Treasure Map: Search History for Young Users

Tatiana Gossen
Otto von Guericke University
Magdeburg, Germany
tatiana.gossen@ovgu.de

Sebastian Stober
Otto von Guericke University
Magdeburg, Germany
stober@ovgu.de

Andreas Nürnberger
Otto von Guericke University
Magdeburg, Germany
andreas.nuernberger@ovgu.de

ABSTRACT

In this paper we describe a novel concept of a search history visualization that is primarily designed for children. We propose to visualize the search history as a treasure map: The treasure map shows a landscape of islands. Each island represents the context of a user query. We visualize visited and unvisited relevant results and bookmarked documents for an issued query on an island. The treasure map offers several advantages over the existing history mechanisms such as context awareness, appropriate metaphor for children, looping visualization, smaller cognitive load and higher efficiency in refinding information. We discuss design decisions that are important to build such map and interact with it and present the first prototype of the map.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Design, Management

Keywords

Search History, Contextual Information, Children, Information Retrieval

1. INTRODUCTION

Nowadays, computer and Internet access is available in almost every household and the number of children who use modern technology and Web services from an early age grows every year [10]. Children, starting from primary school age, are using the Internet not only for fun but also to search for information usually related to their homework for school [6, 17]. In order to better support children at their search tasks, several web search engines with special support for children have been launched, e.g. *askkids.com* and *blindekuh.de*. Currently, their main purpose is to help children find child appropriate content on the Internet. However, the usability of those search engines is another, currently underemphasized, aspect. Search engines for children need to match children's particular skills to increase the usability [5].

There are several conceptual challenges that should be considered when designing appropriate information retrieval systems for children. Young users require emotional support, language support, memory and cognitive support, interaction support and support to judge document relevance more than adults [7]. In this paper we concentrate on memory support. According to information processing theory [13], the information processing of children differs from that of adults in terms of how they apply information and what memory limits they have, e.g. children can represent and process less information. This could explain children's "looping" behaviour during the information seeking process [1, 6]. "Looping" behaviour is characterized by frequent repeated queries, clicking and revisiting of result web pages during a search session.

Children also heavily and more frequent than adults exploit the back-button of the web browser [1]. The back-button is a very simple tool for web history navigation [14]. This is empirical evidence that children require search history and different history mechanisms incorporated in information retrieval systems for children are of importance. Besides the cognitive support, mechanisms of history tracking are important for children as part of an educational process. Children can learn to plan their searches and better understand the workings of a search system by revisioning their own search history. If the "looping" behaviour of children is intentional and part of a learning process, then a search history would also support children and make revisiting relevant information easier. Furthermore, in order to trigger users' memories about previous search actions, it is important to provide contextual information, e.g. to show the relations between the queries and documents.

2. RELATED WORK

The general idea of a search history is to record information-seeking steps of a user, i.e. his or her queries, search results and relevant contextual information [14]. This history provides the user with several benefits. Besides memory support, it also helps to refind, manage and use information, and it even gives the possibility to exchange information and the search process with other users in case of collaborative search [14]. The idea to support the users' search process by means of a search history is not new. In web search, the back/forward buttons, history lists, and bookmarks in the browser are usually used as history tools where the last two require more mental effort [14]. The back-button is easy to use and thus suits the abilities of children, but only allows to navigate back to a short list of previously visited pages. There is no overview of the visited pages.

A browser's built-in history provides a longer list of visited pages in a separate window that can be revisited, however it does not contain explicit information about the users' search queries and corresponding (visited) results. A long list of URLs without any contextual information is also hard to process, which makes it difficult to use for young users. Furthermore, a browser's built-in history without the time information has only limited use for tracking the time-line. The management of bookmarks is also problematic [14]. In order to use bookmarks the user has to annotate a web page as a favorite. Furthermore, he or she has to organize the link bookmarks into folders. Without a folder structure, using a larger number of bookmarks is difficult for users.

Besides the mentioned history tools of a web browser, there exist some more advanced history tools that are used in web-based search systems and on e-commerce sites. These are lists of most-recently issued queries, thumbnails as memory aids in browser and search history, and breadcrumbs, i.e. sequences of links that a user has clicked on since beginning a navigation session [11, Chapter 7]. These small helpers are useful to keep track of some information-seeking steps of a user, but they are scattered and users have to process this information to built an integrated history picture.

