............................ 35
   3.2 Interface Design ............................................... 36
   3.3 Summary ........................................................ 42

4 Vibe - Visualizing and Exploring Blogs ....................... 43
   4.1 System Architecture and General Information Flow ....... 43
   4.2 Server ........................................................ 44
   4.3 Client User Interface .......................................... 46
      4.3.1 Posts and Comments .................................... 47
      4.3.2 Read Wear ............................................... 49
      4.3.3 Sidebar and Overview ................................... 49
      4.3.4 Navigation and Exploration ............................ 50
   4.4 Summary ........................................................ 52

5 Evaluation .......................................................... 53
   5.1 Preliminary Formative Evaluation with Users .............. 53
      5.1.1 Prototype ............................................... 53
List of Figures

2.1 Scenarios of the comment data set with the highest score range on the selected blog entry, using KeyGraph. ................................................................. 6

2.2 The thickness of the lines connecting blog entries in VizBlog [PQ07] denotes the level of similarity between them. ...................................................... 7

2.3 The galaxy metaphor within the In-Spire system presents individual blog posts as dots, clustered according to their thematic content. ......................... 8

2.4 Excerpt of the graph view example of eNulog about the movie *Harry Potter and The Goblet of Fire*. ................................................................. 9

2.5 One of the default Blogger templates. .................................................... 10

2.6 Six different views of the same blog using the five metaphors of Blogger Views: Flipcard - Recent (top left), Flipcard - Label (top center), Mosaic (top right), Sidebar (bottom left), Snapshot (bottom center) and Timeslide (bottom right). . 11

2.7 iBlogVis allows the visualization of blog entries along a timeline. ............ 12

2.8 The circular interface of PostConnect, including complementary information and functionality on the top right corner. ........................................... 14

2.9 TRIB uses the solar system metaphor to represent keywords around an article and comments around each keyword. ............................................... 15

2.10 The Focus-Metaphor prototype interface used for online testing. ............... 17

2.11 Accessing a blog entry using the StarTree Interface, in [Laqua 07]. .......... 18

2.12 Comparison between ”river” (on the left) and ”waterfall” (on the right) metaphors used in TagLines. ................................................................. 18

2.13 Coordinated time, location and tags VisGets, with related results. ............. 20

2.14 Bloglines feed presentation example. .................................................... 21

2.15 Google Reader presenting a headline list of a blog and an expanded item. ..... 22
4.8 A view on the tabs available on the sidebar, for the same blog. From left to right: sorting criteria, tags, commenters and calendar. ............................................. 50

4.9 Here we present a use case scenario involving sequential filtering, using Vibe. After reading a post about ponies (top left), the user selects the corresponding tag, affecting the visible posts (top right). Then, after reading some posts and comments, he decides to filter the posts by commenter (bottom right). The results include only the posts with the tag "ponies" and commented by the selected commenter (bottom right). ......................................................... 51

4.10 Detail of the calendar before and after a filtering operation. ............................. 52

5.1 The alpha version of our prototype, including post navigation, post and comment expansion and read wear information. ......................................................... 54

5.2 The selected blog, as seen on the original interface, with the blog archive and a portion of the tag list visible on the right. ......................................................... 57

5.3 The Google Reader interface with the sidebar collapsed, as used during the testing phase. .......................................................... 57

5.4 The initial view of the selected blog on our prototype. ................................. 58

5.5 Easiness average for each group of tasks, on each interface, as rated by the users on a scale from 1 (not easy at all) to 5 (very easy) ................................. 66

5.6 Average usability scores for each of the tested interfaces, and corresponding standard deviation, as evaluated by the users, using the System Usability Scale [Brooke 96]. ......................................................... 67
List of Tables

2.1 Comparison table between the most significant solutions described previously. . . 28

3.1 The values related to the 127 blogs analyzed in this study, sorted by relevance. . 35

3.2 Statistical results related to the data obtained from the 127 blogs used in this study, sorted by relevance. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35

5.1 Comparison table including the average time of execution for each task, on each interface, and the corresponding standard deviation (S.D.), in seconds. . . . . . 63

5.2 Methods used in the post revisitation task, where users were asked to find a previously visited post, on the three interfaces. Please note that users could check more than one answer. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 65

5.3 Cumulative easiness scores for each group of tasks on the three interfaces. . . . 66

5.4 This table summarizes the ratings given by users to some of the features included in our prototype, on a scale from 1 (strongly disagree) to 5 (strongly agree). . . . . 68

6.1 Comparison table between our solution and the five most significant solutions described in chapter 2. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 73
## List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AJAX</td>
<td>Asynchronous Javascript and XML</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>CSS</td>
<td>Cascading Style Sheets</td>
</tr>
<tr>
<td>GWT</td>
<td>Google Web Toolkit</td>
</tr>
<tr>
<td>HTML</td>
<td>HyperText Markup Language</td>
</tr>
<tr>
<td>RSS</td>
<td>Really Simple Syndication</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>SUS</td>
<td>System Usability Scale</td>
</tr>
<tr>
<td>TRIB</td>
<td>Telescope for Responding comments for Internet Blogs</td>
</tr>
<tr>
<td>URL</td>
<td>Universal Resource Locator</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

In this chapter we describe the problem identified related to the visualization and exploration of blogs, along with the objectives defined in order to provide a proper solution to it. Then, we present the solution that we developed to that end and summarize the most important contributions and results of our work. Finally, we describe the contents of this dissertation in terms of how it is organized.

1.1 Problem Description

A blog can be referred as a frequently updated website in which added entries are usually listed in reverse chronological order [Herring 04]. But, more than websites, blogs are reflections of the multi-diversity we observe on the world we live in, an easy way for those who want to share their interests or experiences online. Most of them represent personal diaries, yet there are many consisting of topic oriented feeds, spanning a wide range of possible formats and being capable of gathering the thoughts of one or more authors.

According to a study performed by Xiaohui and Li [Xiaohui 10], the five major motivations for blogging are the need for affection, expression of one’s thoughts, obtaining benefit, sharing knowledge and information, and communicating with others. Bar-Ilan claims in her study [BI04] that topic oriented blogs “are excellent sources of secondary or tertiary information”, in which the authors aim to concentrate and filter their subjects of interest, disseminating and expressing their opinion about them. Every blog has a life cycle [Gurzick 06], evolving from a non-directed, personal storage to an established and interactive medium, each one captivating different readers.

More than a decade has passed since the term weblog was firstly applied by Jorn Barger [Herring 04] to his collection of links gathered from the Internet and, at its core, current blog format still resembles the one used by him in 1997. In fact, present-day blogs are not easy to navigate, if we are concerned about more than what is immediate. Information discovery is difficult for blog readers while exploring the usual interfaces, since they are rigidly structured and do not usually follow the readers’ needs. Most of the blogs are presented in a one to
three column format and miss useful features such as searching functionality or a post tracking calendar, relying mostly on archive links or tag clouds to allow further exploration. Readers are also unable to get a quick overview of the blog and have difficulty in keeping track of the already read entries and respective comments, specially on frequently updated and community centered blogs. Furthermore, they are often betrayed by the authors’ lack of usability awareness, which sometimes leads to confusing layouts and disorienting colour schemes.

There is the need to enrich the blog reading activity by involving readers as part of the process, studying their interactions with the blogs and bloggers [Baumer 08], but also giving them the freedom to build their own path through the available information. As determined by Laqua and colleagues in their eyetracking experiment [Laqua 07], contextual interfaces enhance blog navigation, providing readers improved orientation and encouraging more exploratory interaction strategies. This is one of the main motivations of our work, as it is described in chapter 2.

1.2 Objectives

The objective of our work is to reach a solution that enhances individual blog visualization and exploration, giving users a solid blog overview, including text and images, and an easier to use interface. This aims for quick identification of blogger posting habits and interests, but also faster access to the potentially most interesting entries. The use of images should also promote recognition rather than recall when searching for entries.

One of the main concepts behind our solution is that the user should be allowed to control what information should be presented at a given time. To support this, we intend to provide an interface that allows the interactive filtering of posts by using its contained information, such as tags, date, authors or commenters. This way the user can take advantage of his interests to enrich his information browsing experience.

This process should also be enhanced with the support of read wear. Every entry, whether it is a post or a comment, should present a visible cue indicating if it was or not read in the past.

To have a better understanding of the existing issues, we studied the most common blog format and identified what aspects should be revised. Further investigation on blog related research, such as community and subject similarity studies, made us aware that post comments are almost as important for users as the posts themselves and that they should also be properly addressed in our solution.

Most of all, our main objective is to meet the expectations of the users, in order to bridge a gap between old and novel interfaces regarding the visual representation and exploration of blogs.
1.3 Developed Solution

In this dissertation we propose a solution to the previously identified problem, focused on flexible navigation, and presentation of enhanced blog entries that promote content recognition. This solution takes into account several aspects identified as important for blog exploration during the survey made on the related work, described in chapter 2, and also some concerns collected from the users during usability evaluation sessions made throughout the design phase.

The general interface of our solution includes a title bar, a navigation window for presenting blog entries and a sidebar that provides overview functionality. On the navigation window, posts and comments are presented in collapsed form at first, to preserve screen space and thus allow a broader visualization experience. Each collapsed post includes the most important information associated with it, in particular the title, author, publishing date and time, tags and length of the post. It also presents the thumbnail of an image contained within the post, to improve recognition, along with options to navigate through the remaining available images.

The information on each post, namely author, tags and publishing date and time, and the name of the users that added comments to it can be used to filter posts. This filtering mechanism is integrated with the overview on the sidebar, which includes replicated filters augmented with information about the number of associated posts. Those filters are organized in lists under tags, commenters and calendar, so to provide users with a better notion on the content and dynamics of the blog.

In addition to these features, our solution also incorporates a form of read wear that allows users to differentiate unread entries from those that were already read, with the use of different background colors.

The solution reached was developed as a Web application which includes a client that comprises the user interface and a server responsible for the management of blog data. The server supports the extraction of content from common blogs publicly available on a popular blog hosting platform and includes a database for storing the relevant information.

1.4 Contribution and Achieved Results

The main contribution of our work is a new design for the interface of blogs that offers a better navigation method, combining entries in collapsed form with the support of a flexible filtering mechanism. This mechanism uses the information associated with posts and comments, namely author, date and time of publication, tags and commenters, to affect the visible posts at a given time, as determined by the user.

The most relevant filters available within posts are replicated in the sidebar, allowing users to decide the filtering criteria based both on contextual exploration and overview. The overview mechanism gives users an aggregated view of the filters available on the blog, with focus on the
most used tags and most active commenters, also including a renewed calendar to better track posts along time.

Besides these contributions, our solution also improves the recognition of posts with the inclusion of image and video thumbnails, along with read wear visual cues that aid users in distinguishing unread entries from the remaining.

The concepts applied to our solution allow users to have an improved experience in terms of visualization and exploration of individual blogs. In particular, the results of the evaluation with users indicate that the combination between the filtering and overview mechanisms leads to improved usability and faster task completion, as compared with other approaches.

1.5 Dissertation Organization

In this dissertation we begin by studying the work related with the visualization and exploration of blogs, in chapter 2, where we cover the context and discuss the contribution of tools aiming to solve similar problems. This study serves the purpose of showing how the problem that we aim to solve is addressed nowadays and serves also for identifying possible directions that can be taken to reach our solution.

In the following chapter (chapter 3) we describe a study performed by us on a set of blogs with the objective of better adjusting our solution to real blog content (section 3.1). The results of the performed research are consolidated in the second half of the same chapter, in section 3.2, where we explain and justify every decision that affected the design of the interface for our solution. The resulting prototype, Vibe, is described in chapter 4, which includes a presentation of its general architecture and information flow, along with a detailed description of the server and client functionality.

The developed solution was evaluated in two phases, each of those described in detail in chapter 5, where we explain the used protocols and results obtained, along with the corresponding discussions.

To finalize, we present the conclusions attained after the evaluation of our work, in chapter 6, highlighting the main contributions and concluding with the possible directions that future work may take.
Chapter 2

Context and Related Work

Blog exploration and visualization tools relate to several research topics that go from interconnection and subject similarity to individual blog exploration. In this chapter we begin by briefly covering some studies related to blog interconnection, community and subject similarity, since these represent the current main research topics that motivate the development of blog visualization tools. They also contribute to a better understanding on how blogs are seen and used.

Taking that into account, we put the focus of our project on the visualization and interactive exploration of individual blogs. To provide a background on individual blog visualization and exploration, we analyze the structure and presentation of the most widely used technique nowadays. Then, we describe some related projects that are worth noticing and serve as motivation for our current work, covering both blog visualization tools and aggregators.

At the end of this chapter, and since this work is about visualization and exploration, we present some techniques that are relevant in this context and that could be adopted by our solution. Finally, we compare the existing solutions for blog visualization and exploration and discuss their main problems and contributions.

2.1 Context

Current research on blog visualization is more centered on blogspace network and information propagation analysis than on individual blog visualization. Here we present some studies related to these topics, to understand both the directions that blog research has taken and how our solution could be inserted into this context.

2.1.1 Blog Interconnection and Community

Studying the interconnection between blogs has revealed to be useful in identifying communities and common interests among bloggers and commenters.
Herring and colleagues [Herring 05] studied the interrelation between blogs by analyzing inbound and outbound blog linking. For this, they used Pajek\(^1\), a tool intended for the analysis and visualization of large networks. They concluded that most of the available blogs are less-connected (mostly with only inbound or outbound links) or unconnected to others, while there are some clusters of reciprocally linked similar blogs. On the other hand, the most popular blogs reveal to be less interconnected with others, even though they have more inbound links.

Using a different approach, Chin and Chignell [Chin 06] propose a hypertext model that discovers blogspace communities by analyzing links left on comments attached to blog entries. When directed to other blogs, those links are believed to convey a greater sense of interconnection between blogs, suggesting a social relationship among bloggers.

The fact that blogs are usually written by people led to the need of studying their emotional states. Harris and Kamvar have developed a system [Harris ] that periodically searches the Web for blog entries containing sentences starting with the expression ”I feel” or ”I am feeling” and records the information that follows, up to the period. The data is then stored in a database, the feeling is identified and then represented as a particle among others. Each particle can be clicked to gain access to the sentence or even related picture that originated it. This tool allows users to see the data through different views, or movements, as the authors call them, that range from particle representation to statistical analysis, including age, gender and location filtering.

Comments and commenters take a significant role on blog activity. Their interactions are an extent to blog entries and may cover interesting related topics not immediately identifiable in the main post. Focusing on the most noticeably active social cues of blogs, Tsuda and Thawonmas [Tsuda 05] propose a KeyGraph [Ohsawa 98] based approach that allows readers to follow discussions in the comments of a blog entry without having to read all the comments.

They started by selecting the most active story in the Hall of Fame of Slashdot Japan

\(^1\)http://pajek.imfm.si/doku.php , last accessed on 27 November 2010
(with 1018 scored comments) and applied KeyGraph to the whole data set. Then they repeated KeyGraph application two times, using successively smaller data sets, narrowed by higher scores, as represented on Figure 2.1. Each one of the data sets was given for analysis to groups of three human subjects, who were asked to select a limited number of keywords and to compose summaries of comment discussions. The results of the experiment led Tsuda and Thawonmas to conclude that the successive application of KeyGraph to smaller chunks of data gives a higher number of discussion scenarios than the common approach.

2.1.2 Subject Similarity

Blogs are also about information and helping readers determining which blogs or posts are more related to their interests.

Taking advantage of the concept of blog entry similarity, Perez-Quiñones and colleagues created VizBlog [PQ07], a tool that represents blogs as nodes in a diagram, connecting them when a blogger comments about another blogger’s posting or when content similarities are detected. Similarity measurements are based on frequent keyword comparison, using a vector space model to calculate a similarity index, which is reflected on the thickness of the line connecting two blogs, as seen on Figure 2.2. The tool also provides both a keyword cloud and basic searching capability, allowing users to search among the top keywords, which results in highlighting the nodes containing the searched word. Then, the user can click on one of the highlighted nodes to access the blog entry.

Takama and colleagues [Takama 06] examined the relationship between news articles and blog entries, using an interactive visualization system that enables users to generate keyword maps in order to follow news distribution in blogspace.

Similarly, but with the needs of business and government intelligence analysts in mind, Gregory et al. [Gregory 07] present a methodology for blog analysis using the In-Spire system [Hetzler 04], a mature document visualization tool. This tool allows users to harvest blogs, view them by thematic content, isolate keywords or bloggers of interest, run queries or visualize changes in content over time. The In-Spire system presents information using several visual
metaphors to expose different facets of the textual data, from which the authors highlight the Galaxy view, depicted in Figure 2.3. Given this scope, the authors concentrated their effort in the isolation of blogs by language and in the identification of sub-themes within language groups, with the aid of machine translation software.

