Interactive Visualization for Opportunistic Exploration of Large Document Collections

Simon Lehmann*, Ulrich Schwanecke, Ralf Dörner

Department of Design, Computer Science and Media, RheinMain University of Applied Sciences, Kurt-Schumacher-Ring 18, 65197 Wiesbaden, Germany

Abstract

Finding relevant information in a large and comprehensive collection of cross-referenced documents like Wikipedia usually requires a quite accurate idea where to look for the pieces of data being sought. A user might not yet have enough domain-specific knowledge to form a precise search query to get the desired result on the first try. Another problem arises from the usually highly cross-referenced structure of such document collections. When researching a subject, users usually follow some references to get additional information not covered by a single document. With each document, more opportunities to navigate are added and the structure and relations of the visited documents gets harder to understand.

This paper describes the interactive visualization *Wivi* which enables users to intuitively navigate Wikipedia by visualizing the structure of visited articles and emphasizing relevant other topics. Combining this visualization with a view of the current article results in a custom browser specially adapted for exploring large information networks. By visualizing the potential paths that could be taken, users are invited to read up on subjects relevant to the current point of focus and thus opportunistically finding relevant information. Results from a user study indicate that this visual navigation can be easily used and understood. A majority of the participants of the study stated that this method of exploration supports them finding information in Wikipedia.

Key words: information visualization, opportunistic exploration, browsing, searching, wikis

1. Introduction

A common approach to collect information about a specific subject in large, cross-referenced document collections like the Wikipedia is to start with an already known term and open the associated article. Within the article, links to other terms probably relevant to the research are found. If more than one article has to be read to get the desired information, a user has to follow some of the links to other articles. They usually have to be read one at a time and again contain links to further articles probably relevant to the subject. While navigating between several articles, a user has to keep track of what they already have read and how the piece

of information they are currently reading relates to everything they already have read. Additionally, a user might want to know what other links they encountered in all the previously read articles and which therefore are probably worth following, especially if many of them lead to a single article.

The problem arises from the complex structure of highly cross-referenced articles. They form a directed graph, which consists of hundreds of thousands of articles and usually significantly more links. The English language version of Wikipedia currently comprises 3 million articles and over 70 million links between them [28].

Common web-browsers used for exploring web-based information resources are not providing any means to help users with the specific tasks presented when researching a subject in such a large network of articles. They only provide a history of visited pages which can be navigated forwards

*Corresponding author. Tel.: +49 611 9495 1294; fax: +49 611 9495 1210

Email address: simon.lehmann@hs-rm.de (Simon Lehmann)

Preprint submitted to Elsevier

September 23, 2009

and backwards. Besides this simple linear history of pages, they do not tell the users what other links they might consider following and how two different pages are related to each other.

In this paper, we present a navigation concept for the exploration of such large information resources, which visualizes the structure of articles a user has already seen and where a user might find further information related to the already researched subject. While there have been similar approaches to on-line, interactive visualizations of hypertext document structures in general, our work focuses on the additional visualization of all related links a user might want to follow, and how to help with choosing potentially interesting articles. By applying a weighting function to the yet unread articles, we can highlight more important articles and help a user to opportunistically explore the vast amount of information.

This paper is organized as follows. In section 2 we will review other approaches taken to interactively visualize large document structures. Section 3 will briefly explain the concept of opportunistic exploration and how we are applying it to the task of searching for information. The visualization and interaction concept we developed is then described in section 4. Section 5 is outlining the implementation of our concept in the web-based application *Wivi*¹. In section 6, the set-up and results of the user study we conducted are described and analyzed respectively. In section 7, we finally give our conclusion and present potential future work.

2. Related Work

There have been several approaches to interactively visualize large document collections in order to make exploring and finding relevant information easier.

The InfoSky system developed by Andrews et al [3] is a visual explorer for news articles which contain no cross-references themselves, but are classified into a hierarchy many levels deep. It visualizes all articles at once in a *galaxy of stars*, where each article is visualized as a star. These stars are clustered by the hierarchical structure of the articles they represent. Like earlier work in this field – such as the landscape metaphor ([7, 29]) or the hyperbolic browser ([20]) – this approach visualizes

all documents at once and makes the documents explorable by using a semantic zoom technique which reveals individual documents. This works well with a fully known set of documents which can be hierarchically grouped (either manually or automatically). Even though visualizing large numbers of documents in the range of ten to hundred thousands is possible, visualizing millions of documents at once in an interactive environment is still a problem [14].

