Treemap-Based Website Navigation For Non-Hierarchical, Interlinked Sites: The Trackback Map

Richard Atterer
University of Munich, Media Informatics Group
Amalienstr. 17, 80333 Munich, Germany
richard.atterer@ifi.lmu.de

Max Tafelmayer
University of Munich, Media Informatics Group
Amalienstr. 17, 80333 Munich, Germany
max@tafelmayer.de

ABSTRACT

Typically, the navigation area of a website is organized as a hierarchical menu of pages and subpages. For some types of websites, such as blogs, this is not a suitable choice: The importance of blog articles changes dynamically, e.g. depending on their age or the amount of public interest they generate. Navigation to other blogs via links to related articles (so-called "trackbacks") plays an important role, both to find related content and to estimate the relevance of an unknown blog based on the reputation of the blog that links to it. In this paper, we propose a new, interactive type of navigation area which addresses the special needs of websites with a flat hierarchy that link to related sites. The Trackback Map relies on a treemap to visualize the relative importance of individual articles on a blog at a single glance. By zooming into the map, the user can reach articles on other blogs that link to the current blog's article, or (to any depth) articles that link to those articles. A prototype of the concept has been implemented as a WordPress plugin. In a user study, it is compared to established navigation concepts, e.g. a tag cloud.

Keywords

WWW, navigation, treemap, Ajax, weblog, blog, trackback

Categories and Subject Descriptors

H.5.2 [**Information Systems**]: Information Interfaces and Presentation—*User Interfaces*; H.5.4 [**Information Systems**]: Information Interfaces and Presentation—*Hypertext/Hypermedia*

1. INTRODUCTION

While creating a website, the designer usually pays attention to organizing the content pages in a hierarchy. For many recent sites that can be roughly categorized as "Web 2.0 sites", this is no longer the case: They are large collections of media items (e.g. text articles in a Wiki or blog) which are connected to each other only by direct links to individual pages, or indirectly via user-defined tags, i.e. keywords. In either case, no strict hierarchy of content exists for the entire site. At most, an artificial (and often not useful) hierarchy is created – for example, blog articles are organized according to their date of publication. Consequently, these sites typically do not feature navigation menus. To find content, users either follow links in the main content area, or they use the site's search functionality.

While our ideas could be adapted to different types of sites, we concentrate on blogs in the remainder of this text. With regard to navigation, blogs are interesting because links to related articles on other sites play an important role. Users are actively encouraged by blog authors to follow these links. Mechanisms (trackbacks and pingbacks) have even been developed to allow an article page to link back to other articles that reference it. As a result, users who browse through blogs often face the situation that they can follow links to many related articles which discuss the current article. However, it is hard to pick the most interesting link, since important clues about the quality of the article behind the link ("how many trackbacks has it received?" – "is it referenced by a well-known blog?") cannot be found out easily. Furthermore, once a link to an unknown blog has been followed, it is hard to find the most important and popular articles on the new site.

This paper presents a new type of web page navigation area which provides a treemap visualization to help in the situations mentioned above. Initially, the Trackback Map shows an overview of the current blog. By zooming into an article with the scroll wheel, the user can display information on articles that reference the article, or (recursively) articles that reference that article. Similar to the ranking algorithm of a search engine, the article importance (i.e. its size in the treemap) is determined by how recent it is and how many comments/trackbacks it has received. At any time, a click on an article causes the browser to load the respective page.

This paper is organized as follows: After a look at related work in section 2, a detailed description of the design and features of the Trackback Map follows in section 3. Its Ajax-based implementation is described in section 4. The evaluation in section 5 shows that our treemap-based solution performs well compared to established navigation areas (free text search, tag cloud). Finally, section 6 presents concluding remarks and discusses areas of future work.

2. RELATED WORK

A number of related tools aim at visualizing website content and making navigation decisions easier. In [5], the hierarchically organized site is displayed as a treemap covering the entire browser window. Zoom operations on categories and pages in the content hierarchy replace menu-based navigation. This concept abandons established web UI concepts, which makes it more ambitious than our approach of only providing a treemap in the navigation area.

