VIBE: Blog Visualization and Exploration

Nelson Marques
Department of Computer Science and Engineering
IST/UTL
nprm@ist.utl.pt

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 presents higher usability scores than 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.

INTRODUCTION
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.

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. The use of images should promote recognition rather than recall when searching for entries, which should also support read wear visible indications. Furthermore, the user should be allowed to control what information should be presented at a given time, for which we intend to provide an interface that allows the interactive filtering of posts by using associated information.

In this document we describe a solution to the previously identified problem, focused on flexible navigation, and presentation of enhanced blog entries that promote content recognition. It includes a filtering mechanism combined with overview functionality and the presentation of posts in collapsed form that include image and video thumbnails to promote recognition rather than recall. Entries are also equipped with read wear cues.

The next sections of this document are organized as follows. We first introduce the interface design of our solution, along with the corresponding decisions, followed by the interaction design, which corresponds to the description of the behavior of the interface. Then, we continue with user evaluation and present the obtained results and findings, which are subject to discussion in the following section. To finalize, we compare our solution with similar solutions in the related work and provide our conclusions and future work.

INTERFACE DESIGN
Our foremost concern with the initial low-fidelity prototype of Vibe 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 time.

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. Each collapsed entry in Vibe can be expanded to display its full content. 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. This 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.

The sidebar of the interface, on the right of Figure 1, provides complementary functionality, such as sorting and filtering techniques that affect the posts presented on the navigation window.

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 concerns 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. As we can see in Figure 2, on the top, comments are 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 2, bottom), 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.

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 criteria 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 for 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, an option to turn off all filters was positioned in the sidebar.

In order to adjust our solution to real content and to handle the relevant information using suitable techniques, we conducted a study using a sample set of 127 blogs extracted from one of the most used platforms, Blogger, and performed a statistical analysis. The results are summarized in Tables 1 and 2.

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

Table 1. The values related to the 127 blogs analyzed in this study, sorted by relevance.

The results obtained from the blog analysis led to changes to the collapsed post, the first of which has to do with the representation of the length of each entry, which was adjusted to fit 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 2 (top). This option is available only if more than one image is present within the content of
Table 2. Statistical results related to the data obtained from 127 blogs used in this study, sorted by relevance.

<table>
<thead>
<tr>
<th></th>
<th>Average</th>
<th>S.D.</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>6/6</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>

Figure 3. The interface on the final prototype of Vibe, with the calendar selected on the sidebar.

Figure 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.

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. 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, which allows to sort by time, post length, popularity, number of images and number of videos.

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 for 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 (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.

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 allow for its collapse, to improve efficiency and prevent users from having to navigate to the top of the post once more to return to its collapsed form.

**INTERACTION DESIGN**

Vibe follows a Web-based client-server architecture that includes a database for storing the necessary information. As depicted in Figure 5, the server is responsible for the extraction of the information from blogs and managing content, while the client collects user input and presents content accordingly.

The communication between the client and the server is made through a set of web services, which allow for determining if a given blog is valid and 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 the previous confirmation that the blog is valid and it is already stored in the database. If that is not the case and the blog cannot be extracted or is not valid, the server will inform accordingly. Otherwise, it will proceed 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, will update the database with that information.

In order to extract blog data, we use the API of Blogger to retrieve the list of most recent posts of a blog, given a valid feed URL. After extraction, the collected information is processed to provide more flexibility and better comply with the mechanisms used by the solution. The resulting data is stored in an SQL database using a custom made object-relational mapping mechanism that translates the objects processed into SQL information. When requested by the client, the obtained results are then converted back to objects.

The user interaction with the system is made through a Web-based client interface supported by a Web browser. This component, along with the associated functionality and communication with the server, was implemented using the Google Web Toolkit1. In Vibe, the interaction spans two main elements on the interface, in particular entries, which include posts and comments, and the sidebar.

In Vibe, each post (see Figure 6) is firstly presented in collapsed form and can be clicked on to give access to the content.Collapsed posts also allow users to sequentially browse through the thumbnails of the contained images and videos. Clicking on a thumbnail, on the other hand, allows them to access videos and see images in higher resolution, in a new layer that resembles a media gallery.

![Image](https://via.placeholder.com/150)

**Figure 6. Detailed view of a post in Vibe.**

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, 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 7. 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.