Very large document collections like the Wikipedia are too large to be visualized interactively at once and in the case of the world wide web, the exact amount of documents and their structure is unknown. One approach to deal with such large and only partially known document collections is to exclusively visualize the immediate surroundings of the document space already explored by a user. The NESTOR tool implemented by Eklund et al [11] simply visualizes the history of a user's browsing of the world wide web. The WebOFDAV system implemented by Huang et al. [17] (and similar, more recent systems [18, 19]) provide the immediate neighbors of visited pages as navigational elements to find new pages to visit. Most of those systems use various force-directed layout algorithms for displaying the graph of pages and some also use clustering methods to improve the generated layouts. Those systems simply display what a user has seen and what can be immediately reached from there, but provide no means to help the user deciding which pages to visit next.

Besides the interactive browsers for general hypertext documents there have been attempts to make more specialized visual browsers for knowledge spaces like the Wikipedia. Hirsch et al. [16] developed two interactive visualizations of Freebase (a "Semantic Wiki") and Wikipedia named Thinkbase and Thinkpedia respectively. Their approach uses similar techniques for visualization as those used for general hypertext documents, but utilizes semantic web data for nodes and links. This leads to a different visualization concept, because some nodes represent actual articles while other nodes represent semantic concepts belonging to the articles. The Thinkpedia visualization also introduces weighting of the semantic concepts based on the relevance value computed by the semantic web service used for retrieving all concepts relevant to an article. This helps a user in finding more relevant data, but since navigating to another article results

¹<http://wivi.slashslash.de/>

in a complete regeneration of the graph which does not contain the previously visited articles anymore, it is only of limited use for extended exploration of the Wikipedia.

Another accentuation technique for expeditiously finding relevant terms in text documents is the textarc visualization [22] which displays each line of a single text on a large circle with a very tiny font size. Inside the circle, the words of the text are displayed with different sizes and positions according to their frequency and distribution in the text. Words of higher frequency are made larger, and thus more visible. The position of a word is determined by the centroid of the points where it occurs in the text on the circle. This allows to visually spot words which are most important, and where the words are used predominantly in the text. Words which are used throughout the text are positioned near the center of the circle, while those that appear only in a certain section are positioned close to that section on the circle. This visualization provides a good overview of the important terms of a longer text like a book, and makes it easy to quickly navigate to parts of the text which deal with a certain term. However, it is only applicable to a single document which also needs to be preprocessed and thus fully known. Like other approaches which focus on visualizing a single or only a few previously selected text documents (e.g. DocuBurst [8], FeatureLens [10]), this is of limited use when dealing with a dynamically growing set of multiple articles of an encyclopedia which are each relatively short compared to a book.

The usage of graph visualizations is not the only way to approach the problem of exploring document collections. The MedioVis system developed by Heilig et al. [15] provides multiple visualizations (coordinated views) like tables, scatter plots or node-link diagrams to search in digital libraries. It follows a top-down approach where a user is able to look at the entire data space at once (by different selectable visualizations) and then drill down to the required information by applying different filters. This is similar to the galaxy of stars used by the InfoSky system, but instead of a single visualization based on a fixed hierarchical structure, MedioVis allows for a more dynamic construction of data sets by its users.

The different systems and approaches to navigating large document collections usually provide some way to navigate to new documents a user has not yet read. While some systems leave the selection of

potentially relevant material to their users, others like e.g. Thinkpedia provide a visual weighting of edges to indicate how relevant two nodes are. By weighting and highlighting new documents, users can intuitively navigate to these documents and opportunistically find information they are interested in, but would not have found otherwise. This technique of 'berrypicking' [4] in online search interfaces can be supported by visualizations in different ways, as the work of [16], [23] or [6] has shown, and thus should also be applied to exploration of large document collections.