NordiCHI 2008: Using Bridges, 18-22 October, Lund, Sweden

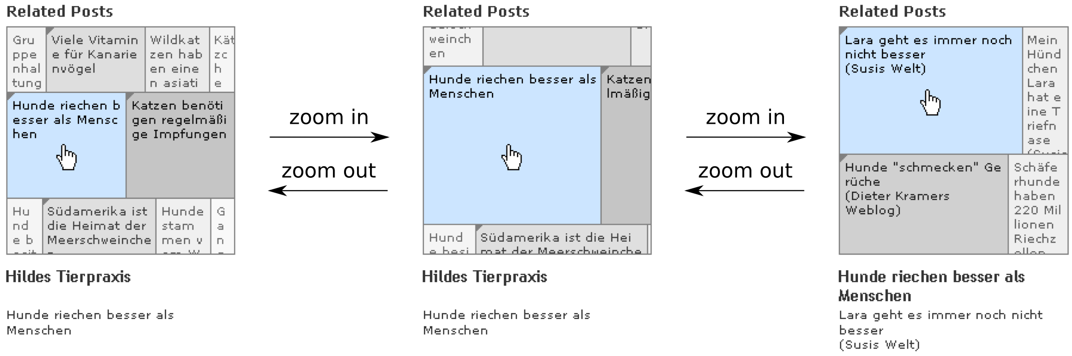

Figure 1: The Trackback Map is a treemap-based navigation area for blogs which visualizes the importance of their articles (left). By zooming into an article with the scroll wheel, users can explore entries on other blogs that link to the article (right).

Other efforts also use treemaps to represent the content hierarchy: The Navigational View Builder [6] concentrates on information extraction from existing sites. The tool is not integrated with a browser or the site navigation. The authors of [8] visualize an online forum and user activity in its different discussion threads on the small display of a mobile device.

Further research projects use treemap representations to visualize other types of navigation-related information. In the case of [3], the personal bookmarks of a user are automatically clustered hierarchically, the size of rectangles in the treemap corresponds to the number of bookmarks per cluster. For navigation using free text search, [7] presents a visualization of search results for large documents that are organized in chapters, sections etc. The generated treemap displays the document hierarchy and uses colours to represent the number of occurrences of the search terms.

Our approach distinguishes itself from these efforts by providing an interactive treemap-based navigation area on an ordinary website, and by visualizing information (importance/relevance score, links from other sites) that takes into account the "collective opinion" of the readers of a site.

3. TREEMAP VISUALIZATION AND INTERACTION CONCEPT

The Trackback Map is designed to be displayed in a vertical navigation area of the web page, i.e. to the left or right of the main content. As shown in figure 1, a heading "Related Posts" is followed by the main square treemap (190×190 pixels in the prototype) and a few lines of text which display information about the part of the map the mouse pointer is currently located over.

3.1 Trackback Map Visualization

After a blog page has loaded, the Trackback Map on it initially shows an overview of the blog (leftmost part of figure 1). Each rectangle corresponds to one article on the blog. If the current page is an article page (rather than the main overview page of the blog), the respective rectangle is highlighted. The prototype employs a strip treemap [1] to visualize the importance of articles. However, the algorithm for generating the treemap has been slightly altered to place the biggest rectangle (most important article) horizontally and vertically approximately in the middle of the map for a more balanced look.

The importance i_a of an article a is calculated as the weighted sum of several factors as follows:

$$i_a = 3 \cdot (\text{trackbacks}_a + \text{pingbacks}_a) + 2 \cdot \text{comments}_a + \text{recent}_a$$

Thus, the number of links from other blogs (in the form of trackbacks or pingbacks) is most important, followed by the number of comments left by users on the article page. The value recent_a depends on the publishing timestamp of the article. It ranges from 0 (oldest article) to 1 (newest article on the blog). Apart from the size of rectangles, their colour also depends on i_a – the background colour of more important articles becomes darker.