Figure 2.3: The galaxy metaphor within the In-Spire system presents individual blog posts as dots, clustered according to their thematic content.

On a more commercially related level, Tirapat and colleagues developed eNulog [Tirapat 06], a tool that uses topic maps to assess if there is a relationship between blog activity about certain products and their success. This tool provides users a different set of views over each topic map, such as a graph view, as seen in Figure 2.4, allowing them to inspect different aspects of the analyzed data set.

Although our work is more focused on individual blog visualization, the concepts behind these tools could be used to extend our solution to other domains. As an example, those extensions could include the identification of the emotional state of the authors, a mechanism to follow discussions in blog comments or the aggregation of blogs with similar content.

2.2 Blog Exploration and Visualization

In spite of the aforementioned centering of blog research on community and information propagation studies, individual blog visualization is still a relevant matter, in which some noteworthy steps have already been taken. Here, we present some of the most relevant and recent works on
Figure 2.4: Excerpt of the graph view example of eNulog about the movie *Harry Potter and The Goblet of Fire*.

... and visualization, but not before discussing the presentation and the structure of the most common blog format.

### 2.2.1 Usual Blog Format

The current format of blogs is mostly imposed by blog creation software, which determines a one to three column layout, a reverse chronological post sequence, and also the presence or absence of some structural features, as referred by Herring et al. in their research [Herring 04].

**Presentation**

A blog that follows the usual format is normally composed of a column containing a sequence of posts and one or two sidebars, all limited by a header and a footer. As one can see in the example depicted in Figure 2.5, the header can include both text and graphics, normally intended for title or subject representation. Most of the time these are clearly oversized and distracting. The footer, on the other hand, is commonly left empty, but there are some who use it for advertising or for content intended to be accessible to the reader in every page of the blog.

The main column of the blog, where the true content is supposed to be, consists of a limited number of sequenced posts, each one comprising a header and a footer on its own, around a body containing the main matter. Even thought post header and footer purposes suffer slight variations from blog to blog, they are generally used for stating the title of the post, the posting date and time, the authorship and sometimes the number of related comments and tags or other types of general metadata. The sidebars usually contain information about the author, links that support blog navigation, links to other sites and sometimes custom plugins and advertising.

Considering all of the above, it is clear how difficult it is for a reader to distinguish between the noise and what is really important in a blog.
Navigation

As previously referred, blog posts are sorted in reverse chronological order, not allowing the reader to customize his reading experience by changing the post sorting or filtering criteria.

The usual blog interfaces require users to scroll down through an arbitrary number of posts before giving access to navigational links to previous posts. This makes browsing slow and tedious, specially for readers following frequently updated blogs. It is also very difficult to identify if a post has or has not been read in the past. The presence of read wear in common blog interfaces is rare and usually connected to standard HTML linking, through which visited and unvisited links differ in color. This method also requires direct access to posts, not covering the common situation where the user reads the posts integrated in the blog post list.

Basic search is also limited to some blogs. If we consider the blogs created with Blogger\(^2\), only the ones retaining the Blogger top bar can offer this functionality. Besides this, most of the blogs are not equipped with overview capabilities, still depending on sidebar archive links to provide further exploration. Usually organized in terms of years and months, this feature can be useful on small blogs, but it poses a problem on those receiving frequent updates, often leading to frustrating hit and miss information browsing sessions. More recent blogs provide tag clouds in order to give hints on general blog context and content. Still, those are clearly not enough to put readers in control of their blog reading experience.

Fortunately, there have been some attempts to solve this problem, with the development of more sophisticated visualization tools as result. Some of the most significant blog visualization solutions are described in the next subsections.

\(^2\)http://www.blogger.com, last accessed on 9 September 2011
2.2.2 Blogger Views

In an effort to modernize current day blogs with the latest in web technology, Blogger announced five new dynamic blog templates earlier this year\(^3\). These templates can be applied to almost every blog hosted on Blogger by adding "/view" at the end of the corresponding blog URL. Blogger Views rely heavily on the presence of image content and are based on five visualization metaphors, namely Flipcard, Mosaic, Sidebar, Snapshot and Timeslide.

![Figure 2.6: Six different views of the same blog using the five metaphors of Blogger Views: Flipcard - Recent (top left), Flipcard - Label (top center), Mosaic (top right), Sidebar (bottom left), Snapshot (bottom center) and Timeslide (bottom right).](image)

The first view, Flipcard, provides the option to arrange posts by most recent, date, label or author. Arranging posts by date, label or author groups them into horizontal lists, labeled according to the selected option, as we can see on Figure 2.6, on the top center. Each post is represented by a square thumbnail image that shows the date of publication when flipped over. The option to display posts by most recent differs from the previous and arranges them in a grid, as we can see on the top left of Figure 2.6, also adding the title of the post and the number of associated comments to the flipcard information. The Mosaic view, as the name indicates, organizes posts into a mosaic of images with different dimensions that reveal the corresponding title when hovered over, as presented on the top right of Figure 2.6. The Sidebar view is probably the one that has more resemblance to the usual blog presentation. It includes a post list on the left of the interface and the full content of the selected post on the right, as we can see on the bottom left of Figure 2.6. The post list is sorted by reverse chronological order and can be browsed to gain access to previous posts. Each post representation on the list includes a small thumbnail of an image included in the post, the title of the post and the number of comments added to it. The Snapshot view differs from the remaining views in that it represents images instead of posts, arranged by most recent activity. This means that the same post can be represented more than once, depending on the number of images contained. Each snapshot has the title of the corresponding post as label and reveals both a small portion of the text of the post and the number of comments when hovered over, which can be seen on the bottom

\(^3\)http://buzz.blogger.com/2011/03/fresh-new-perspectives-for-your-blog.html, last accessed on 14 September 2011
center of Figure 2.6. The last of the presented views is the Timeslide, which presents posts in
a three level vertical list sorted by reverse chronological order, that we can see on Figure 2.6,
on the bottom right. The first level of the list highlights some posts along the timeline of the
blog, each of them including the date of publication, the title, an image and a portion of the
text included in the post, if available. The posts on the second level of the list retain only the
title and portion of text, while the third level posts are limited to the title only. Hovering over
a post using this view reveals the number of corresponding posts.

Clicking on the representation of a post on most of these views gives access to its content on
a new screen, which usually implies hiding the interface through a horizontal shift transition.
After selecting a post of interest, the user can return to the previous view or continue to browse
posts sequentially. The few exceptions to this behavior are related to the Mosaic and Sidebar
views. While on the first the content of the post is shown on the same screen and does not
include access to comments, the Sidebar always maintains a visible post of interest onscreen
with its content fully accessible.

Although there are some concepts present here that contribute to the recognition of entries
by the user and are thus on par with some of the objectives previously defined for our work, the
main purpose of these views is to provide a showcase on recent web technologies such as AJAX,
HTML5 and CSS3. Overall, Blogger Views lack several characteristics required to be considered
a proper evolution of the usual blog interfaces as a means of exploration.

2.2.3 iBlogVis

Figure 2.7: iBlogVis allows the visualization of blog entries along a timeline.

Indratmo, Vassileva and Gutwin developed a tool [Indratmo 08] that represents visually blog
entries and related comments along a chronologically ordered timeline. As seen in Figure 2.7,
each entry is represented by a diamond above the timeline, to which it is connected by a vertical
line. The length of that line expresses the size of the entry. The same idea applies to the
comments, located below the timeline, which are represented by a circle, whose size reflects the number of comments. Clicking on a diamond gives access to the respective blog entry, which appears in a popup window. Users are allowed to zoom over the timeline and filter by both tags and commenters, using a query interface that reflects the results in real time, including a time slider and information tables, along with tag and commenter lists similar to the tag lists seen on some usual blog interfaces. Blog entry navigation is also supported by read wear cues, that allow for quick identification of already read entries.

This tool represents a remarkable advancement in blog visualization, but it lacks some important features, such as advanced search and the ability to jump directly to the most recent post. Furthermore, the tool drastically breaks with the usual blog presentation, which translates into decreased familiarity to the user, also failing to provide proper integration with the content itself. As previously referred, access to the content of the blog is limited to the presentation of the text of each post in a popup window, thus excluding comments, and does not take advantage of graphic elements placed on entries, such as pictures, to ease the identification of potentially interesting entries.

2.2.4 PostConnect

Bross and colleagues present PostConnect [Bross 10], an interactive visualization tool that, similarly to iBlogVis, allows the exploration of individual blog archives. PostConnect uses a circular interface to display all the posts of a single weblog as grey segments, arranged chronologically and clockwise on the outer circle. To explore the contents of the blog, the user first chooses a post of interest, marking the corresponding segment as active, which creates a straight line connection to the center of the interface, identified by A and C in Figure 2.8, respectively. This interaction leads to the generation of visual relationships between the active post and the remaining ones on the blog, based on the associated tags. Tags, in turn, are represented on the center of the interface by a grey pie chart, identified by B in Figure 2.8, each of them reflecting the number of posts they are associated with.

Following the selection of the post of interest, the associated tags are highlighted with unique color codes and connected to the related posts using sinuous lines that reflect the color of the tag they are connected to. These lines also encode the degree of relationship between posts with the use of transparency, through which stronger lines indicate a likewise stronger content or contextual relationship.

The circular visualization component is complemented by a listing with all the posts and tags related to the currently activated post, available in the top right area of the user interface. Here, PostConnect also offers a filtering mechanism that allows for both selecting and limiting the data available for exploration and analysis. Above the lists and filtering mechanism the interface displays synthesized information on the post, including title, author, timestamp, number of comments and current ranking.

Besides the information about the blog, PostConnect also indicates the presence of inbound
and outbound links connected with the blog. These indications are encoded by differently colored segments, identified by D and E in Figure 2.8, and have correspondence with references maintained on the top right corner lists.

Considering the concepts presented here, and taking our objectives into account, we must recognize that PostConnect makes it easy to identify the evolution of tag usage over time, albeit this is still limited to the focus on the selected post of interest. However, both the circular interface and connection mechanism have their limitations. As referred by the authors, it is not reasonable to represent a blog archive containing hundreds or thousands of posts on the same screen, due to visibility and usability issues, requiring users to zoom over particular regions to further explore the content. In fact, the major contribution of this tool results from the combination of the circular interface and the filtering mechanism as a whole.

As far as connections between tags and related posts are concerned, these may lead to visual confusion if there are many tags used at the same time. On the other hand, limiting the number of tags may highlight more frequently used categories, but simultaneously hide potentially interesting related content.

PostConnect is presented as a powerful blog archive exploration tool. Nevertheless, the lack of integration with the actual content of the blog and the complete absence of reference to comments makes it a blog overview mechanism rather than a fully capable blog reader.
2.2.5 TRIB - Telescope for Responding comments for Internet Blogs

Lee and colleagues propose TRIB (Telescope for Responding comments for Internet Blogs) [Lee 09], a visualization software capable of showing clusters of blog comments within the same screen, using user-defined filter dictionaries.

For visualization, TRIB uses the solar system hierarchical metaphor, where the main subject article represents the sun, in the center of the interface, and the associated set of keywords represent the planets around it. Comments, on the other hand, are placed around keywords as satellites whose diameter reflects the length of the corresponding comment. The proximity between elements is determined by the strength of their relation. This means that more frequently used keywords are represented closer to the article and comments containing more occurrences of the same keyword are placed closer to it.

As mentioned before, TRIB requires the definition of a dictionary of keywords in order to classify the replies to a post according to a topic. In Figure 2.9 the authors present the result of using TRIB to view an article with 1,837 comments, classified according to a dictionary for Korean Politics. As we can see there, the main article, represented by a yellow circle in the middle of the interface, is surrounded by keywords, represented as grey squares. Comments filtered by the dictionary are placed around the corresponding keywords as red circles, as opposed to unfiltered comments. These, in turn, are represented by salmon circles and arranged along the diametrical line according to the author identifiers.

Figure 2.9: TRIB uses the solar system metaphor to represent keywords around an article and comments around each keyword.

Regarding interaction, users can click on comment circles to reveal the corresponding text content on an overlapping window. This also highlights temporally adjacent next and previous
comment circles with different colors to ease the identification of the corresponding conversation thread.

TRIB manages to solve the problem of representing a high amount of blog comments on the same screen by using hierarchical visualization. However, that heavily depends on the previous definition of a dictionary, which may reveal to be cumbersome to the user. The use of predefined dictionaries can alleviate the problem, but does not avoid the difficulty of having to jump from circle to circle to follow a certain thread of discussion. The bottom line, and considering the purpose of our work, is that TRIB is too much focused on the visualization of comments, thus not providing the necessary tools to fully explore a blog. On a side note, it could be combined with another tool, such as PostConnect, as a complementary overview mechanism.

2.2.6 The Focus-Metaphor approach

Laqua and Brna [Laqua 05] emphasize the importance of combining design theory, cognition psychology and educational theory in the development of novel interfaces. The Focus-Metaphor approach aims to deliver highly adaptive interfaces that maintain a primary focus element along many secondary and peripheral focus elements. Not immediately related to blog visualization, this approach motivated the development of a visualization tool by Laqua and colleagues, which they used in a comparison study between blog visualization interfaces [Laqua 07]. Little of the version used for the comparison study is described in the related paper, which leads us to the originally developed Focus-Metaphor interface.

In a Focus-Metaphor interface, the elements are hierarchically structured and should not exceed the capacity of short-term memory. The primary focus element is usually the largest element on the interface and is located in the middle of the screen, while the other relevant elements, the secondary focus elements, are organized in groups around the main element. The peripheral elements consist of auxiliary functions that, although not directly related to the primary focus element, are necessary for an overall effective interaction. The main purpose of this concept is to translate user attention on an element into an adequate interface reaction, dynamically switching between primary focus elements and rearranging the remaining elements accordingly. In order to select a primary focus element, the user must choose an element of interest in the interface, which is then transferred to the top layer, providing all of its information. This rearrangement process is done through smooth and continuous motions, involving all the elements in the interface.

The first Focus-Metaphor prototype tested involved nine different modules, seven of which could be used simultaneously, as can be seen on Figure 2.10. The authors used this prototype to study user eye behavior in an eye-tracking experiment, which included the interface used in the first test, along a similar non-automated one and an adapted static grid interface, resembling a common Web interface. After recording and studying user eye behavior such as gaze path, pupil diameter and fixation of the users, Laqua and Brna concluded that the Focus-Metaphor interface greatly improved user attention, while providing higher navigation flexibility.
The Focus-Metaphor approach to interface design is a valuable contribution to the development of user-centered interfaces, giving users a greater sense of control and immersion. Although the Focus-Metaphor interface performed fairly in the study done by its authors [Laqua 07], as they refer, it is difficult to assess if this approach is adequate for the blog exploration environment, since it does not properly address some issues present on the usual blog interface, such as the lack of an overview, while failing to maintain a structural connection with the most commonly used weblog interfaces.

2.2.7 StarTree® Interface

The StarTree Interface is a system developed by Inxight Software, now property of SAP⁴, intended for advanced visualization, allowing the display and exploration of hierarchies and networks. StarTree uses a dynamic navigation hyperbolic tree whose nodes are related to the categories seen on usual blog interfaces. Each of these nodes can be clicked to gain access to the original blog entry, as seen on Figure 2.11. This tool was compared both with a common blog interface and the Focus-Metaphor Interface [Laqua 05] in the previously referred study conducted by Laqua and colleagues [Laqua 07], in which it was considered to outperform both solutions in terms of user error rates.

Despite this performance advantage, and although there is limited information about it, the StarTree Interface is another tool that fails to maintain a bridge between novel interfaces and the usual blog interface. Besides that fact, the tree-based navigation may be important for determining related subjects, but it seems that a high multiplicity of subjects can lead to a confusing and unstable interface that changes from blog to blog, not allowing users to have a proper notion of posting over time, nor an easily understandable overview.

⁴http://www.inxightfedsys.com/products/sdks/st/default.asp , last accessed on 9 September 2011
2.2.8 Taglines

Not directly related to blogs, but relevant to the subject is Dubinko and colleagues’ work [Dubinko 07] in visualizing tags within Flickr\(^5\), a picture sharing website. For this, they resorted to an animation paradigm that allows users to observe and interact with tags as they evolve over time. Tags are shown to users through one of two interchangeable metaphors, ”river” and ”waterfall”, both of which use a time bar and controls for playing and pausing the animation, as seen on Figure 2.12. The use of tags compelled the authors to first define the concept of ”interestingness”, a frequency measure that influences the size of each tag and the number of related pictures that are shown in the interface. The ”river” metaphor, for instance, displays tags ”flowing” slowly from right to left, in order to give a quick overview of tags over time. On the other hand, the ”waterfall” metaphor aims for the study of tags that are considered interesting across multiple days. This is shown by displaying tags along a column and increasing the number of pictures presented in a row if the corresponding tag persists for more than one day.