3. Opportunistic Exploration

When researching a subject, it might happen that users do not have a precise understanding of what they are searching for. They do not know what is of central importance or what exactly belongs to the subject in question. But usually they will recognize a term they actually wanted to find or which might be more appropriate for the subject in question.

In order to get started, users have to pick a term they already know. While reading the corresponding article, users gain more knowledge about the subject and encounter additional cross-references to other terms potentially relevant to their research. These additional cross-references pose new navigational opportunities. When a term or subject among these is more interesting to the users than the current article, they will likely follow this path. This is a very natural way of searching for information [4], but unfortunately it is commonly not very well supported by the well established information resources available today.

To help exploring the articles of the Wikipedia, our visualization brings potentially interesting links to new articles to the user's attention. To determine which articles are presented to a user and how interesting articles are distinguished from other articles, we need some criteria we base our decisions on.

As said earlier, the articles and the links between them form a directed graph $G = (V, E)$, where each article is represented as a vertex $v \in V$ and each link is represented as an edge $e \in E$ going from the article vertex where it is found to the article vertex it points to. In *Wivi*, this article graph contains everything the user has read and everything the user could have reached from the introductory sections of the articles. Initially, the graph contains just the first article the user has started with and the articles linked from the first section of this article.

Figure 1: A partial graph of a large document collection, containing all visited (filled circles) and referenced, unvisited (empty circles) articles.

Further articles and edges are added to the graph with every new article the user visits (an example is seen in figure 1). Because the amount of links found in a single article can easily get very large, only links found in the first section of the article are used to add new navigational opportunities to the graph. This decision is based on the assumption, that the most important cross-references are found in the introductory section, where a rough overview of the article is given.

Among the articles not yet read are some of higher importance than others. While it is impossible to know what a user actually is searching for, we can make some assumptions about their potential interests based on the history of articles they read. We define a relative *degree of interest* (*DOI*) a user has in an article based on the structure and history of the article graph. Every article in the graph has some importance to a user which is independent of which article they are currently reading. This is the *a-priori-importance* (*API*) of an article. Additionally, articles gain or loose importance depending on the current focus of interest of the user. How much an article gains or loses is defined by the *distance* (*D*) between this article and the current focus of the user. These measures can be combined into a function yielding the current DOI of an article v :

$$DOI(v) = API(v) - D(v)$$

This function is essentially the same as the DOI function defined by Furnas [13], except we define D not as a distance between two points.

By assuming that users are interested in all articles they read and that the links the authors of the articles have placed are sensible, we inferred that an unvisited article with more inbound links from already visited articles can be seen as more important to a user. This is used as *API* of an unvisited article in the article graph. With $d_G(v)$ being the inbound degree of an article v and $\Delta(G)$ the largest degree over all vertices, *API* of the unvisited articles can be formally defined as:

$$API(v) = \frac{d_G(v)}{\Delta(G)} \quad (1)$$

Because users are potentially more interested in articles they recently read, and are less interested in articles they visited in the beginning, we use the age of the visited articles to weight their outbound links. The age of a visited article is determined by the number of articles a user has visited since the last visit to that article. In other words, the age of each visited article increases by one with each new article a user visits. This can be seen as the temporal distance between the current focus of interest and the focus at the time the user was reading a previous article. The temporal distance D of an unvisited article v can then be defined as:

$$D(v) = \frac{1}{d_G(v)A(G)} \sum_{v_i \in N_G(v)} a(v_i) \quad (2)$$

$A(G)$ is the highest age of all visited vertices, $a(x)$ the age of a single vertex and $N_G(x)$ the neighborhood of a vertex.

Those two functions define how the DOI of each unvisited article is determined. For every unvisited vertex v of the graph G , the *DOI* function assigns a degree of interest to that vertex depending on the already visited vertices. It yields a value in the interval $[-1, 1]$, where -1 represents the lowest and 1 represents the highest degree of interest.

In essence, this function provides a guess on what a user might read next based on the history of visited articles, weighted by the order they were visited. Based on this DOI function, our visualization is able to put emphasis on potentially interesting articles a user might be looking for. When presenting the user these potential next articles, they are

enabled to opportunistically choose the article they were actually looking for.