Since reading an entry in Vibe implies its expansion, this is used to activate the visual read wear information. In the case of comments, this information is also applied to the squares.

1http://code.google.com/webtoolkit/
representing them on the comment bar, through which read comments are represented as darker squares, as opposed to unread comments, as we can see in Figure 8.

The sidebar on the interface incorporates both complementary functionality and an overview mechanism that also adds to the exploration of the blog. 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 (see Figure 9).

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. The next tab includes the tag list, which 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.

The same concept is applied to the commenters in the following tab. The overview of the blog is completed with 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 to users, as calendars are concerned.

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 10, top image). 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 10, second image). 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 10, third image). 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 10, bottom image). Depending on what aspect the user wants to explore next, he is able to combine several filters until the desired results are presented.

The majority of the filters found in posts are replicated in the sidebar and integrated with the overview mechanism. 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.

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. In order to navigate through the entries on each page, users can scroll through the available information on the navigation window.
Figure 10. Here we present a use case scenario involving sequential filtering, using Vibe. After reading a post about ponies (top image), the user selects the corresponding tag, affecting the visible posts (second image). Then, after reading some posts and comments, he decides to filter the posts by commenter (third image). The results include only the posts with the tag "ponies" and commented by the selected commenter (bottom image).

USER EVALUATION
To measure both the performance of our prototype in the visualization and exploration of individual blogs and the satisfaction of the users while using it, 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.

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.

Figure 11. The selected blog, as seen on the original interface, with the blog archive and a portion of the tag list visible on the right.

During the selection of the blog, we always kept in mind that we were choosing one of the test interfaces simultaneously. Therefore, in addition to the previously enumerated characteristics, we chose a common blog\(^2\) that includes some structural navigation mechanisms, 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 11.

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, to avoid user errors.

The Vibe prototype used in these tests corresponds to the final prototype described earlier. It includes all the functionality that composes our solution.

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.

The prerequisite to participate in the final usability sessions was 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

\(^2\) http://tostoescadesca.blogspot.com/
was 33, 32.2 and 29.3 years old, respectively. 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.

Each test was performed on a familiar place to the user, using the same computer. 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. 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, each of which was recorded separately, using screen capturing software, and the time to complete each task was recorded. 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.

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.

These sixteen tasks were 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).

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[1] (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. The users performing the tasks on our prototype were also asked to answer a few questions about it.

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.

These sixteen tasks were 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).

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[1] (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.

RESULTS AND FINDINGS

The most important results collected during the evaluation phase 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. 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 3 summarizes the time spent by 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.

On the first set of tasks, our prototype had the lowest execution times, with the exception of the second task, where the usual interface took the lead. This confirms the usefulness and flexibility of our solution on providing both an appropriate time sorting method, and suitable filters integrated with the overview mechanism. The usual blog interface, on the other hand, suffered from the need to navigate across several pages of the blog, but benefited from some information present on the calendar. The main reason for the results obtained by Google Reader was the lack of a proper overview.

The following sets of tasks, related to tags, commenters and popularity, showed once more the benefits of combining sorting techniques with a suitable filtering mechanism, as applied to Vibe. While the absence of tags and comments on the Reader explains the very limited results, the usual interface suffered again from its cumbersome navigation, also accusing the need for additional overview mechanisms with advanced filtering methods.

On the tasks to find posts based on the contained information, the three interfaces had similar results, except for two cases. In particular, the lack of a search by word mechanism made it difficult to complete one of those tasks in our solution. On the other hand, the usefulness of the filtering mechanism was confirmed when considering more than one piece of information to find a post.

As for the revisitation tasks, none of the interfaces really stood out, since they obtained approximately similar results.

Considering all 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.

In terms of difficulty classification, most of the users testing our prototype reported having faced less difficulty in performing the given tasks, compared with the other two solutions, as we can see in Figure 12.

As rated by the users on a scale from 1 (not easy at all) to 5 (very easy).

At the end of each test session, the user was asked to evaluate the interface tested, for which we used the System Usabil-
### Discussion

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 somewhat 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 and the characteristics that drew the attention of the users. Characteristics such as the accessibility and simplicity of the interface, along with the presentation of the first words on each post, were 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 was definitely the filtering mechanism, with the calendar appearing as the second most liked feature. Some users also considered 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.