\(^{5}\)http://www.flickr.com , last accessed on November 28 2010
change over time than for information navigation, which requires a more stable and ordered presentation. Nevertheless, a similar view could be interesting in a blog exploration environment as an overview tool.

2.2.9 Lifelines

Departing from the Web context, but still in the visualization domain, Lifelines was developed with juvenile record visualization in mind, but Milash and colleagues [Milash 96] show its versatility in visualizing personal histories of other kind, like medical records. This tool uses a timeline to frame lines whose characteristics are used to illustrate real life measurements, like treatment or assignment length.

The ideas underneath Lifelines, such as the use of a timeline to represent sequences of events, are good starting points for the development of tools that also need to display time-related content. But, as it is, Lifelines would not be adequate as a base for tools aimed at the exploration of wider collections of information, due to its focus on displaying single personal records.

2.2.10 VisGets

VisGets [Dörk 08] consists on a framework composed by widgets that, when used together, aim to deliver a Web information search and display solution. This solution replaces standard complex text-based queries for visually supported dynamic casual queries, along with visual representation of search results. To achieve this, Dörk and colleagues started by choosing time, location and tags as the main information dimensions for their initial system.

In order to give a temporal, spatial and contextual localization of the information displayed, each of these dimensions is materialized through a VisGet. The Time VisGet represents the amount of information published on each time element within the range indicated below the graph. This time range can be widened or narrowed by using interactive sliders, and there is also the option of selecting a time element in order to access more detailed time related information. For the Location VisGet they used a world map based on the Google Maps API, in which individual and aggregated items are represented by squares and circles. Dynamic queries here are done through an interactive zoom function that simultaneously delivers more or less detail and sets the boundaries around the currently accessible information. When clicking on a square or circle, a user is then given the desired details-on-demand by showing the contained information. The Tag VisGet provides an alphabetically sorted set of the most frequently used tags included in the domain collection, allowing multiple tag selections, which are used as topic filters. The overall results are shown below the VisGets, on a matrix containing the headlines of the filtered items, letting users access detailed information by hovering over a result.

VisGets presenting the same domain can be linked, as depicted in Figure 2.13, allowing multidimensional interaction that leads both to weighted brushing and query refinements. Weighted
Figure 2.13: Coordinated time, location and tags VisGets, with related results.

brushing is activated by hovering over a VisGet item, which highlights the strongly related items on all the linked VisGets. Query refinements, on the other hand, reflect the filtering made on a VisGet on every other linked VisGet, allowing the user to progressively narrow the search results without losing previous refinements.

The VisGets system uses a database updated from RSS feeds with items in which the dimension related characteristics can be extracted or derived for display on the interface. The large amount of available information brings up the problem of data transfer management, which is vital to deliver a responsive system. To solve this problem, the authors developed delta queries, a feature that identifies overlapping information after a query refinement and prevents redundant and bandwidth consuming updates.

Although not created with blog visualization in mind, VisGets seem to be a valuable tool for situating and browsing general information, specially when used together, somewhat resembling news aggregators in their purpose. Like an aggregator, the framework has also support for bookmarking, since each query is encoded as a unique URL, as referred by the authors. However, it is uncertain how suitable the underlying framework is for blog exploration. The VisGets presented by the authors in their demonstration seem to be more focused on localization than on information itself, so there would be the need to design appropriate VisGets for this particular case.

2.3 Aggregators

Aggregators or feed readers follow a different approach to information visualization. Instead of focusing on individual website reading, an aggregator collects entries of selected websites, including news sites or blogs, and organizes them in a single page, creating a kind of personal newspaper. These tools can be Web-based, client applications or libraries and can appear built into applications such as Email programs and Web browsers.
An example of a popular Web-based aggregator is Bloglines\textsuperscript{6}, which includes "blog and news feed search, online subscriptions, news reader" and support for embedded video and photo-feeds, among many other features. After registering, the user can add or import feeds to a library, a list with presentation and management features resembling those seen in common Web browser bookmark lists, as depicted in Figure 2.14, on the left. Each stored feed label is followed by the number of unread information items, such as blog entries or news articles. The main window in Bloglines, on the right of Figure 2.14, is intended for presentation and navigation, switching between the starting page, the feed reader, the search results page and the other functionalities of the Web application. However, when it comes to the presentation of blog entries, there is a noticeable difference between different feeds. While some, like the Bloglines news blog, are presented with full posts, others only include the headlines and sometimes thumbnails, requiring users to access the original source to read the full post. Even in this case, both the presentation and the access to the original source differs from feed to feed.

![Figure 2.14: Bloglines feed presentation example.](image)

Also Web-based, but more popular worldwide, Google Reader\textsuperscript{7} is a simple and very usable aggregator. Similarly to Bloglines, it requires registration before giving access to its features, but in the case of the Reader, the possession of a Google account should suffice. When accessing it, the user is presented with a condensed view on the latest unread headlines of the subscribed feeds, which can be related to news sites, blogs or similar websites. By using the left column, seen on Figure 2.15, the user can easily add new subscriptions or access and manage the existing ones. Google Reader also encourages the exploration of other sources by providing a list of recommended articles.

Feed sources are accompanied by the number of unread items contained and selecting one of them provides a list of items on the main column of the reader. The items on the list are shown in a reverse chronological order and can be presented in an expanded or a condensed format. Each item presented in the condensed format includes the corresponding title, a small portion of the text contained and, oddly, the date when it was received by the aggregator instead of the date of publication. The date of publication and the corresponding time are only shown when hovering over the date of receipt. The expanded format is similar to that of a traditional

\textsuperscript{6}http://www.bloglines.com , last accessed on 28 November 2010
\textsuperscript{7}http://www.google.com/reader , last accessed on 13 September 2011
blog, but with a normalized presentation that is usually consistent between sources and that eliminates most of the noise present on the original site, adding some of its own, however, like advertising. The original source can be accessed by clicking on the corresponding icon on the far left of the listed item or on the headline of the article, after expansion, as seen on Figure 2.15. Only a limited number of items is loaded when the user selects a feed from the left column, which translates into improved performance. Although it might not seem clear at first, older items are dynamically loaded when scrolling down through the item list. Besides that, read wear cues are also present, since read items are faded, highlighting unread items from the rest. However, items older than 30 days or than the most recent 10,000 items per account are always marked as read\(^8\), which is understandable from the performance point of view, but has a major impact on the user experience while exploring blog archives. The Reader has also support for site statistics, though that information is hidden under a "show details" link on the top right of the post navigation list and is also affected by the aforementioned 30 day limit.

Google Reader does not have support for comments nor tags. On the brighter side, it boasts a powerful search functionality that allows the search through all sources or only selected ones.

More recently, Google released Reader Play\(^9\), an alternative method of viewing the feeds subscribed through Reader, that is more centered on media items such as images and videos, using the slideshow metaphor. Reader Play focuses on one item at a time, using a fish-eye view that makes use of a horizontal thumbnail list on the bottom of the interface for representing posts. Up to ten items within the same post are represented on a two column list on the right of the interface, as we can see in Figure 2.16, and can be clicked on to switch between items. The default view of a post includes the first sentences of the contained text, also giving the option to access its full content. Posts without videos or images appear as generic RSS thumbnails on the bottom list and have their text highlighted similarly to a newspaper headline when selected.

Navigation on Reader Play is limited to browsing the bottom thumbnail list, which does not support scroll, and to using the slideshow previous and next buttons on the far left and right of the interface, as seen in Figure 2.16. Following the steps of the Reader, Reader Play also has no support for comments nor tags. Despite its limitations and simplicity, there are some concepts

---

\(^8\)http://www.instantfundas.com/2011/08/google-reader-limits.html, last accessed on 10 September 2011

\(^9\)http://www.google.com/reader/play, last accessed on 13 September 2011
in Reader Play that could be applied to our solution, such as the media gallery functionality.

Figure 2.16: Reader Play is more focused on the presentation of images and videos as compared with Reader.

Bloglines and Google Reader are two of the most known available aggregators and the fact that they are Web-based makes them more relevant to our work. Other examples of aggregators include client software like BlogBridge\(^{10}\) and FeedReader\(^{11}\), some Web browsers and Email clients.

Even though feed readers are successful in gathering potentially interesting information and condensing it in a presentable form to the user, they still present some flaws and do not innovate the blog reading experience, lacking overview capabilities. Considering Google Reader as main example, the absence of comments and associated tags seriously limits its usefulness in the exploration of blogs.

### 2.4 Exploration and Visualization Techniques

Taking advantage of proven techniques in the development of a novel interface can be determinant for its success. Techniques such as read wear, fish-eye view or zoom have been proven effective in supporting navigation in interactive applications. Those techniques, among other relevant ones are briefly described in this section.

#### 2.4.1 Faceted Exploration and Navigation

Faceted navigation aims for flexibility, fluidity and seamless integration with other search and navigation methods and has been proven effective in exploring information collections, as stated by Marti Hearst [Hearst 08a], whose work is highly focused on this kind of interfaces. The use of the faceted navigation paradigm has grown in the last few years across diverse areas, being the already referred "wefeelfine.org" project [Harris ], from Harris and Kamvar, a good example

\(^{10}\)http://www.blogbridge.com , last accessed on 28 November 2010

\(^{11}\)http://www.feedreader.com , last accessed on 28 November 2010
of diversion from general information browsing. Another good example of faceted navigation is elastic lists [Stefaner 07]. This novel interface uses interconnected lists, each one with a distinct category on a domain of objects sharing similar characteristics, like the Nobel Prize winners domain, used for the demo depicted in Figure 2.17. The principle behind elastic lists takes advantage of the associated number of occurrences for the elements on each category, which translates into the relative physical dimension of those elements in their respective lists. The selection of an element in a list works as a global query refinement, readjusting the size of the elements accordingly.

One of the most relevant works on faceted navigation is FacetMap [Smith 06], a powerful data model intended for complex metadata browsing that retains a simple and familiar menu interface, as described by its authors. FacetMap aims for iterative and direct interaction with the visual attributes and output of the system in order to smoothly lead the user to the desired result. This tool makes use of the recognition instead of recall design principle by initializing with a query to the data store for facets that might be useful for organizing items. The results of this operation are then adapted to the available screen space and organized in terms of attributes into graphical items. These items can be used as filters to narrow the search until the desired results are identified by the user, as depicted in Figure 2.18. Although performing below some traditional approaches to targeted search tasks, FacetMap is a scalable tool that performs well on browsing tasks while featuring a rich graphical interface.

Faceted navigation may not deliver a complete solution for blog exploration on its own, but it is a powerful technique for data filtering and browsing when used together with complementary information display methods.

2.4.2 Other Visualization Techniques

Highlighting information items is essential for giving users a better notion of position while exploring large amounts of content. While seeming simple at first, this is a job of fine-tuning details that can make a difference, for the selection of a single item can lead to the risk of occluding other relevant information.
Luboschik and Schumann [Luboschik 08] study the benefits of using illustrative halos in interactive visualization, concluding that they enhance the perception of each selected item without disrupting or occluding the context, contributing to the reduction of its localization times. A good example of the application of illustrative halos is presented in their article through the image depicted in Figure 2.19.

The use of animation on interfaces is a touchy subject, because of the risk of distracting the user instead of providing the intended immersion.

Shanmugasundaram and Irani [Shanmugasundaram 08] analyze the effect of animated transitions in zooming interfaces, such as interactive metro maps. Animated transitions are considered to halve user task processing times, even though they do not significantly contribute to the accuracy. This study also suggests that the animation of some tasks can be reduced to $\frac{1}{4}$ of a second without jeopardizing task performance.

The read wear technique is useful for keeping track of user history and facilitating navigation.
The most evident example of read wear is probably standard HTML linking, in which visited links are distinguished from the unvisited ones using different colours. The value of this technique becomes evident when traversing large amounts of data, helping the user to avoid getting lost in navigation.

Alexander and colleagues [Alexander 09] designed a tool that enhances scrollbars with read wear marks and mark thumbnails in order to ease document revisitation. While improving the familiar scrollbar, the tool does not occupy more screen space, as can be seen on Figure 2.20.

![Footprints scrollbar](image)

**Figure 2.20:** The Footprints scrollbar by Alexander and colleagues.

Read wear can be extremely important in the blog domain, as was already demonstrated in iBlogVis [Indratmo 08] and Google Reader, and its potentiality should not be overlooked in the development of similar visualization tools.

Getting an overview of large amounts of content while concentrating on a specific item is a difficult mission, since it requires a clear view of all the browsed content within limited screen space. Fish-eye view has been used on interface development for a long time now and its usefulness is still patent in many interactive systems. With the increase of graphical computing power, this technique could be easily applied to blogs, allowing the user to expand and read a single entry among the remaining blog entries.

Tag Clouds have been increasing its popularity in content-centered systems, including blogs and media storage services. They represent grouped content-related words, whose font size reflects its frequency. As referred by Hearst and Rosner [Hearst 08b] in their study, tag clouds are ”primarily a visualization used to signal the existence of tags and collaborative human activity”, but its usefulness for information-related tasks seems to be inferior to standard listings such as alphabetical listing. As suggested by this study, this could be addressed by adjusting parameters within the tag cloud or by changing the layout. Combining tag clouds with elastic lists may be a plausible partial solution to the problem of blog navigation.
2.5 Comparative Table and Discussion

The solutions described in section 2.2 were designed to solve the problem of blog visualization and exploration, or at least a portion of it. Each one bears distinctive characteristics that are used to perform a comparative analysis in this section.

The chosen characteristics are important for both blog exploration and usability. The presence of a flexible navigation mechanism is of utmost importance, since it allows users to choose the exploration method that best suits their needs. We considered a system as having a flexible navigation mechanism if it has at least two fairly complete alternative exploration methods or one that is highly configurable.

A familiar interface usually leads to decreased user adaptation and learning times when switching from a similar interface. Overview capability allows users to get a global understanding of the blog and the same goes for site statistics.

On the other hand, user statistics help readers keeping track of their habits, just like read wear helps them keeping track of read entries. Word search adds much to navigation, specially for users that have a hint on what they are looking for.

As far as content is concerned, comment and image support is essential, since community interaction and visual content sharing are two of the most important features of blogs. Similarly, tags are considered a *de facto* standard as far as blog content identification is concerned, and therefore should not be overlooked. The presence of thumbnails facilitates the recognition rather than recall of entries as well. There is also the advantage that browser-based solutions have as opposed to desktop applications concerning accessibility.

The comparison between the solutions is summarised in Table 2.1, which includes the usual blog format for contextual purposes.

Not all of the earlier described solutions can be considered for comparison, since it would be unfair to include a tool like Lifelines [Milash 96] along with more comprehensive solutions like iBlogVis [Indratmo 08]. As far as aggregators are concerned, we chose Google Reader to represent them due to its worldwide significance as compared to similar tools. Besides this, the substantial differences between this tool and Reader Play, as well as the lack of integration between them, made us consider the latter as a totally independent solution.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Browser-based</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Comments</td>
<td>Yes</td>
<td>Yes</td>
<td>Limited</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Familiar GUI</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Flexible Navigation</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Images</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Overview</td>
<td>Limited</td>
<td>Limited</td>
<td>Yes</td>
<td>Limited</td>
<td>Limited</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Read Wear</td>
<td>Limited</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Limited</td>
<td>No</td>
</tr>
<tr>
<td>Site Statistics</td>
<td>Limited</td>
<td>No</td>
<td>Yes</td>
<td>Limited</td>
<td>Limited</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Limited</td>
<td>No</td>
</tr>
<tr>
<td>Tags</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Thumbnails</td>
<td>n/a</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>User Statistics</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Word Search</td>
<td>Limited</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 2.1: Comparison table between the most significant solutions described previously.
In this comparative analysis we begin with a review on the usual blog interface. As previously referred, the usual interface inherently lacks several features that we considered as being important for a blog exploration tool. Between those is the limited use of site statistics to ease decision making when browsing. The navigation is rigid and support for read wear is limited to HTML link color changes. The presence of an overview mechanism usually does not go beyond an archive of posts and there is also no place for user statistics. On the other hand, characteristics such as access to comments, the use of tags and direct access through any browser contribute to the definition of current day blogs. Not every blog has support for word search, though.