4. Interactive Visualization

The goal of *Wivi* is on the one hand to prevent the user getting lost in the vast amount of articles to explore and on the other hand to highlight opportunities for their next navigation step. While this might be done solely in the article text itself, there are several reasons why a separate graphical representation is more suitable for this purpose.

The amount of text present in most articles usually is large enough to span multiple screen pages even on large displays. A highlighting technique applied to links within a text can only bring attention to links currently visible within the current window. While the problem of highlighting potentially interesting links in the current text might be overcome in some way, it is considerable more difficult to maintain a representation of the previously visited articles and their connections to the current article just within the text itself, because not every article previously visited is necessarily present as a hyperlink in the text.

In order to present this information to the user, a separate representation of the articles needs to be provided. By choosing a graphical representation over a simple textual listing of the browsing history, the connections between the history of visited articles and potentially interesting, yet unvisited articles can be visualized.

Our visualization is based on the article graph $G = (V, E)$, as defined earlier. Every node of this graph is visualized as a text label on a grey background shape. The visualization strictly distinguishes between visited and unvisited articles, both in shape and layout of the nodes, as they fulfill two different functions – visualizing the past and the possible future. A schematic view of the general layout can be seen in figure 2.

The visited articles represented by circular shapes and are laid out by using the radial-tree algorithm [9]. In order to use this algorithm, a tree has to be generated from the general article graph. This is done by using a breadth-first traversal starting from the first visited article on the graph, obeying the direction of edges and stopping at unvisited articles. The layout algorithm then starts at the root of the tree (the first visited article) and lays out each node by using preorder traversal. The first article is put into the center of the viewport

Figure 2: Placement of unvisited articles on concentric circles based on their DOI-value around the already visited articles which are laid out using a radial tree algorithm.

and each level of depth of the tree is displayed using a circular layout where the radius increases with each level. Both the edges present in the generated tree as well as those present in the article graph are drawn as straight, gray lines between the nodes. The tree edges are drawn slightly wider and darker than the other edges to make the hierarchy of the navigation history stand out.

The still unvisited articles are represented by rectangular shapes, and are laid out on three rings around the visited articles. Which ring an article is assigned to is determined by the DOI function as defined earlier. Each ring represents a third of the the functions' range $([-1; 1])$. An article found to be more relevant is placed on a ring closer to the center articles. This proximity to the already visited articles indicates to the user that this article is considered as potentially important to what they are searching for and might be worth reading. Most edges connected to unvisited articles are hidden to reduce the visual clutter they would produce otherwise. Only the edges connected to the article currently read by the user are shown.

Due to the space consuming nature of article texts and the visualization of structural information, they have to be separated from each other. However, the user should be able to keep the connection between the structure and the currently

Figure 3: The user interface of *Wivi*. On the left, the current article text is displayed, on the right the visualization of the article graph.

viewed article. As the work of Lai et al. [19] indicates, users do not want to use graphical representations alone for navigation. Thus, *Wivi* provides the user with an integrated browser, which contains both the visualization and the display of the current article itself.

Because the article text and the visualized article graph are displayed together, both can be used for navigation. The user can read through the article and click on any link to another article, just like in a standard web-browser. Alternatively, the user can also click on any article node in the visualization. In either way, the application then loads the new article and inserts it and any newly found referenced articles into the article graph. To make the connection between the article text and the visualization more apparent, the corresponding article node of the currently loaded article and its connected edges are highlighted in green. Also, when the user hovers over a link in the article text, which has a corresponding node in the article graph, this node and its edges are temporarily highlighted in blue (see figure 3). This highlighting is also done, when the user hovers over a node in the article graph itself. In this case, all hidden edges are made visible during highlighting, which allows the user to find out which unvisited articles are referenced by a certain visited article or which visited articles reference a certain unvisited article (see figure 4).