Several systems were proposed to support the search history in a more advanced way. Kaasten and Greenberg [12] developed a revisitation system that integrates the functions of the back-button, history and bookmarks of a web browser. The system provides a time-ordered history list of visited web pages together with small thumbnails for better recognition of web pages. A user can mark a visited page as a favorite. This page is then marked with a dog-ear in the thumbnail history to make it distinguishable from other pages. The system provides filtering for favorite pages, for pages from specified domains and for frequently visited pages. However, mainly due to the absence of an overview, this revisitation system is still a list of URLs with filtering options.

Another example is the *Ariadne* system [18] that was designed to explicitly support collaborative search through a graphical representation of the search process. It uses the metaphor of playing cards. *Ariadne* visualizes the search history as sequence of cards containing thumbnail outlines of the query and result screens. These cards are arranged in several horizontal rows. Each row contains a different search activity (querying, viewing result lists and saving individual documents), and time is indicated by position from left to right. The *Ariadne* interface allows a user to review and re-visit his sequence of actions. *Ariadne* shows very few items at once, requires scrolling to see a new portion. The interface is too abstract for children. This means that the *Ariadne* interface still requires much cognitive effort to process the search history and thus does not fit children's cognitive abilities.

The DLITE system [2] offers a more graphical interface to process the search history. It uses the metaphor of a workspace containing different components, i.e. queries, document components and result sets that are represented as graphical objects. Users of the system are able to issue queries and get document collections, manipulate and view the objects within those collections. Workspaces also contain a number of services. Users are involved in the retrieval processes and see the history information in the same window. After a query is created, the user can drag it to a query service to issue it and get the result set back.

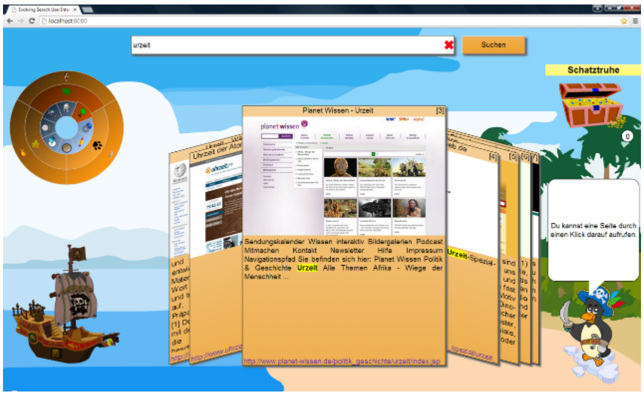

Figure 1: Screenshot of the Knowledge Journey user interface: a guidance figure and a treasure chest on the right hand side, query input elements on the top, a navigation menu on the left hand side and a coverflow with search results in the middle [8, 9].

A set of retrieval results is graphically represented as a circular pool, where the documents are represented as icons distributed along the edge of the pool. Users can interact with each document. If a document icon is activated, the document is shown in the web browser. After the result set is returned, a user can drag the set to a bibliography service to automatically generate a bibliography for all the documents dragged to the service. The DLITE system uses a workspace metaphor that is most likely not familiar to children. Being very different from traditional user interfaces, it requires much time to learn how to use the system even for grown ups.

To our best knowledge there exist no search history systems designed for children. In our work we use the existing ideas of search history support and design a novel interface for young users that supports search history. We introduce the search history, called *treasure map*, in the next section.

3. TREASURE MAP

The search history in form of a *treasure map* is an extension to a previously designed search interface for children [8, 9]. This interface provides search functionalities through both text input and navigating menu categories, and has a result storage, called treasure chest, to support cognitive recall. The search user interface uses the metaphor of a treasure hunt where a user takes a journey to gather relevant search results (see Fig. 1). In this setting the intention is two-fold: to increase the user satisfaction – What is a good journey without a treasure map?! – and to increase the efficiency of searching by providing intuitive access to past search results.