Recently presented as an evolution of the default Blogger templates, with the use of new technologies, Views share some limitations with the usual interfaces, also adding some of their own. Besides the referred limitations, Blogger Views drop the already limited read wear support and tags, further exchanging post archives for visual representations, using grids or lists of images. The use of thumbnails to facilitate the identification of posts, however, is a step taken on the right direction.

iBlogVis is probably the solution that best targets the problem discussed in this report, since it was developed with blog visualization and exploration in mind. It achieves to address structural flaws in the usual blog format such as the presence of overview capability using a timeline, also supporting visual statistics of the blog and comments access. The presence of read wear on blog entries is also an important feature that adds to navigation support. However, iBlogVis does not take advantage of image presence in blog entries and does not give access to the content of comments. Besides that, the demonstrated prototype is not Web-based, which reduces its accessibility. It also fails to maintain fundamental similarities with the usual format, to which users are accustomed.

PostConnect shares some similarities with iBlogVis in that is also aimed towards the visualization and exploration of individual blog archives. It has a flexible, novel navigation system that interconnects posts and tags, also integrating some site statistics. The limited access to the content, the absence of comments, images and read wear support, on the other hand, present serious limitations to a solution intended for blog visualization and exploration.

TRIB is a tool that is heavily focused on the presentation of the comments added to a post. Therefore, its support to blog navigation and overview is very limited overall. Nevertheless, it could be considered as a valuable complementary overview tool for the visualization of comments.

The Focus-Metaphor and StarTree interfaces lack related information, which clearly limits both of them in this comparison. While the Focus-Metaphor framework has previously been described as a good step taken in the development of user centered interfaces, there is not enough information to assess about the specific features of the interface used for blog navigation. Only image support and a high level overview could be identified in the gathered information. As far as StarTree is concerned, its most distinguishing characteristic is precisely tree navigation. It is uncertain however how adequate the StarTree interface is to blog exploration, since different amounts of information should also affect the navigational support provided by the tree. The lack of a thumbnail in each node prevents faster identification of interesting entries.
Although Taglines is not directly related to blogs and therefore does not present a complete solution to the blog visualization problem, it could be adapted to this environment, since it bears important attributes such as visual statistics and a timeline based overview. Its main purpose of presenting pictures over time makes the use of image thumbnails its stronger feature.

VisGets is, first of all, a generic framework for information browsing instead of a solution directed to the discussed problem. Though it may be possible to develop suitable widgets using this framework, we can only consider its presented features. Nevertheless, there are important items to highlight, such as the presence of a powerful search and overview system that correlates all the used widgets.

Google Reader is not exclusively directed to blog visualization and is also capable of reading news sites and similar sources providing automatic feeds, delivering a very good performance at it. But that is probably one of the main reasons that prevents it from fulfilling all of the requirements in the comparison table. Even though it maintains a bridge with the usual interfaces used for blog navigation, complemented with fish-eye expansion of entries, Google Reader lacks the presence of thumbnails to facilitate the recognition of collapsed entries. Comment support is only limited to presenting the number of comments associated with a given entry, but not always. Users interested in reading the comments must access the original source, which includes having access to the associated tags. Google Reader also includes useful site statistics related to recent posting and the presence of user trend statistics is a welcome addition. Again on the brighter side, word search support is a strong feature of Google Reader that is absent in most of the other solutions. The same goes for read wear support, although not without limitations.

Related to Reader, but with a totally different approach, Reader Play shifts the focus of the aggregator to the presentation of media content. However, it turns out to have greater limitations related to overview and navigation, only surpassing its relative in the support for thumbnails, which only adds to facilitating post recognition to some degree.

The majority of the compared solutions achieve to maintain a consistent presentation between different sources of information. However, none is capable of addressing all of the requirements for delivering a familiar yet enhanced blog reading experience. This is the main objective of our work, which should combine the most relevant features of the described solutions, such as a high level overview with visual statistics and image support, while using a familiar interface. It should also allow integrated post and comments reading combined with read wear and a flexible navigation mechanism.

Now that we studied possible blog exploration techniques, by addressing both usual and novel interfaces, there is the need to study blog content, not only because it is the main reason why users read blogs, but also because it can have a heavy influence on our solution. Given this, we performed a blog content analysis, which we present on the next chapter.
2.6 Summary

In this chapter we studied and analyzed some research related to blog visualization, starting with the current major areas of interest, namely blog interconnection, community and subject similarity. Then we described some relevant solutions to the problem of blog exploration and visualization, covering both usual, novel interfaces and aggregators, in order to provide a proper background on our current work and the underlying motivations. The following section was dedicated to relevant research related to visualization techniques that could enrich our solution. Lastly, we performed a comparative analysis between the most important solutions attempting to solve the problem of exploration and visualization of blogs or similar and discussed their contributions and shortcomings, with the objective of helping define the course to take in the design phase of our solution.
Chapter 3

Blog Analysis and Interface Design

Designing a solution for the visualization and exploration of blogs requires not only the knowledge of existing solutions attempting to solve similar problems, but also the analysis of the content that the solution is aimed at. Thus, we begin this chapter by presenting a study that we performed on a set of blogs. From this study we collected valuable information with the intent of adjusting our solution to the reality of current blogs, based on statistical analysis. The resulting data, along with the conclusions attained from the related work, lay the base for the design of the user interface for Vibe, as we further present in this chapter.

3.1 Blog Analysis

It is important to analyze blog content, so that the resulting solution properly handles the relevant information using suitable techniques. To this end, we conducted a study using a sample set of blogs extracted from one of the most used platforms, Blogger, and performed a statistical analysis.

3.1.1 Experiment

The blogs used for this study were selected by using the Blogger homepage\(^1\) as a starting point, for which we used the Blogger API for blog data extraction. The first blog selected was the first on the most interesting blogs short list presented on the Blogger homepage\(^2\). The following blogs were collected by selecting the Next Blog option on the Blogger bar, presented at the top of each blog. A few blogs did not have this option, so in those cases we used the back button of the Web browser to return to the previous blog and selected the Next Blog option again. Although this means that blog selection is not deterministic, it is not random either, since the subject of the blog resulting from the Next Blog option selection is usually fairly similar to the subject of the previous one, as we could infer. This also means that there is the risk of finding

\(^1\)http://www.blogger.com/start
\(^2\)On October 5, 2010
a previously collected blog again. In these cases we decided to select the Next Blog option once more, until a new blog was found.

Given the fact that we used the Blogger API for most blog data extraction in this experiment (we used a HTML parser to collect image links), only the ones successfully read by the API were analyzed. We analyzed 127 blogs, from a total of 150 collected. The 23 blogs not analyzed include blogs reported as containing unrecognized content and also blogs from which we were unable to extract image links.

We must also highlight a limitation in our blog extraction method, related to the Blogger API, which limits the number of extracted posts at once to 500, so only the most recent 500 posts for each blog were taken into account. 12 of the collected blogs were identified as being in this situation. Nevertheless, this should not significantly affect the results of our study, considering its purpose.

On the content level, and although this is not easily quantified, the vast majority of the blogs collected and analyzed can be considered exclusively personal blogs or highly related to the personal experience of the authors.

For this experiment we measured post and comment length, the number of images, tags and comments per post and the number of posts per month and per year. We also measured the number of authors and commenters per blog and the comments per commenter in each blog.

### 3.1.2 Results

In this experiment 28,872 posts were successfully extracted from 127 blogs. Each blog had an average of 227 posts, with each post motivating 3 comments on average. 95% of these posts had up to 10 comments, of which 50% had no comments. Each post had an average 204 words and 676 words was the maximum number of words for 95% of the posts.

Regarding posting frequency, on average, and for each blog, more than 7 posts were added per month. Also, 95% of the blogs received 23 or less monthly updates. As for year updates, each blog received 59 new posts on average and 95% of the blogs had 199 or less post additions. The updates in each blog were usually made by the same author and 95% of the blogs had 5 authors or less.

Adding tags only affected 36% of all the posts extracted, with each post having on average approximately 1 tag. No more than 4 tags were added to 95% of the posts. Each post included 2 images on average and 95% of the posts extracted had 8 images or less. Approximately 45% of these posts had no images at all.

As far as comments are concerned, a total of 91,916 were extracted from the selected blogs, with an average of 42 words per comment. 95% of those comments had 127 words or less. Furthermore, 13,219 commenters were identified. Since our goal does not involve studying blog community in its whole, we did not cross-reference commenter identifications and considered the
commenters from different blogs as separate populations. Due to different commenting policies between blogs, it is difficult to relate anonymous commenters. Therefore, different commenters were determined only by different authorship details. With an average of 104 commenters per blog, 95% of the blogs gathered 259 or less commenters. Each of these commenters posted an average of 7 comments and 95% of them posted no more than 21 comments.

The most relevant values regarding the amount of data extracted during this experiment and the statistical results obtained are summarized on Table 3.1 and Table 3.2, respectively.

<table>
<thead>
<tr>
<th></th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Posts</td>
<td>28 872</td>
</tr>
<tr>
<td>Comments</td>
<td>91 916</td>
</tr>
<tr>
<td>Tags</td>
<td>26 858</td>
</tr>
<tr>
<td>Images</td>
<td>56 041</td>
</tr>
<tr>
<td>Commenters</td>
<td>13 219</td>
</tr>
<tr>
<td>Authors</td>
<td>207</td>
</tr>
</tbody>
</table>

Table 3.1: The values related to the 127 blogs analyzed in this study, sorted by relevance.

<table>
<thead>
<tr>
<th></th>
<th>Average</th>
<th>Standard Deviation</th>
<th>95% of population ≤</th>
</tr>
</thead>
<tbody>
<tr>
<td>Posts per blog</td>
<td>227,3</td>
<td>145,0</td>
<td>500</td>
</tr>
<tr>
<td>Words per post</td>
<td>203,8</td>
<td>262,4</td>
<td>676</td>
</tr>
<tr>
<td>Comments per post</td>
<td>3,2</td>
<td>11,1</td>
<td>10</td>
</tr>
<tr>
<td>Tags per post</td>
<td>0,9</td>
<td>1,6</td>
<td>4</td>
</tr>
<tr>
<td>Images per post</td>
<td>1,9</td>
<td>3,8</td>
<td>8</td>
</tr>
<tr>
<td>Words per comment</td>
<td>42,3</td>
<td>47,4</td>
<td>127</td>
</tr>
<tr>
<td>Commenters per blog</td>
<td>104,1</td>
<td>333,2</td>
<td>259</td>
</tr>
<tr>
<td>Comments per commenter</td>
<td>6,9</td>
<td>42,9</td>
<td>21</td>
</tr>
<tr>
<td>Authors per blog</td>
<td>1,6</td>
<td>2,6</td>
<td>5</td>
</tr>
<tr>
<td>Posts per month</td>
<td>7,5</td>
<td>8,2</td>
<td>23</td>
</tr>
<tr>
<td>Posts per year</td>
<td>58,9</td>
<td>64,5</td>
<td>199</td>
</tr>
</tbody>
</table>

Table 3.2: Statistical results related to the data obtained from the 127 blogs used in this study, sorted by relevance.

### 3.1.3 Implications on the Solution

Previous research led us to conclude that it is important for users to be aware of the length of each blog entry before reading it. This applies to both posts and comments, for which we intend to give a visual representation of the length, acquired from the number of words counted. Given the resolutions available on current generation computer screens, the obtained results indicate that, for most entries, each word could easily be represented on screen by a bar with one pixel of width. Entries with more words than a certain value would be represented with the maximum length and marked as extensive. On the other hand, since many posts have very few words, a minimum value should be used for the length of the corresponding visual representation, so that it remains accessible according to Fitts’s law [MacKenzie 92]. Nevertheless, the obtained values should be considered as merely indicative and a percentage value should be used instead, given...
the multitude of screen resolutions available and the negative impact that a fixed size bar would imply.

Measuring the number of commenters per blog and the number of comments per commenter gives the user a good notion about the level of interaction between the author or authors of the blog and the community. Taking the obtained results into account, we believe we should keep a list of the most active commenters in each blog, so that users can easily follow their comments.

The number of tags per post obtained in the results suggest that no more than 4 tags should be considered and represented in our solution. The problem here should be identifying the relevant ones for the posts that have more than 4 tags, for we must also provide an option to access the remaining tags.

Regarding the number of images, the results indicate a big disparity between values. This suggests that we should give the users an option to have easy access to the images in each post, including a thumbnail and maybe immediate access to a limited number of quick links. Also, given the number of posts without images, a visual representation of the post could be placed as a thumbnail, including the first words of the post.

A similar disparity was identified in the number of posts per month and per year. This means that while in some blogs filtering by month could return tens of posts, in others that could return only a post or two. To minimize this, we should at least provide the user an option to adjust the time window or choose between presenting posts per month or per year.

These results have some implications on our solution, as we further explain in the following section.

### 3.2 Interface Design

In this section we present the design decisions that affected our solution, along with a series of low-fidelity prototypes that reflect the evolution of the concept prior to the development phase. The presented decisions and prototypes take into account our previous research on the existing solutions and on blog content analysis.

The initial low-fidelity prototype of Vibe is the first visible result of our survey on the work related to the visualization and exploration of blogs. As shown in Figure 3.1, our foremost concern was to address the navigation method within the blog. The typical blog user interface lists entries (or posts) in reverse chronological order. In our solution, we decided to keep this order to reduce user adaptation and learning times.

Each entry in the navigation window (on the left of the interface) is presented in collapsed form, to avoid visual clutter and to allow the simultaneous display of several entries, thereby increasing navigation efficiency. Similarly to the mechanism seen previously in Google Reader, each collapsed entry in Vibe can be expanded to display its full content, which, in the initial low-fidelity prototype, included the access to the comments. A collapsed entry contains information
that is typically found in the header and footer of a post seen through the usual interface. This includes the title of the post, the publication date and time and the tags associated to the post by its author. Adding to this, we opted to include the thumbnail of an image within the post, the length of the post and the length of the corresponding comments. While the inclusion of a thumbnail on each collapsed post increases its size in the list, as compared, for instance, with those of Google Reader, it also contributes to the recognition of the content within the post, even if it has not yet been read. In turn, presenting the length of the post serves as an indication for how much time the user will take reading it, influencing his decision to read it or not. The same concept is applied to the presentation of the length of the comments, which also serves as an indication on the number of comments for that post.

The sidebar of the interface, on the right of Figure 3.1, provides complementary functionality, such as sorting and filtering techniques that affect the posts presented on the navigation window. Still, the focus on navigation led to most of the intended functionality on the sidebar remaining undefined until more recent prototypes.

In order to validate the ideas applied to the low-fidelity prototype, we performed several informal usability evaluation sessions throughout the design phase of our solution, involving both users and usability experts. As a result, the prototype suffered some modifications, the most important of which concern the presentation of entries and the functionality on the sidebar, as indicated before.

Entries, by their turn, were rethought around two main concepts. The first concept led to defining comments as visually separate entries, through which each comment can be accessed independently from the post it is attached to. This solves two problems identified during the usability evaluation sessions, in particular, the accumulation of comment length bars on the collapsed post, which led to visual confusion, and the flexibility issue that required users to
expand a collapsed post whenever they wanted to read one or more comments. As we can see in Figure 3.2, on the top left, comments are now represented below the collapsed post as squares on a separate bar. This bar can be clicked on to reveal the list of comments, all in collapsed form, to maintain consistency. Collapsed comments are inspired in collapsed posts, although retaining only the name of the commenter, its length and the time of publication (see Figure 3.2, bottom left), to preserve space in the interface. Like a collapsed post, a collapsed comment can also be clicked on to reveal its full content. In addition, each comment can be accessed directly by clicking on the corresponding square on the comment bar. Since every comment square has similar presentation, hovering over it reveals the name of the commenter and the time of publication.

Figure 3.2: Evolution in the design of important components of the interface, in particular posts, comments and sidebar.

The second and probably most important concept affecting entries determines that each bit of information associated with a post should be able to be used individually as a filter, to affect which posts are visible at a given time. This means that selecting time, date or tag elements leads to the application of a filtering criterion that narrows the posts visible on the navigation window accordingly. To the already referred filtering elements we added the name of the author of the post and the name of the commenters. While filtering by the name of the author allows for presenting only the posts that were published by that author, which is useful in blogs with more than one, filtering by the name of the commenter displays only the posts that were commented by that person. This means that filtering operations actuate only on posts and no filtering result should contain less than a post. To maintain coherence, we do not allow users to use the publication date and time on comments for filtering purposes, since it could lead to empty results.