3.2 Interaction With the Trackback Map

The most obvious type of interaction with the treemap is clicking on rectangles, which causes the browser to load the article that was clicked on. As with normal text links, it is possible to use the middle mouse button or the browser's context menu to open articles in new browser tabs.

When the pointer is placed over a rectangle, the mouse scroll wheel can be used for zoom operations. As shown in the middle and right part of figure 1, a smooth animation is used to first enlarge the article under the mouse pointer, and then cause it to fill the entire navigation area. Once this has happened, the article information fades out and is replaced with information on blog entries that link to the article. This way, users can quickly explore blog articles that link to the article the pointer is over. Zooming is only possible over articles which have at least one link from another blog. A small mark in the upper left corner of the article's treemap rectangle appears if this is the case. The zoom behaviour is similar to the one described in [2], e.g. with changes to the aspect ratio of a rectangle during the zoom animation.

Further zooming is possible for the articles that are displayed now. Using the scroll wheel, users can zoom in until no more links by other blogs exist, or they can zoom out until the Trackback Map reaches its initial state again. Theoretically, it is possible for cycles to be present in the graph of links. The prototype does not prevent infinite zooming from taking place. However, in practice such loops are unlikely to occur. Zooming in on an article is only possible if the respective blog has installed the Trackback Map.

It should be noted that the initially displayed treemap only shows articles from a single blog (the current website), whereas the treemap that is displayed after one or more zoom operations consists of articles from different blogs. However, in both cases the calculation of article importance is based on the same principle. This mixture of internal and external links in the navigation reflects the popularity of citing related articles on blogs. As outlined in section 1, it supports the users in the decision as to which of the external links to follow.

Strictly speaking, the Trackback Map is a series of individual treemaps linked via zoom operations, rather than one large treemap:

The importance calculation only takes into account comments and links for the current article, it disregards links to articles that link to the current article. Primarily, this is due to constraints of the implementation – fetching information about trackbacks recursively from many external sites would cause performance problems. In our experience, using only local information for the calculation results in a good enough approximation for the treemap to be useful.

4. IMPLEMENTATION

The Trackback Map is implemented as a WordPress 2.6 plugin. WordPress is a popular, open source and web based blogging software written in PHP 4. Its automatic support for trackbacks, the flexible plugin architecture and great usability made it the ideal choice for the implementation. The Trackback Map plugin consists of two major parts: the frontend and the backend.

The frontend is written in JavaScript and makes heavy use of jQuery 1.26, a JavaScript library that is integrated into WordPress. The initial view of the articles in the Trackback Map and all subsequent views are created using JavaScript. The rectangles and their contents that form the treemap are simple HTML elements that are styled and positioned with CSS. This allows the Trackback Map to be displayed in any modern web browser without the need for additional browser plugins. The jQuery Mouse Wheel plugin is used to zoom into and out of the treemap with the scroll wheel.

When a user zooms into the treemap to see the trackbacks of an article, an Ajax request is made to the backend of the plugin. The request contains the URL of the article as a parameter. Next, the backend forwards this request to the blog the article belongs to (which need not be the current blog) and retrieves the response. The importance of the trackbacks in the response is calculated by that blog according to its settings. Finally, the response is handed back to the frontend, parsed and the new view is generated. To speed up subsequent Ajax requests to the same URL, all responses are cached for five minutes.

When a user zooms out of the treemap, the data from the previous zoom level is retrieved from a history object that stores the data from all zoom levels.

The backend is written in PHP 4 and handles the WordPress integration, the importance calculation of articles and trackbacks, the Ajax requests and the communication with the other blogs. During plugin activation, a new table is created in the database to cache additional information about trackbacks from other blogs. The cached information consists of the number of comments, trackbacks and pingbacks a trackback has and is automatically updated every 12 hours. The exchange format for all communication is XML, which is sent via sockets over the HTTP protocol. The Trackback Map can only display trackbacks of an article if our plugin is installed on the blog the article belongs to.

The following settings of the plugin can be adjusted in the administration area of WordPress: The width and height of the treemap and the factors of the importance calculation.