### Table 3. Average time of execution for each task, on each interface, and the corresponding standard deviation, in seconds.

<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>Avg. 115, S.D. 29</td>
<td>Avg. 89, S.D. 49</td>
<td>Avg. 36, S.D. 27</td>
</tr>
<tr>
<td>In what month and year did the author publish more posts?</td>
<td>Avg. 51, S.D. 26</td>
<td>Avg. 125, S.D. 46</td>
<td>Avg. 77, S.D. 40</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>Avg. -134, S.D. -34</td>
<td>Avg. -34, S.D. -19</td>
<td></td>
</tr>
<tr>
<td>What is the most used tag in the blog?</td>
<td>Avg. 29, S.D. 22</td>
<td>Avg. -24, S.D. 20</td>
<td>Avg. 17, S.D. 11</td>
</tr>
<tr>
<td>What were the two most used tags during the last year?</td>
<td>Avg. 68, S.D. 40</td>
<td>Avg. -38, S.D. -34</td>
<td>Avg. -41, S.D. -37</td>
</tr>
<tr>
<td>Who is the commenter that commented more posts in the blog?</td>
<td>Avg. -17, S.D. 19</td>
<td>Avg. -17, S.D. 11</td>
<td>Avg. -17, S.D. 11</td>
</tr>
<tr>
<td>Who is the commenter that commented more posts during this year?</td>
<td>Avg. -17, S.D. 19</td>
<td>Avg. -17, S.D. 11</td>
<td>Avg. -17, S.D. 11</td>
</tr>
<tr>
<td>What is the most popular post in the blog with most comments?</td>
<td>Avg. 136, S.D. 30</td>
<td>Avg. 25, S.D. 20</td>
<td>Avg. 17, S.D. 11</td>
</tr>
<tr>
<td>What is the most popular post published in the last year?</td>
<td>Avg. 73, S.D. 14</td>
<td>Avg. 7, S.D. 4</td>
<td>Avg. 7, S.D. 4</td>
</tr>
<tr>
<td>Access/open the most recent post tagged with ‘dinheiro’.</td>
<td>Avg. 30, S.D. 39</td>
<td>Avg. 17, S.D. 10</td>
<td>Avg. 17, S.D. 10</td>
</tr>
<tr>
<td>Access/open the post about ‘caldeirada de perce’</td>
<td>Avg. 60, S.D. 110</td>
<td>Avg. 40, S.D. 40</td>
<td>Avg. 40, S.D. 40</td>
</tr>
<tr>
<td>Access/open the post tagged with “beleza” and published in August 2010, on a Wednesday.</td>
<td>Avg. 58, S.D. 37</td>
<td>Avg. 10, S.D. 10</td>
<td>Avg. 10, S.D. 10</td>
</tr>
<tr>
<td>Access/open the same post found in the last task.</td>
<td>Avg. 28, S.D. 26</td>
<td>Avg. 9, S.D. 4</td>
<td>Avg. 9, S.D. 4</td>
</tr>
<tr>
<td>Access/open the most recent read post in the blog.</td>
<td>Avg. 41, S.D. 73</td>
<td>Avg. 49, S.D. 49</td>
<td>Avg. 49, S.D. 49</td>
</tr>
<tr>
<td>Access/open the most recent unread post in the blog.</td>
<td>Avg. 23, S.D. 10</td>
<td>Avg. 10, S.D. 10</td>
<td>Avg. 10, S.D. 10</td>
</tr>
</tbody>
</table>

### Table 3. Average usability scores for each of the tested interfaces, and corresponding standard deviation, as evaluated by the users, using the System Usability Scale.

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 most used functionality and also lower usability scores given to the usual interface are attained. As opposed, the somewhat contradicting low difficulty and also lower usability scores given to the usual interface during task completion and the high usability scores attained.

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, were 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 was definitely the filtering mechanism, with the calendar appearing as the second most liked feature. Some users also considered 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.
terface 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. 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.

**RELATED WORK**

The current format of blogs is mostly imposed by blog creation software, as referred by Herring et al [3] in their research. A blog that follows the usual format is normally composed of a column containing a limited sequence of posts and one or two sidebars, limited by a header and a footer. While the footer is commonly left empty, most of the time the header includes oversized and distracting title text or graphics.