Filters can be combined sequentially to successively narrow post sets. In addition, selecting a previously applied filter again reverts its effect on the result set. To increase efficiency, two
related options were added to each post, namely an option to turn off all filters associated with
tags and an option to turn all previously selected filters, identified by "view all" and "Clear
Filters" in Figure 3.2 (top left), respectively. In accordance with valid suggestions received
during the informal usability sessions, these two options were removed from the collapsed posts,
also to remove redundancy. The option to turn off all tag filters was completely removed from
the interface, since only one tag filter is supported at a given time. As for the option to clear
all filters, it was correctly repositioned in the sidebar, as we can see in Figure 3.2 (right), since
it refers to a global option that should be available at all times on the interface.

The results obtained from the blog analysis led to other changes on the collapsed post, the
first of which has to do with the representation of the length of each entry, which was adjusted to
fit only the space between the thumbnail and the vertical space occupied by the post publication
time. Remaining changes include the ability to browse through the images within the post by
using navigation arrows under the presented thumbnail, as we can see on Figure 3.2 (top left).
This option is available only if more than one image is present within the content of the post.
For the cases where no image is found, the first few words of the post are presented instead.
Adding to this is the possibility to see the images in a larger resolution, on an additional window,
by selecting the corresponding thumbnail.

Still resulting from the data obtained from the blog analysis is the limitation to the number
of posts available per page, which was added so not to penalize the loading of blogs with many
posts, due to the fact that there is no common denominator among blogs concerning the number
of posts over time. To support this decision, we added navigation options on the bottom of the
interface accordingly, along with the information about the current page and the currently
displayed posts.

Also in this iteration, we added the support for read wear, defined by different background
colors or shades on the collapsed posts and comments, as we can see on Figure 3.2 (bottom
left). This feature determines that unread entries are highlighted from the rest and faded after
expansion, in order to facilitate the identification of previously read entries. Since individual
comments remain hidden until the expansion of the corresponding comment list, comment read
wear information is also presented on the squares that represent them in the comments bar. The
option to access the post on the original interface was also added, although this was transferred
from the title on the collapsed post, as in Figure 3.2 (top left), to the end of the content on the
expanded post, on the final prototype, to improve consistency and presentation.

The changes made to the sidebar are also the result of the usability evaluation sessions as
well as the application of the information visualization guideline "overview first, zoom and filter,
details on demand" [Shneiderman 96]. As we can recall, the initial low-fidelity prototype lacked
the presence of a solid overview mechanism, which was limited to a sketch of interconnected tag
and commenter lists. The need for an overview referring to time led us to add a post tracking
calendar to the functionality on the sidebar, as we can see on Figure 3.2, on the right. The
focus on the exploration of a single blog at a time also led us to remove typical aggregator
functionality, such as the options to add and select blogs, identifiable on the left of Figure 3.1.
These options were partially replaced by a field on the title bar of the interface for inserting the URL of the blog. In spite of these modifications, some uncertainty remained until the final design of the sidebar, which we describe below.

The sidebar functionality present on the final prototype serves both as an overview of the blog and as a complementary exploration mechanism. As illustrated by Figure 3.3, the functionality on the sidebar adds flexibility to the exploration of the blog by reflecting most of the filters available on the posts presented in the navigation window, thus improving the cohesion of the interface. The only exception has to do with hour and minute filters, since the occupied space on the interface would not justify the little contribution to overview. The presence of filters also in the sidebar implies that filters can be combined according to user preference, whether they are found in posts or in the sidebar.

Figure 3.3: The interface on the final prototype of Vibe, with the calendar selected on the sidebar.

The filters in the sidebar are divided into three categories, namely tags, commenters and calendar. Each filter in the sidebar is augmented with the number of posts associated with it that fulfil the current combination of filters (filtering criteria), along with the number of posts that contain that filter overall (see Figure 3.4). This information is used to sort the filters on each category, so that the first filter in each list is the one that has more associated posts in the current selection. The only exception applies to the filters found in the calendar. While tags and commenters are presented as separate lists, the calendar aggregates all the date filters found in posts, week day, day, month and year, in particular, as we can see on Figure 3.3, on the right. These filters are sorted chronologically in order to facilitate reading, as this increases the similarity with traditional calendars.

To prevent users from obtaining empty results when selecting filters, we applied a principle to the sidebar filters that extends to the rest of the interface. This principle indicates that only valid filters are visible for the user to select, which means that selecting a filter on the interface removes every filter on the sidebar that is not associated with the resulting posts.
Figure 3.4: Detail of a filter on the calendar. Given the applied filtering criteria we can see that, from the five posts published in day 14, three were published in the year of 2010. In turn, a total of nine posts were published in 2010. Notice that the sum of the posts that satisfy the criteria on each filter column, on the calendar, is equal to the total of posts that satisfy the criteria, five.

Besides preventing user errors, this facilitates the exploration of the blog by allowing users to successively narrow post sets until the desired result is obtained.

The remaining functionality on the sidebar includes the already referred option to clear all the selected filters and also global sorting criteria. The option to clear all filters is available on the top of the sidebar, as it is considered to be frequently used and improves the efficiency of the filtering mechanism, since it has the same effect as deselecting every applied filter one by one. The available sorting options, in turn, are aligned with the flexibility of the interface and are related to the most relevant information present on posts. Users are allowed to sort posts by time, post length, popularity, number of images and number of videos. Some of these criteria were added as a result of the informal usability tests. In particular, we added popularity, which refers to the number of comments per post, and number of videos, as support for video thumbnails was added later to the prototype. Each criterion allows for ascendant or descendant sorting of posts.

Taking this information into account, and considering the evolution of our prototype along time, we turn our attention to the navigation window of the interface once more. In addition to the previous improvements made to collapsed posts, a couple of modifications were included in the final prototype, as a result of a series of informal usability tests, this time using a functional prototype. The aforementioned tests determined the need for the support for video thumbnails, as well as time relevance indications concerning how long ago the post was added, since these were considered by users as contributing significantly to the identification of posts. The location of these new elements was determined so that the impact on the existing information was minimized. Therefore, time relevance information was placed below the time filters, also for coherency, and a space for the video thumbnail was reserved on the right of time information, as we can see on Figure 3.3 (center). Video thumbnails only appear if there are any videos within the posts and the remaining concepts concerning image thumbnails are applied here too. To comply with these decisions, the concept that determined the visualization of images in bigger resolution on an additional window was expanded to support videos. The resulting interface consists of an additional gallery layer that presents the focused media on the left and a list of thumbnails on the right, that we can see in Figure 3.5. Changing the focused media is possible by selecting a thumbnail from the list on the right or by using the sequential browsing options.
on the bottom of the interface. Users can return to the post list by selecting the appropriate option on the bottom of the gallery interface or by selecting the title of the blog, on the top.

Figure 3.5: Gallery view of the images contained within a post. On the left we can see the focused image and on the right the list of remaining images as thumbnails.

Further analysis to the interface led us to identify the need for a better identification of the currently used filtering criteria. The corresponding information was added to the bottom of the interface, and allows users to have a better notion of what is being displayed on the interface. Additionally, we decided to include an option at the end of the content of each expanded post that allows for its collapse, to improve efficiency and to prevent users from having to navigate to the top of the post once more to return to its collapsed form.

The result of the series of design decisions presented in this section was materialized into a fully functional prototype that we describe in detail in the next chapter, with particular focus on its behavior.

3.3 Summary

In this section we presented a study that we performed on the content of 127 common blogs, in order to better adjust our solution to real data. We presented the experiment, the obtained results and the corresponding implications on the solution. Then, we explained the decisions that defined the design of the interface of our solution, according to a series of informal usability tests. In this section we showed the evolution of our prototype along time through the introduction of the most relevant low-fidelity prototypes, finishing with the resulting interface included in the final solution.
Chapter 4

Vibe - Visualizing and Exploring Blogs

In the previous chapters we provided the groundwork for our solution, which began with the study on the related research, continuing with the analysis of a set of blogs and the consolidation of the decisions that affected the design of the user interface. Taking that into consideration, now we turn our attention to the resulting Vibe prototype. In this chapter we first present the high level architectural design of Vibe and its main components, also covering the flow of information between them. Then, we describe the server and its operation, finishing with a description of the user interface, detailing its several components and functionality.

4.1 System Architecture and General Information Flow

Vibe follows a Web-based client-server architecture that includes a database for storing the necessary information, for which one of the important aspects is modularity. This translates into a separation between the elements responsible for content extraction and storage and the ones responsible for the presentation, as depicted in Figure 4.1, so that the modification of one of these modules does not greatly affect the others. While the server is responsible for the extraction of the information from blogs and managing content, the client side is exclusively responsible for collecting the input from the user and presenting the content accordingly.

The communication between the client and the server is made though a set of web services and comes down to a few operations, which include determining if a given URL is valid and the blog available on the server, requesting blog information based on user input and informing the server of changes related to read wear. Accessing a blog within the server requires previous URL validation and confirming if the blog is already stored in the database. If that is not the case and the blog cannot be extracted or is not valid, the server informs accordingly. Otherwise, it proceeds with the extraction of the relevant data from the selected blog. Only then the server sends the corresponding blog information as requested by the client. The information sent by
the server includes posts and available filters, for which it needs to receive the current state of
the client, referring to filtering and sorting criteria. As for updating read wear state, whenever
the user opens an entry, its reference is sent by the client to the server, which, in turn, updates
the database with that information.

4.2 Server

The server contains all the functionality of Vibe that has to do with the extraction, processing,
storage and management of blog content, as we can see in Figure 4.2. In particular, after
receiving the request made on the user interface to read a blog, the server begins by determining
if that particular blog URL is valid or not and if it has already been stored in the database,
informing the user interface. If the server does not find the blog in the database, it proceeds
with its extraction, if possible. Blogs that are considered invalid or that cannot be extracted for
some reason force the server to abort the extraction and communicate that occurrence to the
client, thus allowing for the selection of another blog.

Concerning the extraction phase, the first barrier that we faced was the absence of a standard
method to extract the information contained within common blogs among several platforms.
This led us to choose between extensive blog parsing and the search for a better defined API on
one of the platforms. Although it would still be possible to validate our solution by manually
extracting and using the content of a few blogs, we opted for taking advantage of the publicly
available API of Blogger, one of the simplest and most expressive blog platforms currently avail-
able. Still, not every blog hosted by Blogger is compatible with the API, although compatible
blogs are not difficult to find.

The Blogger API allows for accessing the data contained within a blog by providing the list of
its most recent posts, upon receiving a valid blog feed URL. It is possible to define the maximum number of posts requested to the Blogger platform by adding a parameter on the URL, although there is a limit of 500 posts per request, which also limits our solution at this level. From the received list of posts, the server is able to further request the relevant information associated with each post. For each post, the requested information includes its unique reference, title, author, date and time of publication, plain text content, tags/categories and comments. Since each comment is an entry by itself, a similar request is made for each comment to collect the corresponding required information. As far as media information is concerned, we were not able to use the Blogger API, so we opted for an HTML parser\(^1\) for extracting the full body of the post and thereafter the URL of the contained images and videos.

After extraction, the collected information is processed in order to provide more flexibility and better comply with the mechanisms used by the solution. In particular, the date and time of publication is divided into week day, day, month, year, hour and minute of publication, so that it can be used for filtering on the interface. The length of each entry, in words, is calculated here as well, based exclusively in the plain text content, as provided by the Blogger API. If available, video information is also processed at this moment, in order to identify the platform and reference of the video, since three platforms are supported\(^2\) on the final Vibe prototype.

The resulting data is stored into a SQL database\(^3\) using a custom made object-relational

---

\(^1\) Jericho HTML parser. Available in http://jericho.htmlparser.net

\(^2\) Vibe supports videos from YouTube, Vimeo and Sapo, but only YouTube thumbnails are presented.

\(^3\) Supported by Apache Derby. Available in http://db.apache.org/derby
mapping mechanism that translates the objects processed into SQL information.

As far as blog information retrieval is concerned, every request sent by the client to the server includes the set of active filters and the sorting criteria selected by the user on the interface, as previously referred. This information is used by the server to properly prepare the SQL queries that are used to retrieve two types of data, namely the posts and associated information and the sets of filters to be included in the sidebar of the interface. The obtained SQL results are then converted to objects using the Object Factory module, so they can be sent to the client, as requested.

4.3 Client User Interface

The user interaction with the system is made through a Web-based client interface supported by a Web browser. The user interface, along with the associated functionality and communication with the server, was implemented using the Google Web Toolkit\(^4\).

![Figure 4.3: General workflow concerning the Vibe web client.](http://code.google.com/webtoolkit)

As we can see in Figure 4.3, the filtering mechanism relies on the interaction of the user with the Graphical User Interface to operate, in particular with the elements that contain filters. Upon receiving the proper input, the filtering mechanism makes use of the proxies of the services available on server side to request the corresponding blog information. This communication is made asynchronously, so that the user interface is not blocked while waiting for the response of the server. After receiving the response, the requested information is then processed and prepared, in order to update the user interface accordingly. The remaining communications

\(^4\)Available in http://code.google.com/webtoolkit
concerning this module include the request for blog extraction and read wear update, as depicted in Figure 4.3.

The main visual components of the graphical user interface include a navigation window on the left and a sidebar on the right, topped by a title bar that includes the option to open a blog, as we can see in Figure 4.4. While the navigation window is dedicated to presenting the posts within the blog, the sidebar includes additional functionality and overview capability. The user interface can be separated into four main aspects or components, in particular posts and comments, read wear, sidebar and overview, navigation and exploration, as we detail in the following subsections.

![Figure 4.4: General view of the Vibe graphical user interface. On the navigation window (left), we can see three unread posts (the first three) and two read posts. On the sidebar (right), the focus is currently on the tab containing all the tags used within the blog.](image)

### 4.3.1 Posts and Comments

Posts and comments contain the main content within a blog. In Vibe, each post is firstly presented in collapsed form and includes data typically found in the header and footer of posts seen through the usual interfaces, along with additional information. As illustrated by Figure 4.5, on the left of each collapsed post we can see a thumbnail of a contained image, the title, the name of the author that published it, a bar representing the length of the text within the post and the tags associated with the post. On the right, each collapsed post presents the date and time of publication, associated time relevance information and a thumbnail of a contained video, if available within the post. Clicking on the background of the post expands it, giving access to its full content, similarly to what is found on the original interface.

Below the post we can see a bar with a list of squares, each one representing an individual comment associated with the post. When hovered over, each square provides a tooltip that
identifies the corresponding comment in terms of author and publication date and time. Clicking on the background of the comment bar reveals the list of comments associated with the post, all in collapsed form as well, as depicted in Figure 4.6. Collapsed comments include the name of the corresponding author (the commenter), the text length bar and the publication date and time, also supporting expansion on demand. For efficiency, comments can be accessed directly by clicking on the respective square on the comment bar. On the other hand, clicking on the background of posts, comments and comment lists allows for restoring their collapsed form. To the same end, expanded posts also include a button at the bottom, so not to force users into scrolling to the top of the post to collapse it.

In addition to image and video thumbnails, each post includes associated information and navigation arrow buttons, if applicable, for users to sequentially browse through the remaining thumbnails. Clicking on a thumbnail, on the other hand, allows users to access videos and see images in higher resolution, and also browse the remaining similar media within the post. This is done in a new layer that resembles a media gallery, where a focused element is presented on the left of the interface and the remaining similar media elements are presented on the right column, as thumbnails (see Figure 3.5, in the previous chapter). Users can change the focused element by clicking on a thumbnail or by using the buttons on the bottom of the media gallery layer. In order to go back to the typical Vibe interface, users can click on the corresponding option on the bottom of the interface or on the title of the blog, since it remains on another layer.
4.3.2 Read Wear

Read wear information is useful for users to quickly identify entries that have been read in the past. This concept is applied to both posts and comments and is denoted by different background colors on the collapsed counterparts, as we can see in Figure 4.7. While an unread entry uses a brighter background, a read entry is marked with a darker background and lower contrast against the contained text. Reading an entry in Vibe implies its expansion, which is then used to affect visual read wear information. In the case of comments, this information is also applied to the squares representing them on the comment bar, through which darker squares indicate read comments, as opposed to unread comments.

Figure 4.7: Visual comparison between read wear states, concerning a post and comments that have not been read yet (on the left) and after being read (on the right).

4.3.3 Sidebar and Overview

The sidebar on the interface incorporates both complementary functionality and an overview mechanism that also adds to the exploration of the blog, as we explain in the next subsection. The options on the sidebar include a button to clear the filters currently applied to the blog, a set of sorting criteria for the posts, a list of tags, a list of commenters and a calendar. Every option on the sidebar, except for the option to clear all filters, which is available at all times, is presented in a different tab, so that only one category is available at a time.