5. EVALUATION

To evaluate the Trackback Map, a user study was conducted. Our hypothesis was that the participants would complete the tasks faster if the Trackback Map was used. Thus, the measured time acted as the dependent variable in the user study. The results of the questions were analysed with a two-tailed, paired t-test to check for statistical significance.

Each of the fifteen participants received a questionnaire that was divided into three parts: Personal data, five tasks, and final feedback.

Figure 2: During the user study, participants completed tasks by using either the Trackback Map or a search form together with a tag cloud (see `trackback-map.medien.ifi.lmu.de`).

Each of the five tasks consisted of two very similar sub-tasks, followed by questions. One of these sub-tasks had to be performed with the help of the Trackback Map (figure 2, left). For the other one, only a full text search and a tag cloud was available (figure 2, right). The tasks were based on three fictional blogs about pets. The main blog was the starting point of the tasks, and the other two blogs referenced articles of the main blog to generate the necessary trackbacks.

The participants were between 25 and 56 years old. Eight participants were male, seven female. Their education and profession varied, e.g. software developer, designer and electrician. Only some had a technical background. All participants use the Internet very often and are experienced consumers of digital media. However, only five use blogs every day, the rest only infrequently. Comments and especially trackbacks are hardly used by the participants to get further information about blog posts. Nine participants did not know trackbacks at all, and only five had heard of treemaps. To make sure every participant would be able to understand the concept of the Trackback Map, the following terms were explained prior to the tasks: Blog, comment, trackback, tag, tag cloud.

Before the tasks, the participants were encouraged to explore the Trackback Map on their own and find as many features as possible. This exploration phase revealed that some of the features of the Trackback Map are not as intuitive as expected. All fifteen participants used the Trackback Map to navigate through the blog by clicking on the rectangles, but only half seemed to notice the different sizes of the rectangles, recognized the small mark in the upper left corner and discovered the zoom function. Before proceeding to the following tasks, the features of the Trackback Map were explained to the participants.

The question of the first task was: “Which article of the blog about *[subject]* is the most important one?”, and the subject changed between the two questions. The measured task completion times indicate that the Trackback Map performs better. The difference is statistically significant ($p < 0.05$). Afterwards, the participants were asked why they had chosen the articles. After using the Trackback Map, almost all named the size of the respective rectangle as the determining factor. Without the Trackback Map, half based their decision on the subject, the rest mentioned the number of comments or the actuality.

The question of the second task was: “Which article out of two is more important?” The two articles to be compared were different for the two sub-tasks. Again, users performed significantly better while using the Trackback Map ($p < 0.05$). When asked why they had chosen one article over the other, two thirds of the Trackback Map users said they observed the size of the respective rectangle, one third mentioned the subject. Without the Trackback Map, almost all looked at the subject, the rest mentioned the number of comments or the actuality.

The question of the third task was: "Which trackback of a given article is the most important one?" The article changed between the two sub-tasks. As before, users were faster with the Trackback Map (again, $p < 0.05$). Half of the Trackback Map users based their decision on the rectangle size, one third mentioned the subject and the rest had other reasons. Without the Trackback Map half said they looked at the subject, one third mentioned the actuality and one fifth mentioned the author.

For the fourth task, the test users were asked to find a piece of information on another blog through a trackback of a given article. While users of the Trackback Map performed better, the difference was not statistically significant ($p = 0.15$).

The instructions for the fifth task were: "Count all trackbacks and trackbacks of trackbacks of a given article". The article changed between the two sub-tasks. This task was fairly easy with the Trackback Map because only a few zoom operations were required, but involved many clicks if the Trackback Map was not available. Accordingly, the Trackback Map users performed much better, the difference is highly significant ($p < 0.003$).

The final feedback provided by the participants showed wide acceptance of the Trackback Map. Twelve participants found that the use of the Trackback Map gave them advantages finishing the tasks. All participants stated that they had a very good understanding of the Trackback Map after the tasks, and nearly all participants found it easy to learn and had fun using it. Some complained about the graphical presentation and the interaction. Nearly all participants wanted to use the Trackback Map in the future – more as a permanent navigation than an optional navigation.