Each post comprises a header and a footer on its own, around a body containing the main matter. Post header and footer are generally used for stating the title of the post and common associated metadata. The sidebars usually contain information about the author, links that support blog navigation, links to other sites and sometimes custom plugins and advertising.

The usual blog interfaces require users to scroll down through an arbitrary number of posts before giving access to navigation links to previous posts, as opposed to Vibe, where navigation links are always visible. It is also very difficult to identify if a post has or has not been read in the past, since read wear cues are usually applied to HTML links and have little visibility or are inaccurate.

Basic search is limited to some blogs and most of them 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. The calendar used in Vibe, on the other hand, provides a much compact and flexible navigation that can be used in conjunction with other filters.

In an effort to modernize current day blogs with the latest in web technology, Blogger recently announced five new dynamic blog templates. 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.

Although there are some concepts present 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, Blogger Views lacks a proper overview and a flexible navigation mechanism in order to provide an updated blog exploration experience.

Indratmo, Vassileva and Gutwin [4] developed a tool that represents visually blog entries and related comments along a chronologically ordered timeline. Each entry encodes the length of the text contained and indicates read wear status, with the same applying to comments. Users can zoom over the timeline and filter by tags and commenters, using a query interface that reflects the results in real time.

This tool represents a remarkable advancement in blog visualization, but it drastically breaks with the usual blog presentation, also failing to provide proper integration with the content itself, as opposed to Vibe. iBlogVis also does not take advantage of graphic elements placed on entries, such as pictures, to ease the identification of potentially interesting entries.

Bross and colleagues present PostConnect [2], an interactive visualization tool that uses a circular interface to display all posts of a single blog. Upon user interaction, a set of visual relationships is generated between the selected post and the remaining posts, based on the associated tags. Tags are represented in the center of the interface and reflect the number of posts they are associated with. The circular component is complemented by a listing with all the posts and tags related to the currently activated post, and a filtering mechanism that allows for both selecting and limiting the data available for exploration and analysis.

The major contribution of PostConnect results from the combination of the circular interface and the filtering mechanism as a whole. However, it does not support the representation of a large amount of entries, due to visibility and usability issues, requiring users to zoom over particular regions to further explore the content. The connections between tags and related posts may also lead to visual confusion if there are many tags used at the same time. Like iBlogVis, PostConnect also lacks integration with the actual content of the blog and the complete absence of reference to comments makes of it a blog overview mechanism rather than a fully capable blog reader like Vibe.

Lee and colleagues propose TRIB (Telescope for Responding comments for Internet Blogs)[5], a tool capable of showing clusters of blog comments within the same screen, using user-defined filter dictionaries. In TRIB, each article represents a sun and the associated set of keywords represent the plan-
ets around it. Comments reflect their length and are placed around keywords as satellites. The proximity between elements is determined by the frequency of the used keywords.

TRIB solves the problem of representing a high amount of blog comments on the same screen with hierarchical visualization. However, beyond the need to define a dictionary, the interaction presents additional difficulty for traversing comments to follow a discussion thread. The absence of functionality to view the remaining content also clearly limits the ability of TRIB to explore blogs.

Aggregators or feed readers follow a different approach to information visualization, collecting entries of selected websites and organizing them into a sort of personal newspaper. Google Reader\(^4\), for example, presents the user with a condensed view on the latest unread headlines of the subscribed feeds. Each feed appears as a list of items in the main column of the reader, in reverse chronological order, and can be presented in condensed or expanded format. While in the condensed format each post occupies one line that includes the title, a few words from the text contained and date, the expanded view is similar to the original post. The Reader has limited support for read wear and site statistics, covering only the last 30 days and 10 000 posts.

Although it has integrated word search, Google Reader does not include comments nor tags, which reveal to be very useful when properly integrated, such as in Vibe, which uses them to bridge contextual navigation with an overview of the blog. As opposed to our solution, it also does not take advantage of images to promote the recognition of entries.

CONCLUSIONS AND FUTURE WORK

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.

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, especially 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 users on this feature does not justify the space it occupies on collapsed posts.

Future work may consist in the expansion of the concept to cover related peripheral areas. In particular, adapting Vibe to support information related to social networks or reference collections, as well as integrating blog aggregation capability are valid and possible paths for future work.

REFERENCES