Selecting the first tab on the sidebar allows users to sort posts by date, popularity, post length, number of images and number of videos present within the post. Sorting can be ascendant or descendant and affects the posts on the navigation window accordingly. The second tab indicates the number of tags used within the blog and clicking on it presents a tag list sorted by the number of available posts that contain each tag. The tag list allows for getting an overview of the blog in terms of topic, since the tag on the top is the most used tag in the blog. A similar concept is applied to the following tab, which indicates the number of commenters in the blog. Expanding this tab presents the list of the commenters that have participated in the blog, ordered by the number of posts each one has commented, so users get an overview of the most active commenters. The overview of the blog is completed with the last tab, the calendar, which gives an overall indication of post publication over time. There, users are able to immediately identify the most active periods of time and get a notion of the most common trends of publication along each year, month or week. As opposed to the information on the remaining tabs, the calendar is chronologically sorted, since this sorting method is more familiar.
4.3.4 Navigation and Exploration

The exploration method in Vibe revolves around the integrated filtering mechanism. This mechanism makes use of the information available on collapsed posts to perform sequential filtering operations, so that users can successively narrow the visible results according to their preference. The available filters for each post include the name of the author, associated tags, week day, day, month, year, hour and minute of publication, and the name of the commenters that commented the corresponding post as well.

These filters provide the advantage of contextual exploration. Let’s consider a scenario where a user browsing a blog gains interest in a post about a particular subject and then wants to read more posts about that subject (see Figure 4.9, top left). In order to do that, the user can click on the corresponding tag, which leads to a new view containing only the posts about that subject (Figure 4.9, top right). Now consider that, as the user reads the resulting posts, he continues to read the associated comments and then becomes interested in the point of view of a particular commenter (Figure 4.9, bottom left). Clicking on the name of the commenter allows for further narrowing the posts, which had already been filtered by tag, so that only the posts associated with that subject and commented by that person are visible (Figure 4.9, bottom right). Depending on what aspect the user wants to explore next, he is able to combine several filters until the desired results are presented. Also note that, to keep the current state visible to the user, the information on the currently selected filters is highlighted at the bottom.
of the interface, serving as a description for the posts displayed at a given moment.

![Image 1](image1.png)  ![Image 2](image2.png)  ![Image 3](image3.png)  ![Image 4](image4.png)

**Figure 4.9:** Here we present a use case scenario involving sequential filtering, using Vibe. After reading a post about ponies (top left), the user selects the corresponding tag, affecting the visible posts (top right). Then, after reading some posts and comments, he decides to filter the posts by commenter (bottom right). The results include only the posts with the tag "ponies" and commented by the selected commenter (bottom right).

The majority of the filters found in posts are replicated in the sidebar and integrated with the overview mechanism that we described on the previous subsection. Like the posts, the available filters on the sidebar are adjusted to reflect the filtering operations, thus preventing users from selecting filters that would lead to empty results. In order to revert the effect of a filter, users can click on the same filter again to turn it off. Considering the same example, if the user wants to see every post commented by a selected commenter, regardless of the subject of the post, he may simply click on the corresponding tag again to deselect it. If the user intends to see all posts instead, he can simultaneously turn off all filters by clicking on the button "Clear filters", on the top of the sidebar.

Imagine now that the user intends to know what is the most common day of the week used by the author of the blog to publish posts. To that end, the user first selects the calendar tab on the sidebar and then identifies the day of the week that indicates having most associated posts. Since clicking on a filter affects the remaining filters, as illustrated in Figure 4.10, the user is also able to progressively traverse years and months in order to identify existing posting patterns. This applies also to tags and commenters on the sidebar.

The posts presented in the navigation window of the interface are divided into pages, each of them containing up to twenty posts. Users are able to scroll through each page and can change pages by using the navigation buttons at the bottom of the navigation window. These buttons allow for sequential browsing and also include shortcuts to the first and last pages of the current view of the blog, thus providing easier access to the first and last posts of the blog, according to
the selected sorting method. In order to navigate through the entries on each page, users can scroll through the available information on the navigation window.

The success of our proposed solution depends on a series of factors. To evaluate our solution we performed a series of tests that we present in the next chapter.

4.4 Summary

In this chapter we presented Vibe, the prototype that is the result of our proposed solution for the problem of the visualization and exploration of individual blogs. We began by presenting the general architecture and information flow, continuing with a description of the server and set of operations it is responsible for. Finally, we presented the user interface and detailed its functionality, divided into a set of aspects and components, in particular posts and comments, read wear, sidebar and overview, navigation and exploration.
Chapter 5

Evaluation

Evaluating the success of our solution requires the involvement of the users. To that end, we identified the need for two series of usability tests, corresponding to formative and summative user evaluations. These tests are detailed in the next sections and include the description of the prototypes used, the users involved and the procedures followed, continuing with the presentation of the results obtained and the corresponding discussions.

5.1 Preliminary Formative Evaluation with Users

The main objective of formative usability tests is to evaluate the primary concepts behind the solution, using a prototype with limited functionality, in order to better adjust the path to take during the development phase.

5.1.1 Prototype

The prototype used in this series of usability tests, which we named alpha prototype, contained only a portion of the functionality of the final solution and was primarily focused on blog navigation. The main features of the alpha prototype included blog extraction, post and comment metadata presentation, post filtering, image thumbnail navigation and read wear.

When compared with the final solution, which was described in the previous chapter, the alpha prototype had a different presentation and color scheme overall and did not include the sidebar, as we can see in Figure 5.1. As far as content is concerned, the alpha prototype still did not support video thumbnails and did not include access to the expanded version of images either. There was no reference to time relevance information on each post and the same goes for the information on the current filtering criteria at the bottom of the interface. In turn, the content presented in each expanded post was limited to plain text.
5.1.2 Users

The users that participated in this test were occasional or regular blog users, two of which had experience with using aggregators to view blogs. For these tests we asked for the participation of five users, of which three were male. Their ages ranged between 19 and 35 years old.

5.1.3 Procedure

In these tests we concentrated on addressing some major issues regarding the usability, utility and completeness of the interface. Thus, we made these tests more informal, in order to put the users as comfortable as possible. To the same end, each test was performed at a familiar place to the user.

Each test consisted on presenting a previously processed blog to the user, using the alpha prototype. The user was then asked to use the prototype freely during approximately 10 minutes and to emit opinions related to its functionality and design. We also took note of the errors made during the test, which essentially consisted in mismatches between the behavior of the interface and the behavior expected by the user.

5.1.4 Results

The major issues that we identified, concerning the user interface of the alpha prototype, had to do with its purpose and scope. For example, one user asked if the tool was supposed to provide only individual blog visualization or also aggregation capability. A related aspect had to do
with the functionality expected from the tool. One user emphasized the need for a popularity indicator for each post, both local, related to the number of comments, and global, considering, for example, its impact on the members of social networks. The same user said that it would be nice to have a list of references for each post and also the ability to know in advance the type of media contained by the post, whether it is mainly text, images or videos. He also said that he would prefer to see more than one thumbnail immediately visible in each collapsed post.

Another referred aspect was the identification of the main topic of the blog. Some users felt the need for a global list of the most used tags in the blog or even a list of the most used words in every post. The same applied to the most active commenters in the blog. Regarding the post and comment bars, most of the users asked if its length was absolute or relative and one of them even suggested that length levels should be used instead. Most of the users understood the filtering mechanism, but felt the need for a button that cleared all the filters. As for the images, some users attempted to access an expanded version of the thumbnail by clicking on it. Considering the time of publication, a user said that it would be more important for him to have time relevance references for locating posts in time, such as "a week ago" or "two years ago", as compared with the actual date and time.

As far as the visual aspect of the interface is concerned, most of the users liked its color scheme, but one of them considered it somewhat overloaded, referring to the excessive use of borders, and appealed for a cleaner and more modern interface. With the same intent, this user also indicated that the filter buttons could be replaced by links. Other notes related to the visual impact of read wear, where one user said that the difference between read and unread posts wasn’t visible enough and another user said that the corresponding background colors were swapped. The lack of text formatting as compared with the original post was also referred by some users as a negative aspect.

5.1.5 Discussion

The results obtained with this series of usability tests affected our prototype at different levels. Some of the identified problems had already been foreseen, such as the lack of tag and commenter lists and the need for the expansion of image thumbnails, along with a button to clear all filters. These requirements were considered in the most recent low-fidelity prototype and also applied afterwards to the prototype itself. On the other hand, aspects such as the inclusion of a popularity meter or blog aggregation functionality required some clarifications on the interface.

Hopefully, the inclusion of the sidebar would cover the majority of the reported issues. In particular, the identification of popular posts was solved by associating the concept of popularity with the number of comments per post and by adding the corresponding option as a sorting method. However, the inclusion of a global popularity meter was discarded, since it would fall out of the scope of our project. The same went for the inclusion of external references and most used words per post.

With regard to presentation, we changed the appearance of filters to better comply with the
low-fidelity prototype, so they are not mistaken for regular buttons. Changing them to common HTML links would also prevent them from conveying the intended status information and could even mislead users. Furthermore, we also removed the "view all" tag filter, in order to remove unnecessary redundancy, since only one tag is allowed at the same time and clicking on the active tag filter has the same effect.

The presentation of only one thumbnail at a time in each post was kept, in order to avoid conflicts with the remaining information on the collapsed post and to maintain consistency. Regardless of this decision, we opted to add another thumbnail on the opposite side of the collapsed post to indicate the presence of videos within the post, if available. We also added time relevance information to the metadata in the collapsed post, placing it below the time filters. The length bar was kept as originally intended, since its main objective is to immediately convey the length of the post at a glance.

Additionally, we made some updates to the visual aspect of the interface, which include added emphasis to the read wear indication on each post and comment. We also followed the suggestion of the users on presenting the originally formatted full post content after expansion of collapsed posts, so not to disrupt the original storytelling.

The changes described here were included in the prototype used for the final series of usability tests, which is described in the next section.

5.2 Final User Evaluation

The final series of user evaluation sessions serve the purpose of measuring the extent to which the concepts behind the solution are able to fulfil the objectives that were set in the beginning. This evaluation aims for measuring both the performance of our prototype in the visualization and exploration of individual blogs and the satisfaction of the users while using it, as compared with other interfaces. Thus, we decided to perform a comparative evaluation between the original blog interface, our prototype and an aggregator, Google Reader, based on the same exploratory tasks.

5.2.1 Interfaces Tested

Considering that we wanted to compare three different interfaces in the visualization and exploration of the same blog, it was of central importance that the selected blog fulfilled a predetermined set of requirements. According to our objectives, the blog should contain several posts published over time (i.e. at least during some months) and also include images, tags, comments and videos. Although not as crucial, we also decided that the blog should be written in Portuguese and aimed at a broader audience, in order to decrease the effort made by the users while performing the tasks.

During the selection of the blog, we always kept in mind that we were choosing one of the test
Figure 5.2: The selected blog, as seen on the original interface, with the blog archive and a portion of the tag list visible on the right.

interfaces simultaneously. Therefore, in addition to the previously enumerated characteristics, we chose a common blog\(^1\) that includes some structural navigation mechanisms\(^2\), such as an archive of posts, organized into years and months, and a list of the tags used in the blog, both available in the right column of the interface, as can be seen on Figure 5.2. It should also be referred that the extraction method used on our prototype had some influence in the selection of the blog as well, since it was an obvious requirement for it to be able to fully extract the relevant contained information.

\(^{1}\)Available in http://tostoescadecasa.blogspot.com

\(^{2}\)Please refer to the description of the usual blog interface in subsection 2.2.1.

Figure 5.3: The Google Reader interface with the sidebar collapsed, as used during the testing phase.

As previously indicated, we opted to include an aggregator among the tested interfaces, for which we chose Google Reader, since it is one of the interfaces that most influenced the design.
of our prototype and it is also easily accessible. As opposed to its original purpose, the Reader was used to visualize and explore a single blog, and thus its sidebar was collapsed, as we can see in Figure 5.3, in order to avoid user errors, since its functionality is limited to the management of multiple blogs\(^3\).

![Image](image.png)

Figure 5.4: The initial view of the selected blog on our prototype.

The Vibe prototype used in these tests corresponds, in essence, to the final prototype described in the previous chapter. It includes all the functionality that composes our proposed solution, as we can see on Figure 5.4. For a complete description on the prototype, please refer to chapter 4.

### 5.2.2 Users

The final series of usability tests involved thirty users, of which seventeen were male. The users were aged between 14 and 64 years old and had 31.5 years on average.

In order to participate in the final usability sessions, it was set as prerequisite that the users should at least be familiar with Web interfaces. Still, every participant revealed to usually follow at least one blog. Ten users indicated that they usually left comments on posts and six users actively participated in blogs as authors. The majority of the users (19) spent, on average, less than one hour per day in blog activities, while ten users said they usually took between 1 and 3 hours on the same activities. The remaining user reported spending 4 to 8 hours per day reading and participating on blogs. Of the thirty users, only one indicated he used exclusively an aggregator to access blogs. Six other users shared his preference in this matter, but also used the original interfaces, while most of the users (23) accessed blogs only using the latter.

In order to comply with the testing procedure, the users were divided into three groups, G1, G2 and G3. G1 and G3 included four females each, while G2 included the remaining five female users. The average of ages for G1, G2 and G3 was 33, 32.2 and 29.3 years old, respectively.

\(^3\)Please refer to section 2.3 for a more complete description on Google Reader.
In terms of experience using blogs, users were also approximately evenly distributed between different groups. G1 tested the original blog interface, G2 tested Google Reader and G3 tested Vibe.

5.2.3 Procedure

As previously referred, the final series of usability tests involved thirty users, divided into three groups of ten people, and each group was given a different interface to perform the tests on.

Each test was performed on a familiar place to the user, using the same computer. For this, we resorted to a recent 14" laptop equipped with an external mouse, to facilitate both mobility and interaction. The tested interfaces were accessed through the same Web browser\textsuperscript{4}, in full screen mode, using 1366x768 pixel resolution.

The test itself was divided into three parts. First, we gave a brief description of the experiment and allowed the users to explore another blog during approximately 5 minutes\textsuperscript{5}. This blog shared some characteristics with the blog where users performed the tasks, in the second part of the test, using the assigned interface. Then, the users were given a set of tasks to perform on the previously selected blog, using the same interface. A time limit of 3 minutes was set for each task. The tasks were divided into six categories, as we explain in the next subsection, each of which was recorded separately, using screen capturing software. Finally, the users completed a questionnaire including both general questions and questions about the interface used. The users performing the tasks on our prototype were also asked to answer a few questions about it.

As a reference, the complete test script used during the tests is available in annex A, at the end of this document.

5.2.4 Tasks and Metrics

To evaluate the performance and usability of the three interfaces, we defined a set of sixteen tasks involving navigation, exploration and identification of information within the selected blog. Most of these tasks were inspired in the tasks used by Indratmo, one of the authors of iBlogVis [Indratmo 08], in a similar study performed for his PhD thesis [Indratmo 10], which were adapted to comply with our objectives. To the same end, we also defined an additional set of revisitation tasks.

The resulting sixteen tasks were then divided into six categories and at the end of each category the users were asked to indicate the associated level of easiness on a scale from 1 (not easy at all) to 5 (very easy). Below we present the translated tasks, organized by category\textsuperscript{6}, along with the associated easiness questions:

\textsuperscript{4}Firefox 6. Available in http://www.mozilla.org/firefox
\textsuperscript{5}Available in http://raizesantenas.blogspot.com
\textsuperscript{6}Category titles were omitted in the original test scripts so as not to influence users.
Time

• In what date and time did the author publish his first post?

• In what month and year did the author publish more posts?

• What was the most frequent day of the week used by the author, during the last year, to publish posts?

Understanding how posts were published in time was...

Not easy at all □  Not easy □  Normal □  Easy □  Very easy □

Topic

• What is the most used tag in the blog?

• What were the two most used tags during the last year?

Understanding what was the main topic of the blog was...

Not easy at all □  Not easy □  Normal □  Easy □  Very easy □

Commenters

• Who is the commenter that commented more posts in the blog?

• Who is the commenter that commented more posts during this year?

Identifying regular commenters was...

Not easy at all □  Not easy □  Normal □  Easy □  Very easy □

Popularity

• What is the most popular post in the blog (with most comments)?

• What is the most popular post published in the last year?

Identifying the most popular posts in the blog was...

Not easy at all □  Not easy □  Normal □  Easy □  Very easy □

Finding entries

• Access/open the most recent post tagged with "dinheiro".

• Access/open the post about "caldeirada de peixe".
• Access/open the oldest post published in September 2010.

• Access/open the post tagged with "beleza" and published in August 2010, on a Wednesday.

Finding posts using their associated information was...