There was also some negative feedback: Almost all participants said that the readability was not good due to the arbitrary line breaks. About one third of the participants mentioned that the function of the small mark in the upper left corner is not clear and intuitive, that the mark is not visible enough, that the importance calculation is good but needs a different presentation, that an indicator of the current zoom level is required, or that the display of the full text of an article or trackback must be placed directly beside the rectangle.

To summarize, the results of the user study are very positive: The Trackback Map performed significantly better in four out of the five tasks, the result of the fifth task is even highly significant. Together with the mostly positive feedback from the test users, it seems evident that our prototype is a valuable navigation method for the tasks it was tested with.

6. CONCLUSION

In this paper, we have introduced a new navigation concept for non-hierarchical, interlinked websites. We concentrated on blogs with their lack of a menu-based navigation and emphasis on linking to related content on other blogs. Our interaction concept makes several contributions: First, it uses techniques from search engine ranking, assigning importance scores to individual articles on a blog by looking at links from external sites. Second, the data is visualized in a treemap that appears on the web page as a navigation area, allowing users to obtain a quick content overview as well as navigating easily to important articles on the blog. Finally, the zoom concept is unique in the way it visualizes trackbacks (incoming links) from other sites. Without its presence, it would be hard to extract the respective information from blog pages even though that information is relevant for navigation decisions.

The advantages of the concept also became obvious during the evaluation of our open-source prototype: When using the Trackback Map, users were much faster when navigating through blog articles. Likewise, direct feedback by the test participants in a questionnaire was positive.

The current version of the prototype has some minor problems. Some aspects of the user interface (such as the zoom indicator or the readability of labels) should be improved. Furthermore, as for any content that is under the control of a website, it is possible for blogs to cheat by using a different importance calculation than the default one. Finally, the treemap visualization currently only takes into account direct links to the current blog. For better accuracy, direct and indirect links could be taken into account, similar to a search engine's PageRank algorithm.

7. REFERENCES

[1] B. B. Bederson, B. Shneiderman: Ordered and quantum treemaps: Making effective use of 2D space to display hierarchies. In *ACM Transactions on Graphics*, vol. 21, no. 4, pages 833–854, 2002

[2] R. Blanch, É. Lecolinet: Browsing Zoomable Treemaps: Structure-Aware Multi-Scale Navigation Techniques. In *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pages 1248–1253, 2007

[3] Y. Feng, K. Börner: Using Semantic Treemaps to categorize and visualize bookmark files. In *Proceedings of SPIE – Visualization and Data Analysis*, vol. 4665, San Jose, CA, USA, pages 218–227, January 2002

[4] R. Gandhi, G. Kumar, B. Bederson, B. Shneiderman: Domain Name Based Visualization of Web Histories in a Zoomable User Interface. In *Proceedings of the 11th International Workshop on Database and Expert Systems Applications*, IEEE Computer Society, Los Alamitos, CA, USA, pages 591–598, 2000

[5] T. Di Mascio, I. Salvatore, L. Tarantino: Navigation by zooming in boxes: preliminary evaluation. In *Proceedings of the working conference on Advanced visual interfaces (AVI)*, Venezia, Italy, pages 262–265, 2006

[6] S. Mukherjea, J. D. Foley: Visualizing the World-Wide Web with the Navigational View Builder. In *Proceedings of the Third International World-Wide Web Conference*, Darmstadt, Germany, pages 1075–1087, April 1995, Elsevier

[7] S. Schlechtweg, P. Schulze-Wollgast, H. Schumann: Interactive Treemaps With Detail on Demand to Support Information Search in Documents. In *Proceedings of the Eurographics/IEEE TCVG Visualization Symposium on Data Visualization 2004*, pages 121–128, Eurographics Association, Aire-la-Ville, 2004

[8] Yi Wang, Li-Zhu Zhou, Jian-Hua Feng, Lei Xie, Chun Yuan: 2D/3D Web Visualization on Mobile Devices. In *Proceedings of the 7th International Conference on Web Information Systems Engineering (WISE) 2006*, Wuhan, China, pages 536–547, October 2006, Springer LNCS 4255