When changes of the graph or layout of the visualization are made, the user must be able to understand how it has changed. Not only does this

Figure 4: Highlighting of visited (left) and unvisited (right) article nodes when hovered over with the mouse. All out- and inbound edges are made visible.

help maintaining the mental map, but it is also important if the user wants to go back to earlier visited articles and thus has to locate them. Transitions between two states of the graph can be animated, which allows the user to efficiently perceive the changes and maintain their mental map [21, 12]. The animation of position changes, which happens when the DOI of one or more unvisited articles changes or new articles are inserted, are done by interpolating between polar coordinates. Unlike an interpolation between Cartesian coordinates, this creates circular movements and results in reduced crossing of animation paths, which helps the user to follow the transition between two states [30]. Figure 5 shows how the visualization changes when new articles are visited.

Besides the interaction necessary to navigate between articles, the application also provides a zoom lens in the visualization which follows the mouse pointer. It enlarges nodes close to the mouse pointer and makes them fan out, to make reading and selecting overlapping nodes easier. The distortion of the node positions is also based on polar coordinates which integrates well with the other animations.

5. Implementation

We realized *Wivi* as a web-based application. While our approach to navigate Wikipedia and similar document structures could have been implemented in a native desktop application, we decided against it. Today, web-based applications can be run in a common web-browser present on virtually every desktop computer with internet connectivity. Wikipedia itself is a web-based application which makes its information easily and simply accessible by anyone with a web-browser. Because we want

Figure 5: An example of how *Wivi* visualizes the history and navigational opportunities over the course of time. The currently read article is marked green and all its out- and inbound edges are shown. The different computed DOI values, changing with every navigation step, can be clearly seen.

to improve the exploration of Wikipedia, our visualization has to be at least equally simple to access and use.

The visualization and user interface part of the application was developed with the Adobe Flex Framework [2] and the prefuse flare visualization library [27]. By using the Adobe Flash [1] platform, which is available on most web-browsers, we were able to provide an interactive visualization which can be used inside a normal web-browser and still use high quality drawing and animation techniques with reasonable performance. This also had positive effects on the complexity of the implementation, because we did not have to deal with the stateless nature of HTTP.

Because of the security restrictions of today's web-browsers and especially the Adobe Flash Player in which the client part is run, we could not directly access the data of the Wikipedia from within the client application itself. Instead we needed to implement a server-side proxy, which provides the necessary data needed for the visualization – mainly article texts and links to other articles. The server-part was implemented in Java as a Webservice. It was deployed on an Apache Tomcat [25] installation with an Apache Axis2 [24] web-service engine.

To retrieve the article texts and the links to other articles found within one article, we use the HTTP-based API provided by the MediaWiki software on which the Wikipedia runs. This API allows accessing most of the information the wiki contains. To use the API from our webservice, we developed a client library in Java, which hides the details of sending and retrieving the raw data over the HTTP protocol from the application.

The separation into a client part, which is responsible for the graphical user interface and the state of the application, and a server part, which implements the operations to retrieve article texts and links to other articles, makes it possible to change the way how the visualized data is retrieved. Especially, it allows to change how links to other articles are retrieved and thus makes implementing other methods for extracting links from articles simple. As mentioned earlier, we currently use the links from the content of the introductory section of the article.

Figure 6: The architecture of *Wivi*.

6. Evaluation

After implementing *Wivi* we conducted an anonymous remote usability test to gain data on how people benefit from our visualization when searching for information in the Wikipedia. We only asked the participants to perform one single task: To search for some subject they were interested at the time. We also offered a random selection of articles to choose from as a starting point, in case a participant could not think of a subject they were interested in. In order to allow the participants to familiarize with *Wivi* and the task, every participant was allowed to perform the task up to five times.

Additionally, all participants had to fill in a questionnaire, where they had to answer the following questions:

1. How often do you use the Wikipedia?
2. Do you use the search feature of the Wikipedia?
3. How many links do you follow on average, when doing research on a subject?
4. How do you rate your computer usage and skills?
5. How well did you know ...? (This question was asked for every subject a participant had searched for)
6. How much did the graphical representation of the visited articles help you with your search?

7. How much did the weighting of unvisited articles help you with your search?
8. How do you rate the usefulness of the graphical representation in general?
9. Would you use the application for future research in the Wikipedia?

6.1. Set-Up

Due to the web-based nature of *Wivi*, we were able to conduct a remote usability test instead of doing a test in a local usability lab. By choosing a remote usability test, we could reach significantly more participants. It also resembles more closely the typical scenario a user encounters when doing some research on a subject they are interested in [26].