Not easy at all □   Not easy □   Normal □   Easy □   Very easy □

**Revisitation**

• Access/open the same post found in the last task.

Please check all that apply. I found the previously read post by...

□ applying the same method used in the last task.

□ recognizing an image in the post.

□ recognizing a video in the post.

□ recognizing the title of the post.

□ recognizing the date of the post.

□ using other method: ____________________________

• Access/open the most recent read post in the blog.

• Access/open the most recent unread post in the blog.

Identifying posts already read in the past was...

Not easy at all □   Not easy □   Normal □   Easy □   Very easy □

During the experiment we used screen capturing software to record the actions performed by each user on the screen, for each group of tasks. This information was used to collect the times of execution for each task. Besides this, as previously indicated, we also measured the level of easiness for each group of tasks, on each interface.

At the end of each usability test session we measured the level of usability of the corresponding interface, according to the evaluation made by the user. For this, we used the System Usability Scale [Brooke 96] (SUS) by John Brooke, which served as a standard comparison method between the three interfaces. Users were also asked to indicate what they liked best and what they liked least in the tested interface. Additionally, we collected the levels of usefulness of some of the most important features included in our solution, as evaluated by the ten users that performed the test on it.
5.2.5 Results

The main comparative results obtained during these usability tests are summarized in Tables 5.1, 5.2, 5.3, 5.4 and Figures 5.5 and 5.6. The most important results collected refer to the time spent during the execution of each task, the level of difficulty faced while performing each set of tasks, the usability score given to the interface and the usefulness of the most important features on the Vibe prototype.

Beginning with the time of execution for each task, we must note that only the cases where the user gave the right answer, in time, were considered for this purpose. There are, however, some situations where more than one answer could be accepted, particularly for revisitation tasks, depending on previous history and the interface used. These situations were verified using the corresponding video recording, thus confirming its usefulness. As a reference, users were able to complete 98% of the tasks using Vibe, 63% on the original interface and 34% on Google Reader, on average. Table 5.1 summarizes the time spent by the users to accomplish each task, on each interface, including the average and standard deviation values. At the end of the table we added the average task completion time for each interface.

From the analysis of Table 5.1, the first evidence that catches our attention is the disparity between the results achieved by different users on the same interface. This is denoted by the high standard deviation values obtained for each task, arising as a result of the participation of only ten users testing each solution. Although a larger group of users could have increased the relevance of the results, the validity of the obtained data cannot be denied and is considered as means of comparison between the performance of the three interfaces, for each of the presented tasks, as follows.

The first task in the test required the identification of the first post in the blog, for which our prototype had the lowest execution times among the three solutions. This confirms the usefulness and flexibility of the interface on providing both an appropriate time sorting method, on the side bar of the interface, and an alternative method for immediately navigating to the last page of the blog, if the original reverse chronological order of posts is maintained. The usual blog interface, on the other hand, reported the highest times, indicating the difficulty that users had in navigating through the several pages of the blog, often ending up looking for the post on the blog archive. On Google Reader, the obtained times reflect the need that users had to scroll down through an undisclosed amount of collapsed posts, due to the used dynamic loading mechanism. However, only two people found the right answer to this question on the Reader, since each collapsed post immediately presented the date of receipt instead of the date of publication. Although these dates usually overlap, this was not the case. In order to see the date and time of publication, users were required to hover over the displayed reception date.

On the second task, which asked for the month and year that have the largest amount of posts published in the blog, the usual interface took the lead, given the immediate presentation of the corresponding number of posts next to each year and month on the blog archive. In turn, Google Reader forced users to count the posts on the presented list, in order to collect the
<table>
<thead>
<tr>
<th>Tasks</th>
<th>Usual Interface</th>
<th>Google Reader</th>
<th>Vibe</th>
</tr>
</thead>
<tbody>
<tr>
<td>In what date and time did the author publish his first post?</td>
<td>115</td>
<td>89</td>
<td>36</td>
</tr>
<tr>
<td>In what month and year did the author publish more posts?</td>
<td>51</td>
<td>125</td>
<td>77</td>
</tr>
<tr>
<td>What was the most frequent day of the week used by the author, during the last year, to publish posts?</td>
<td>-</td>
<td>134</td>
<td>34</td>
</tr>
<tr>
<td>What is the most used tag in the blog?</td>
<td>29</td>
<td>-</td>
<td>22</td>
</tr>
<tr>
<td>What were the two most used tags during the last year?</td>
<td>68</td>
<td>-</td>
<td>40</td>
</tr>
<tr>
<td>Who is the commenter that commented more posts in the blog?</td>
<td>-</td>
<td>-</td>
<td>17</td>
</tr>
<tr>
<td>Who is the commenter that commented more posts during this year?</td>
<td>-</td>
<td>-</td>
<td>19</td>
</tr>
<tr>
<td>What is the most popular post in the blog (with most comments)?</td>
<td>136</td>
<td>-</td>
<td>30</td>
</tr>
<tr>
<td>What is the most popular post published in the last year?</td>
<td>73</td>
<td>-</td>
<td>14</td>
</tr>
<tr>
<td>Access/open the most recent post tagged with &quot;dinheiro&quot;.</td>
<td>30</td>
<td>75</td>
<td>39</td>
</tr>
<tr>
<td>Access/open the post about &quot;caldeirada de peixe&quot;.</td>
<td>60</td>
<td>38</td>
<td>110</td>
</tr>
<tr>
<td>Access/open the oldest post published in September 2010.</td>
<td>37</td>
<td>34</td>
<td>39</td>
</tr>
<tr>
<td>Access/open the post tagged with &quot;beleza&quot; and published in August 2010, on a Wednesday.</td>
<td>58</td>
<td>72</td>
<td>37</td>
</tr>
<tr>
<td>Access/open the same post found in the last task.</td>
<td>28</td>
<td>23</td>
<td>26</td>
</tr>
<tr>
<td>Access/open the most recent read post in the blog.</td>
<td>41</td>
<td>43</td>
<td>73</td>
</tr>
<tr>
<td>Access/open the most recent unread post in the blog.</td>
<td>23</td>
<td>19</td>
<td>10</td>
</tr>
<tr>
<td>Averages</td>
<td>58</td>
<td>64</td>
<td>39</td>
</tr>
</tbody>
</table>

Table 5.1: Comparison table including the average time of execution for each task, on each interface, and the corresponding standard deviation (S.D.), in seconds.

required information, thus justifying the highest times obtained. The times obtained with Vibe are linked with the fact that people were not used to using calendars the way we present them in our solution, which leads to longer adaptation times.

The last time related task could only be performed within the given time using our solution, with the exception of one user testing the Reader. This user was able to guess the day of the week that was most frequently used by the author to publish posts by taking only the presented dates into account, despite the absence of references to the days of the week. The users testing the usual interface concluded that it would take too long to traverse the whole blog trying to
identify the required information, only available on each post.

The identification of the most used tag on the blog was only possible with the usual interface and with our solution, since the Reader does not support the presentation of tags. Given this, Vibe achieved slightly better results, which supports our decision to sort tags by the number of associated posts instead of alphabetically. The following task, however, required the combination of filters, which is not supported by the usual interface, and therefore could only be accomplished on our prototype. Still, one user was able to guess the two most used tags during the previous year of the blog after looking to the tag list on the usual interface.

Similarly to the previous task, the tasks related to the identification of commenters and popular posts could only be accomplished in time using our solution, since the Reader does not support comments and the usual interface requires users to visit all the posts in the blog to meet that end. Once again, one user testing the usual interface proved to be an exception and managed to identify the most popular posts after quickly navigating through the whole blog.

The tasks for finding entries using information within the posts returned mixed results across the three interfaces. The first task in this category asked users to find the most recent post with a specific tag. While only two users on the Reader were able to find the right post by guessing, based on the text of the corresponding post, the users on the other two interfaces accomplished this task with similar times. However, the usual interface obtained slightly better times, benefiting from the alphabetical order of the tag list as opposed to the most used tags order applied to our prototype.

The following task was introduced in order to evaluate the impact of a search by word mechanism in the exploration of a blog. As expected, both the usual interface and the Reader fared better than our solution in this task, with the Reader taking a solid lead. This can be explained by the fact that the Reader maintains its search functionality visible at all times, as opposed to the Blogger search bar, which is only available at the top of the usual interface, as seen on Figure 5.2, and disappears when scrolling down. Still, our prototype performed well on this task, given the absence of a search by word mechanism, with seven out of ten users identifying the right post. Most of these users found the post by relating the associated tags and thumbnail to the given task.

Finding the oldest post published on a given month returned quite similar execution times on the three interfaces, which indicates a balance between the accessibility of the blog archive on the usual interface, the visibility of dates on the Reader and the usefulness of the calendar on Vibe. It is plausible, however, that the extended use of the calendar would have positively influenced execution times on our solution.

The last task in this category required the combination of different bits of information in order to find the desired post. The resulting times prove the efficiency of the filtering mechanism used on our solution, which clearly benefits from a small learning curve drawn along the previous tasks. On the other hand, Google Reader was hindered by the lack of tag support, while the usual interface showed the impact of a cumbersome navigation method.
The last set of tasks had to do with revisitation, the first of which asked users to find the post identified in the last task once again. The resulting times obtained on this task are very similar across the three interfaces and somewhat inconclusive. A possible explanation to this may have to do with the interference of short-term memory in the recall of the information associated with the recently found post. It is possible that keeping the two corresponding tasks apart could have led to very different results between the three interfaces. Actually, this can be confirmed by analyzing the answers given for this task, which can be seen on Table 5.2. Unsurprisingly, most of the users reported having applied the same method used in the previous task, with the recognition of the date of the post appearing as second most common choice. However, the recognition of the image in the post had a significant contribution to the identification of the post both on the usual interface and on our prototype.

In a similar manner, the last two tasks have apparently contradicting resulting times. While on the identification of the most recent read post in the blog, our prototype is behind the other two interfaces, accessing the most recent unread post revealed to be faster. This may be explained by several factors, one of which has to do with the learning time related to this commonly overlooked aspect. Perhaps having more influence on the results is the fact that some users felt compelled to open or access the most recent post of the blog while navigating through it, during the previous tasks. This led to indicating the first of the presented posts as the obvious answer for the most recent read post.

Considering the previous results, the average time to complete a task is once again lower on Vibe, as compared with the other two interfaces. This means that, for relevant blog exploration tasks, users should be able to take less time in our solution.

Referring to the revisitation task "Access/open the same post found in the last task", the majority of the users replied having used the same method as before, with few users reporting otherwise. Also, since we allowed more than one answer, many users indicated the recognition of the title, date or image of the post. We took notice that two users resorted to methods for revisiting the post that had not been predicted. On the usual interface the user performed a word search on the title of the post, while the other user, on the Reader, recognized the first words of the post in its collapsed counterpart. The answers given for this task are summarized in Table 5.2.

<table>
<thead>
<tr>
<th>I found the previously read post by...</th>
<th>Usual Interface</th>
<th>Google Reader</th>
<th>Vibe</th>
</tr>
</thead>
<tbody>
<tr>
<td>applying the same method used in the last task.</td>
<td>8</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>recognizing an image in the post.</td>
<td>5</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>recognizing a video in the post.</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>recognizing the title of the post.</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>recognizing the date of the post.</td>
<td>6</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>using other method.</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.2: Methods used in the post revisitation task, where users were asked to find a previously visited post, on the three interfaces. Please note that users could check more than one answer.

According to what was previously referred, each user was also asked to indicate how easy
it was to accomplish each set of tasks on the tested interface. The corresponding results are summarized in Table 5.3.

<table>
<thead>
<tr>
<th>Task</th>
<th>Usual Interface</th>
<th>Google Reader</th>
<th>Vibe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Not easy at all</td>
<td>Normal</td>
<td>Easy</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>Topic</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Not easy at all</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Normal</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Easy</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>Commenters</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Not easy at all</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Normal</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Easy</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Popularity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Not easy at all</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Normal</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Easy</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Finding entries</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Not easy at all</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Normal</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Easy</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Revisitation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Not easy at all</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Normal</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Easy</td>
<td>0</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 5.3: Cumulative easiness scores for each group of tasks on the three interfaces.

As we can see in Table 5.3, most of the users testing our prototype reported having faced less difficulty in performing the given tasks, as compared with the other two solutions. In particular, the majority of the users testing Vibe found the tasks related to topic, commenters, post popularity, finding entries and revisitation to be easy or very easy. Tasks related to time were classified as normal or easy to accomplish.

Figure 5.5: Easiness average for each group of tasks, on each interface, as rated by the users on a scale from 1 (not easy at all) to 5 (very easy).

As for the usual interface, most of the users indicated time related tasks as normal or not
easy to complete, while tasks related to topic, finding entries and revisitation were classified as normal or easy. On the other hand, tasks related to identifying commenters and popular posts were classified by the same users as not easy at all or at least not easy to accomplish. At this level, Google Reader obtained mixed results, with most of the classifications ranging almost equally from not easy at all to easy. The only exceptions are the classifications given to the sets of tasks related to commenters and popularity, which were classified as not easy at all by most users on the Reader. The easiness average for each group of tasks, on each interface is presented on the bar chart in Figure 5.5.

At the end of each test session, the user was asked to evaluate the interface tested, for which we used the System Usability Scale. The resulting average score for each interface is presented in Figure 5.6, including the corresponding standard deviation values.

![Figure 5.6: Average usability scores for each of the tested interfaces, and corresponding standard deviation, as evaluated by the users, using the System Usability Scale [Brooke 96].](image)

As we can conclude from the obtained values, the average score given to Vibe clearly shows that our solution is the best suited to explore individual blogs of the three interfaces tested, which is confirmed by the low standard deviation value. The usual interface received the lowest score, justified by the cumbersome navigation and aging navigation method with lack of a complete overview mechanism. Google Reader fared better here than the usual interface, which justifies its wide use as an aggregator, to which the presentation of several posts per screen combined with the collapsing mechanism appears to have contributed greatly. When comparing these results, the probability that the means of the corresponding populations are equal, using Student’s t-test, is given by $p<0.02$ for Vibe versus the original interface, $p<0.03$ for Vibe versus Google Reader and $p<0.05$ for Google Reader versus the original interface, which serves as a confirmation for the success of our solution as compared to the other two, in terms of usability.

Regarding the characteristics that drew the attention of the users, on the usual blog interface the users liked the calendar and the tag list the most, but often complained about the navigation system used and the lack of a more flexible filtering mechanism or advanced search functionality.

On the Reader, the users highlighted the ability to see many posts on the same screen and also the ability to expand them. Characteristics such as the accessibility and simplicity of the
interface, along with the presentation of the first words on each post, are also referred to as positive aspects. On the other hand, users missed the presence of tags and comments the most.

The most liked feature on our solution is definitely the filtering mechanism, with the calendar appearing as the second most liked feature. Some users also consider the presentation of tags, the access to an image gallery for each post and the sorting mechanism as important functionality. The most negative aspect on the interface is the lack of search by word functionality. One user also referred that he would have liked the possibility to apply other sorting methods to the tag and commenter lists, while another user didn’t like how comments were presented, in reference to the need to expand each comment individually.

Also related to these answers is the classification of some features of our prototype, for which the corresponding acceptance was considered critical to the success of our solution. These results are presented on Table 5.4.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Strongly disagree</th>
<th>Disagree</th>
<th>Indifferent</th>
<th>Agree</th>
<th>Strongly agree</th>
<th>Average (1-5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Images and videos help me find posts that I have read in the past.</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>6</td>
<td>2</td>
<td>4.0</td>
</tr>
<tr>
<td>Knowing the length of a post before reading it is useful.</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>3.2</td>
</tr>
<tr>
<td>The filtering mechanism is useful for exploring the blog.</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>6</td>
<td>4.6</td>
</tr>
<tr>
<td>The expansion/collapsing mechanism is useful for navigating through posts and comments.</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>1</td>
<td>3.8</td>
</tr>
<tr>
<td>Having information about the comments each post received is useful.</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Having direct access to each comment is useful.</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>3.7</td>
</tr>
<tr>
<td>Read and unread posts and comments are easily distinguishable.</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>4.1</td>
</tr>
<tr>
<td>The functionality on the side bar is useful for getting a global view of the blog.</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>6</td>
<td>4.6</td>
</tr>
<tr>
<td>The calendar allows for understanding the blog dynamics through time.</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>7</td>
<td>4.5</td>
</tr>
</tbody>
</table>

Table 5.4: This table summarizes the ratings given by users to some of the features included in our prototype, on a scale from 1 (strongly disagree) to 5 (strongly agree).
As we can see in Table 5.4, most of the users agreed with the majority of the concepts applied to Vibe, with the exception of the length of the post, since most of the users felt indifferent towards its inclusion. Confirming the answers given previously, users liked the filtering mechanism best, with the functionality on the sidebar and calendar achieving similar preference.