As the test was to be conducted by the user themselves, without a moderator guiding them, we had to make the test completely self-explanatory and keep the tasks they should perform very simple and easy to understand. The test consisted of three parts, each separately presented on a simple web page. The first page, which was the entry point to the test, contained a short introduction to *Wivi*, the goals of the test and the following steps. When visitors agreed to participate in the test, they were taken to a separate page on which the task they should perform was explained. This page also contained a link to open the application in a new browser-window. When the users finished the task, they were presented the final questionnaire which they had to complete.

Each participant was assigned a random and unique session identifier at the beginning of the test, which was then used to associate the activity and questionnaire data to the participants. The activity of a participant within the application was recorded on the server via a separate web-service, which stored the type of activity together with the session identifier and the current timestamp. This data was used to reconstruct and understand what the participants were doing and how long they were using the application.

6.2. Results

During the period of 14 days in which the test was conducted, a total of 157 people were willing to participate in the test. Out of those, 72 participants (45.9%) finished the test by filling in the questionnaire. From the recorded activity of each participant, we computed the time a participant spent

using the application by computing the difference in time between the first and last action a user has performed. This duration was then used to filter out the data of those participants, who had spent less than 100 seconds using the application. This was done because we assume that at lower periods of use, a reasonable evaluation of the application is not possible. After this final filtering, the data of 56 participants remained and were used for our further statistical evaluation.

For statistical tests, we used the Mann-Whitney U Test for unpaired and the Wilcoxon signed rank test for paired values [5]. The significance threshold was chosen at $p = 0.05$ and p values lower than 0.003 are regarded as highly significant.

The majority of participants (69.6 percent) found the user interface of *Wivi* easy to understand and use. This indicates that further usage was not prevented by a cumbersome or unclear interface and thus the main part of the session duration was spent with actually using the application. It might also be inferred that the other ratings are based on the actual usefulness of the visualization.

As figure 7 shows, the usefulness of the visualization and the application was positively rated by the majority of participants. The question, if they would use *Wivi* for future research in the Wikipedia, was confirmed by 74.5 percent of all participants. This already indicates that our visualization presents a viable alternative for searching and browsing the Wikipedia and can be used without further instructions.

Comparing the subjective usefulness of the visualization of visited articles to the visualization of unvisited articles shows a highly significant ($p < 0.003$; signed rank test) difference. While the layout of the unvisited articles was seen as moderately useful, a majority of participants declared that the visualization of visited articles was useful for searching the Wikipedia (see figure 8). These different ratings are probably caused by the fact that the visualization of visited articles has a clear structure which is based on the navigational history and the links between the articles. This structure is easier to comprehend, because it is directly controlled by the user and is always made visible by the edges drawn between the nodes. On the other hand, the way how unvisited articles are placed on the outer rings might not be immediately understood, due to the invisibility of the edges leading to the nodes and the implied age of the visited articles. This was also explicitly mentioned by several participants in the

Figure 7: Rating of the visualization and *Wivi* in general

Figure 8: Rating of the visualization of the visited and the unvisited articles

comment field of the questionnaire.

Taking into account the previous knowledge the participants had for the articles they started with, a more differentiated result can be seen. The participants can be separated into two groups by the previous knowledge they initially had: a first group K_{low} , containing 21 participants who had no or little previous knowledge on average, and a second group K_{high} , containing 35 participants who had moderate to very good previous knowledge on average. The ratings of these groups regarding the visualization of unvisited articles show a visible difference, as it can be seen in figure 9. Comparing the ratings of the visualization of unvisited and visited articles from participants of group K_{low} shows that they rated them similarly (no significant difference was found). However, the other group K_{high} rated the visualization of visited articles significantly better ($p < 0.05$; signed rank test) than the visualization of unvisited articles. This reveals an interesting result: participants, who had no or only little previous knowledge, found the visualization of the visited and unvisited articles equally useful, possibly because they do not know what to specifically look for and thus benefit from the weighting of unvisited articles more than those participants who are already familiar with the subject.

Even though this might lead to the conclusion that our visualization is more suitable for researching a new subject, a further look at the general rating of the visualization shows that participants of group K_{high} rate it higher than those of group K_{low} (see figure 10). So it can be concluded that both groups benefit from our visualization.

7. Conclusion and Future Work

As previous research has shown, interactive visualization of large document collections can be used to improve navigation and finding relevant information. Our approach combines both a visualization of visited articles and articles that could be immediately reached from all visited articles. It also calculates a degree of interest of the unvisited articles based on the structure and history of the article graph. With *Wivi*² we created a browser for the Wikipedia or other wikis, which implements the visualization of already visited articles in a hierarchi-

²<http://wivi.slashslash.de/>; available in English and German

Figure 9: Rating of the visualization of the visited and unvisited articles, grouped by the average previous knowledge the participants had for the chosen subjects

Figure 10: Rating of the visualization in general, grouped by the average previous knowledge the participants had for the chosen subjects

cal tree layout and shows the related unvisited articles weighted by their degree of interest on circles around the visited articles. As the result of a user test shows, this approach is generally accepted and positively perceived as a viable interface to browse and search the Wikipedia. Especially the visualization of the visited part was well received, but also our concept of weighting and displaying the unvisited articles to enable opportunistic exploration appears to be promising.

As future work it would be interesting to explore other ways to visualize the unvisited articles and how the underlying weighting might be improved. One way to improve the weighting might be to take the categories of the articles into account, which provide some sort of clustering of articles. Also, it would be interesting to know how different approaches to extract the links between articles affect how well the concept of opportunistic exploration works. While *Wivi* uses a simple way to retrieve other articles, the implementation of more sophisticated methods could be easily integrated.

References

- [1] Adobe Systems Incorporated, 2009. Adobe Flash Player. <http://www.adobe.com/de/products/flashplayer/>.
- [2] Adobe Systems Incorporated, 2009. Adobe Flex 3. <http://www.adobe.com/de/products/flex/>.
- [3] Andrews, K., Kienreich, W., Sabol, V., Becker, J., Droschl, G., Kappe, F., Granitzer, M., Auer, P., Tochtermann, K., 2002. The infosky visual explorer: exploiting hierarchical structure and document similarities. *Information Visualization* 1 (3/4), 166–181.
- [4] Bates, M. J., 1989. The design of browsing and berrypicking techniques for the online search interface. *Online Review* 13 (5), 407–424.
- [5] Bradley, J. V., 1968. *Distribution Free Statistical Tests*. Prentice Hall, Englewood Cliffs, NJ, USA.
- [6] Bryan, D., Gershman, A., 1999. Opportunistic exploration of large consumer product spaces. In: *EC '99: Proceedings of the 1st ACM conference on Electronic commerce*. ACM, New York, NY, USA, pp. 41–47.
- [7] Chalmers, M., 1993. Using a landscape metaphor to represent a corpus of documents. In: *Proceedings of the European Conference COSIT '93*. Springer, Berlin/Heidelberg, Germany, pp. 377–390.
- [8] Collins, C., Carpendale, S., Penn, G., 2009. Docuburst: Visualizing document content using language structure. In: *Proceedings of Eurographics/IEEE-VGTC Symposium on Visualization (EuroVis '09)*. Eurographics Association, pp. 1039–1046.
- [9] Di Battista, G., Eades, P., Tamassia, R., Tollis, I. G., 1998. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, Upper Saddle River, NJ, USA.
- [10] Don, A., Zheleva, E., Gregory, M., Tarkan, S., Auvil, L., Clement, T., Shneiderman, B., Plaisant, C., 2007.