### 5.2.6 Discussion

Here we summarize and discuss the main results obtained during the final series of usability tests, as presented in the previous subsection.

Beginning with the execution times, Vibe allowed faster completion on 10 of the 16 tasks, on average, corresponding to 62.5%, as compared to the other two interfaces. Even though Vibe was slower on the remaining tasks, this is justified by the influence of the learning time associated with the filtering mechanism, upon careful analysis of the evolution during each test session, as most of the subsequent related tasks revealed significantly lower times. Despite these slightly disappointing results, in light of our expectations, the main objective set in the beginning was successfully achieved.

The remaining results reflect the clear intention of users in adopting a solution that shares similar characteristics and functionality to those applied to Vibe. This can be inferred through the association between the low difficulty given to our solution during task completion and the high usability scores attained. As opposed, the somewhat contradicting low difficulty and also lower usability scores given to the usual interface indicate that, in spite of being familiar with common blogs, users are not satisfied with their usability.

As a side note, and although this detail was not included in the test protocol, every user that did not evaluate our solution was given the opportunity to use Vibe freely on the same blog after the corresponding test session. The majority of these users praised the interface of Vibe, as opposed to the one they had tested, classifying it as a sort of breath of fresh air, after experiencing some frustration during the execution of the tasks. A few users even joked with demands to repeat the test, but using our prototype instead.

In addition to the highly positive reactions from all the users that had contact with our solution, most of the users testing our prototype agreed with the concepts that serve as base for our solution, highlighting the filtering and collapsing mechanisms, along with side bar functionality, in particular the calendar, from the analysis of Table 5.4. The only exception at this level has to do with the presentation of the length of the entries, which received some indifferent feedback from the users.

In spite of the presence of a few inconclusive results, as referred throughout our discussion about the final series of evaluation sessions, the overall outcome of the evaluation performed gives a clear indication that our work is on the right track towards the better visualization and exploration of individual blogs. However, there is still much that can be improved in future work, as we further discuss in the next chapter.
5.3 Summary

In this chapter we covered the evaluation method used to validate our solution, in light of our objectives. We began by describing the first series of usability tests concerning our prototype, which were formative tests, characterizing the prototype, the users and the procedure involved. The performed tests returned a set of results that were also described, along with the corresponding discussion of the most relevant issues. Then, we continued with the description of the final series of usability tests, highlighting the interfaces used, the users involved and also the details of the procedure. These final tests returned a set of important and highly positive results, as analyzed and discussed at the end of the chapter.
Chapter 6

Conclusions and Future Work

In this chapter we present and discuss the final conclusions attained and the contributions of our work, reflecting on how new related problems can be approached in the future.

6.1 Summary of the Dissertation

In this dissertation we proposed a new solution for the visualization and exploration of individual blogs. In chapter 2, we started by studying the work related with the visualization of blogs, in order to better understand the context where our work is inserted. Then, we continued with the study of the usual blog interface, along with novel interfaces that have been developed in recent years. This study was finalized with a comparative analysis intended to point out both their positive and negative aspects, from which we realized that none of the analyzed solutions was able to deliver a fairly complete and flexible solution for the visualization and exploration of individual blogs.

In chapter 3 we introduced a study on blog content that we used to support our proposed solution, based on real data. Further in the same chapter we explained the design decisions that determined the elements and characteristics of the user interface of Vibe, supported by the evolution of the associated concepts applied to a series of low-fidelity prototypes, until the final prototype.

Then, in chapter 4, we presented the prototype that we developed to support our solution, covering its general architecture and communication between its components, in particular the Web client and the server. Then, we got into more detail about the architecture of both the server and the client, further describing the corresponding flows of information. Finally, we focused on the client user interface and described its main components and aspects from the perspective of their behavior, highlighting the presentation of posts and comments, the impact of read wear, the functionality on the sidebar and the flexibility of the integrated navigation and exploration mechanism.
In chapter 5 we described the experimental evaluation to which we submitted our functional prototype, which covered two phases. The first phase comprised the informal evaluation, with users, of an early version of the prototype, which included only a portion of the intended functionality. The results obtained from this evaluation series enabled the validation of some concepts and the adjustment of others for inclusion in the final prototype. The second evaluation phase corresponded to the final usability test series. This series involved 30 users in the comparison of three interfaces used to explore the same blog. The obtained results, in particular the usability scores, user opinion and task execution times, allow us to demonstrate the success of our solution, as compared with the other two interfaces.

6.2 Final Conclusions and Contributions

With our work we have obtained a solution that improves the visualization and exploration of individual blogs, in particular of those receiving frequent updates and that include images and comments.

To that end, our solution provides a better navigation combined with a global filtering mechanism that makes use of the information associated with each post, in particular author, publishing time and date, tags and commenters. This mechanism allows for the contextual exploration of blogs, since users can filter posts according to the information that is immediately visible, but also has support for filtering upon overview. As such, a combined overview mechanism is provided on the sidebar, on the right of the interface, which aggregates the most relevant filters, in separate lists. The filters available on the sidebar are augmented with the number of associated posts, so that users can easily identify the most used tags, active commenters and time related publishing trends, thus acquiring a better understanding of the content and dynamics of the blog.

Another important contribution of our work is the easier identification of the content within posts, for both new and returning blog visitors. While the presence of image and video thumbnails allows visitors to get a good notion of what each post is about without the need for reading it, returning visitors can also count with the support of read wear. Read wear highlights unread posts and comments, as compared with those which were read in the past, therefore aiding users in locating themselves while exploring a blog with the promotion of recognition instead of recall.

The results obtained during our experimental evaluation, which were presented in chapter 5, allow us to justify our conclusions. Upon analyzing the outcome of the evaluation phase we are able to conclude that users like our solution, as compared with common interfaces that have similar purpose. That is confirmed by the high usability scores given to Vibe during the test sessions, along with the overall positive opinions received. Vibe also manages to enable faster completion of common blog exploratory tasks, on average, against the usual blog interface and Google Reader.

In terms of concepts applied to the solution, the opinions given by the users and described in
the results also contribute to confirming their usefulness, specially for the filtering and overview mechanisms. On the other hand, the presentation of the length of each post is an aspect to be revised, since the lack of interest revealed by the users on this feature does not justify the space it occupies on collapsed posts.

Taking into consideration the related work once again, we are now able to compare Vibe with the most relevant of the previously analyzed solutions.

<table>
<thead>
<tr>
<th></th>
<th>Usual Blog</th>
<th>Blogger Views</th>
<th>iBlogVis [Indratmo 08]</th>
<th>PostConnect [Bross 10]</th>
<th>Reader</th>
<th>Vibe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browser-based</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Comments</td>
<td>Yes</td>
<td>Yes</td>
<td>Limited</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Familiar GUI</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Flexible Navigation</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Images</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Overview</td>
<td>Limited</td>
<td>Limited</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Read Wear</td>
<td>Limited</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Limited</td>
<td>Yes</td>
</tr>
<tr>
<td>Site Statistics</td>
<td>Limited</td>
<td>No</td>
<td>Yes</td>
<td>Limited</td>
<td>Limited</td>
<td>Yes</td>
</tr>
<tr>
<td>Tags</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Thumbnails</td>
<td>n/a</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>User Statistics</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Word Search</td>
<td>Limited</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 6.1: Comparison table between our solution and the five most significant solutions described in chapter 2.

As we can see in Table 6.1, our solution attains all but two aspects that we defined as being important for a solution aiming to improve the visualization and interactive exploration of individual blogs. The absence of user statistics and word search in our solution is due to the fact that these fall out of the scope of our work, requiring further study for proper integration of the corresponding functionality. Still, Vibe manages to surpass every other solution covered in the related work in terms of important features.

### 6.3 Future Work

Considering the focus of our research, as described in this report, future work related to Vibe may involve a review of its general user interface, in order to further enhance the experience provided to users exploring individual blogs. In particular, the use of new technologies such as HTML5 may facilitate the creation of a more appealing user interface, taking as an example the new blog templates presented by Blogger that we analyzed in chapter 2. Furthermore, and based on our conclusions, removing the presentation of the length of each entry or replacing it with an indication that takes less space or with its first words, could also work towards the interest of the users. In fact, this may leverage the addition of new facets to the interface, so to complement the existing functionality.

On the technical aspect, other improvements may be done in terms of compatibility and
efficiency of the information extraction and communication mechanisms. This may involve adding support to other blog platforms, as well as providing a means to circumvent the post limit imposed by the Blogger platform for each feed request. The integration of a more dynamic information loading mechanism on the interface would also be a welcome addition. At this level, a loading mechanism similar to the one used in Google Reader could replace the page navigation used at the bottom of the interface of the final Vibe prototype.

Future work may also come from the expansion of the concept to cover peripheral areas. This expansion can take into account several suggestions received during the evaluation phase. In particular, the addition of functionality such as an advanced search by word or even image based search are valid possibilities. On the other hand, adapting Vibe to support information related to social networks or reference collections, as well as integrating blog aggregation capability and user statistics are also possible paths for future work.
Bibliography


[Lee 09] Yun-Jung Lee, Min-Jung Bae, Gyun Woo, and Hwan-Gue Cho. A personalized visualizing and filtering system for a large set of responding


Appendix A

Evaluation Session Form

Sessão de Avaliação de Usabilidade

Obrigado por ter aceite participar nesta experiência. O nosso principal objectivo com esta experiência é a avaliação comparativa de algumas interfaces para leitura de blogues individuais, com base na eficácia dos seus conceitos subjacentes.

Esta sessão tem o tempo estimado de 30 minutos e terá início com um período de experimentação livre de cerca de 5 minutos. A experiência continuará com a realização de uma série de tarefas usando uma interface de leitura de blogues. Finalmente, solicitaremos a resposta a um pequeno questionário sobre si e sobre a interface utilizada. Durante esta fase poderá utilizar a interface livremente sobre o mesmo blogue.

Por favor, tenha sempre presente que o objectivo desta experiência é avaliar a utilidade e eficácia da interface e não o seu desempenho enquanto utilizador da mesma. Por esse motivo, por favor, sinta-se à vontade para ”pensar alto” durante a realização das tarefas.

Gostaríamos também de, desde já, agradecer o seu tempo e esforço.

Formulário de consentimento

As tarefas realizadas durante esta experiência serão gravadas utilizando um programa de captura de ecrã. Por favor, tenha em conta que toda a informação obtida durante esta experiência permanecerá anónima e será utilizada no âmbito de uma tese de mestrado sobre o assunto aqui tratado.

Assinatura:
Questionário

N.º de identificação:

Informação pessoal

1. Idade:

2. Sexo: Masculino □ Feminino □

3. Por favor, marque as respostas que se aplicam:
   □ Eu leio blogues.
   □ Eu deixo comentários em blogues.
   □ Eu participo num blogue como autor.

4. Caso leia blogues, por favor marque as respostas que se aplicam:
   □ Eu leio blogues através das interfaces originais.
   □ Eu leio blogues através de um agregador/síndicação de conteúdos (RSS).

5. Quantas horas por dia gasta nestas actividades (em média)?
   Menos de 1 hora □ 1-3 horas □ 4-8 horas □ Mais de 8 horas □

6. Qual a característica/funcionalidade de que mais gostou no leitor de blogues?

7. Qual a característica/funcionalidade de que menos gostou no leitor de blogues?
Tarefas

Por favor, responda às seguintes questões, pela ordem dada e uma de cada vez, mediante consulta do blogue selecionado. Cada tarefa tem a duração limite de 3 minutos.

• Qual a data e hora em que o autor publicou o seu primeiro post?

• Qual o mês (e ano) em que o autor publicou mais posts?

• Qual o dia da semana mais utilizado pelo autor, durante o ano passado, para publicar posts?

Perceber como os posts foram publicados ao longo do tempo foi...
Muito difícil □ Difícil □ Normal □ Fácil □ Muito Fácil □

• Qual a tag mais utilizada no blogue?

• Quais as duas tags mais utilizadas durante o ano passado?

Perceber qual o tópico principal do blogue foi...
Muito difícil □ Difícil □ Normal □ Fácil □ Muito Fácil □

• Qual o comentador que comentou mais posts no blogue?

• Qual o comentador que comentou mais posts publicados durante este ano?

Identificar comentadores habituais foi...
Muito difícil □ Difícil □ Normal □ Fácil □ Muito Fácil □

• Qual o post mais popular no blogue (post com mais comentários)?

• Qual o post mais popular publicado no ano passado?

Identificar posts populares no blogue foi...
Muito difícil □ Difícil □ Normal □ Fácil □ Muito Fácil □
• Aceda/abra o post mais recente com a tag “dinheiro”.

• Aceda/abra o post sobre “caldeirada de peixe”.

• Aceda/abra o post mais antigo publicado em Setembro de 2010.

• Aceda/abra o post com a tag “beleza”, publicado em Agosto de 2010, numa quarta-feira.

Encontrar posts utilizando a informação associada foi...
Muito difícil □ Difícil □ Normal □ Fácil □ Muito Fácil □

• Aceda/abra o post encontrado na tarefa anterior.

Escolha as opções que se aplicam. Encontrei o post previamente acedido através...
□ da aplicação do mesmo método utilizado na tarefa anterior.
□ do reconhecimento de uma imagem no post.
□ do reconhecimento de um vídeo no post.
□ do reconhecimento do título do post.
□ do reconhecimento da data do post.
□ da aplicação de outro método: ________________________________

• Aceda/abra o post mais recente que já tenha sido lido.

• Aceda/abra o post mais recente ainda por ler.

Identificar posts já lidos foi...
Muito difícil □ Difícil □ Normal □ Fácil □ Muito Fácil □
Leitor de blogues

Por favor, responda a algumas questões sobre o leitor de blogues:

<table>
<thead>
<tr>
<th>Discordo bastante</th>
<th>Concordo bastante</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Imagens e vídeos ajudam-me a encontrar posts que já li no passado.</td>
<td>□ 1 2 3 4 5</td>
</tr>
<tr>
<td>2. Saber o comprimento do conteúdo de um post antes de o ler é útil.</td>
<td>□ 1 2 3 4 5</td>
</tr>
<tr>
<td>3. O mecanismo de filtragem é útil para a exploração do blogue.</td>
<td>□ 1 2 3 4 5</td>
</tr>
<tr>
<td>4. A expansão/colação de posts e comentários é útil para navegar no blogue.</td>
<td>□ 1 2 3 4 5</td>
</tr>
<tr>
<td>5. Ter informação sobre os comentários que cada post recebeu é útil.</td>
<td>□ 1 2 3 4 5</td>
</tr>
<tr>
<td>6. Ter acesso directo a cada comentário é útil.</td>
<td>□ 1 2 3 4 5</td>
</tr>
<tr>
<td>7. A diferença entre posts e comentários lidos e não lidos é perceptível.</td>
<td>□ 1 2 3 4 5</td>
</tr>
<tr>
<td>8. As funcionalidades na barra lateral são úteis para obter uma visão global do blogue.</td>
<td>□ 1 2 3 4 5</td>
</tr>
<tr>
<td>9. O calendário permite perceber a dinâmica do blogue ao longo do tempo.</td>
<td>□ 1 2 3 4 5</td>
</tr>
<tr>
<td>NÚMERO</td>
<td>DESCRIÇÃO</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>1.</td>
<td>Penso que gostaria de usar este sistema frequentemente.</td>
</tr>
<tr>
<td>2.</td>
<td>Achei o sistema desnecessariamente complexo.</td>
</tr>
<tr>
<td>3.</td>
<td>Achei que o sistema foi fácil de utilizar.</td>
</tr>
<tr>
<td>4.</td>
<td>Penso que iria precisar do suporte de alguém especializado para poder usar este sistema.</td>
</tr>
<tr>
<td>5.</td>
<td>Achei que as várias funcionalidades do sistema estavam bem integradas.</td>
</tr>
<tr>
<td>6.</td>
<td>Achei que havia demasiada inconsistência neste sistema.</td>
</tr>
<tr>
<td>7.</td>
<td>Imagino que a maioria das pessoas iria aprender a usar este sistema muito rapidamente.</td>
</tr>
<tr>
<td>8.</td>
<td>Achei o sistema muito incômodo de utilizar.</td>
</tr>
<tr>
<td>9.</td>
<td>Sentí-me muito confiante ao utilizar o sistema.</td>
</tr>
<tr>
<td>10.</td>
<td>Precisaria de aprender muitas coisas antes de me poder habituar a este sistema.</td>
</tr>
</tbody>
</table>