- Discovering interesting usage patterns in text collections: integrating text mining with visualization. In: CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management. ACM, New York, NY, USA, pp. 213–222.
- [11] Eklund, J., Sawers, J., Zeiliger, R., 1999. Nestor navigator: A tool for the collaborative construction of knowledge through constructive navigation. In: Proc. AusWeb '99: Fifth Australian World Wide Web Conference.
 - [12] Friedrich, C., Houle, M. E., 2002. Graph drawing in motion ii. In: GD '01: Revised Papers from the 9th International Symposium on Graph Drawing. Springer, London, UK, pp. 220–231.
 - [13] Furnas, G. W., 1986. Generalized fisheye views. SIGCHI Bull. 17 (4), 16–23.
 - [14] Harrison, C., 2006. Wikiviz: Visualizing wikipedia. <http://www.chrisharrison.net/projects/wikiviz/index.html>.
 - [15] Heilig, M., Demarmels, M., König, W. A., Gerken, J., Rexhausen, S., Jetter, H.-C., Reiterer, H., 2008. Medioviz: visual information seeking in digital libraries. In: AVI '08: Proceedings of the Working Conference on Advanced Visual Interfaces. ACM, New York, NY, USA, pp. 490–491.
 - [16] Hirsch, C., Hosking, J., Grundy, J., Feb. 2009. Interactive visualization tools for exploring the semantic graph of large knowledge spaces. In: Workshop on Visual Interfaces to the Social and the Semantic Web (VISSW2009).
 - [17] Huang, M. L., Eades, P., Cohen, R. F., 1998. Webof-dav — navigating and visualizing the web on-line with animated context swapping. In: WWW7: Proceedings of the seventh international conference on World Wide Web 7. Elsevier B. V., Amsterdam, The Netherlands, pp. 638–642.
 - [18] Huang, M. L., Eades, P., Lai, W., 2003. On-line visualization and navigation of the global web structure. International Journal of Software Engineering and Knowledge Engineering 13 (1), 27–52.
 - [19] Lai, W., Huang, X., Nguyen, Q. V., Huang, M. L., 2007. Applying graph layout techniques to web information visualization and navigation. In: CGIV '07: Proceedings of the Computer Graphics, Imaging and Visualisation. IEEE Computer Society, Washington, DC, USA, pp. 447–453.
 - [20] Lamping, J., Rao, R., 1994. Laying out and visualizing large trees using a hyperbolic space. In: UIST '94: Proceedings of the 7th annual ACM symposium on User interface software and technology. ACM, New York, NY, USA, pp. 13–14.
 - [21] Misue, K., Eades, P., Lai, W., Sugiyama, K., 1995. Layout adjustment and the mental map. Journal of Visual Languages & Computing 6 (2), 183–210.
 - [22] Paley, W. B., 2002. Textarc: Showing word frequency and distribution in text. Interactive Poster at the IEEE Symposium on Information Visualization 2002 (InfoVis 2002).
 - [23] Pu, P., Janeczek, P., Jun. 2003. Visual interfaces for opportunistic information seeking. In: Proceedings of the 10th International Conference on Human - Computer Interaction (HCI '03). Human Computer Interaction Group, Swiss Federal Institute of Technology, Lausanne (EPFL), Lawrence Erlbaum Associates, Philadelphia, PA, USA, pp. 1131–1135.
 - [24] The Apache Software Foundation, 2009. Apache Axis2. <http://ws.apache.org/axis2/>.
 - [25] The Apache Software Foundation, 2009. Apache Tomcat. <http://tomcat.apache.org/>.
 - [26] Tullis, T., Fleischman, S., McNulty, M., Cianchette, C., Bergel, M., Jul. 2002. An empirical comparison of lab and remote usability testing of web sites. In: Proceedings of the Usability Professionals Association Conference. UPA, Bloomingdale, IL, USA.
 - [27] UC Berkeley Visualization Lab, 2009. prefuse flare visualization toolkit. <http://flare.prefuse.org/>.
 - [28] Wikimedia Foundation, 2009. Data dump of the english language wikipedia. <http://download.wikimedia.org/enwiki/20090816/>.
 - [29] Wise, J. A., Thomas, J. J., Pennock, K., Lantrip, D., Pottier, M., Schur, A., Crow, V., 1995. Visualizing the non-visual: spatial analysis and interaction with information from text documents. In: INFOVIS '95: Proceedings of the 1995 IEEE Symposium on Information Visualization. IEEE Computer Society, Washington, DC, USA, pp. 51–58.
 - [30] Yee, K.-P., Fisher, D., Dhamija, R., Hearst, M., 2001. Animated exploration of dynamic graphs with radial layout. INFOVIS '01: Proceedings of the IEEE Symposium on Information Visualization 2001, 43–50.