The treasure map contains a landscape of islands. We specify an island as a triple:

$$Island = \langle q, docs, tr \rangle, \quad (1)$$

where q is a query issued by an user or a selected menu category; *docs* is a set of relevant documents returned by the system and $tr \in docs$ are those documents that the user stored in the treasure chest. Each result set of a query is visualized as an individual island on the treasure map (see Fig. 2). On each island, each retrieved document is visually represented by a red X marking for a potential treasure, a white X for a page already visited, or a chest in case the respective result has been bookmarked, i.e. it is in the user's treasure chest. An example is shown in Fig. 2.

The position of the results on each individual island is determined using multidimensional scaling (MDS) [15]. MDS is a dimensionality reduction technique that computes an embedding in a low-dimensional euclidean space trying to preserve the distance relations as faithfully as possible. In this application, the results are arranged on the 2D map according to the distance derived from the cosine similarity of their corresponding high-dimensional TFiDF vector representations [16]. This produces neighborhoods of similar results.

Initially, only ten results are retrieved for each query. If more results are requested, these can be later added to the map. Their positions are determined based on the distances to the initial results, which serve as reference objects (called "landmarks"). This special variant of MDS called Landmark MDS has been proposed by de Silva [4, 3] as a computationally efficient approximation for large datasets. Here, it is applied with a different purpose – namely to consider result set extensions without having to recompute the arrangement. This way, a user can relate the new results with the previously known ones, which have not changed their positions.

Each island fits into a square bounding box of unit size but has a unique shape and landscape, which makes it easier to memorize and differentiate from others. A simple webservice generates the islands based on Perlin noise[1] subject to the following two constraints:

1. The border of each island map has to be water such that it fits into the global treasure map.
2. All landmarks, i.e. ten result documents retrieved initially, should be on land.

The first constraint can be ensured by adding a border region by simple coordinate transformation. The latter is satisfied by adjusting the sea level accordingly such that all landmarks are above. However, results added later may by chance still be under water. This is tolerated as it outweighs the possible confusion caused by changing the landscape of an island to accommodate new results. A possible alternative would be to move new results placed in the sea to the closest land position.

In the following we discuss our design decisions with regard to the map generation:

- *Island name:* An intuitive solution is to name an island after the user query. However, this information can be enriched by the representative words from the retrieved documents.

[1]The island generator is based on a python script by Christopher Breinholt, available at `http://breinygames.blogspot.com/`

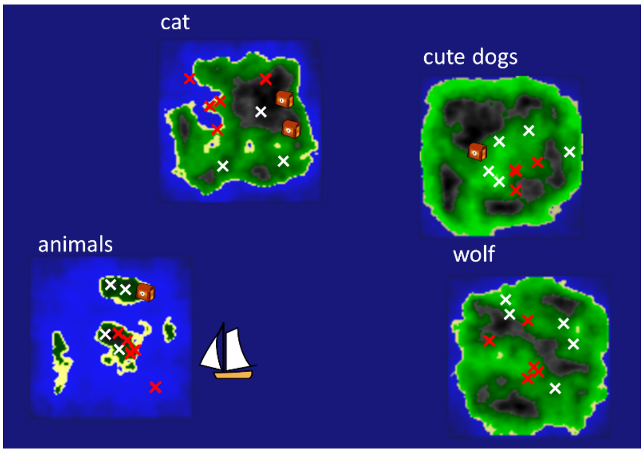

Figure 2: The treasure map contains a landscape of islands. Each island represents a search query issued by an user and is named after the query. On each island, ten retrieved results for the corresponding query are placed. The documents are visualized as X markings in white (already visited) or red (a potential treasure) and chests in case the respective results have been bookmarked. The boat is used in animation and visits the islands in the order the corresponding queries have been issued.

- *Island position:* The arrangement of the islands on the global treasure map is challenging. The arrangement has to be constructed incrementally adding one island at a time whereby the position of existing islands must not change when a new one is added in order to avoid confusion. The islands must not overlap. We propose to incrementally add islands to the map, whereby the position of a new island depends on the similarity to previous islands. The similarity is calculated based on the similarity of documents that belongs to the query. Islands representing similar search result sets are positioned close to each other.

- *Time line:* In order to visualize the order in which the queries were issued, we propose to use an animation where a ship following a dashed line on the water visits the islands in a sequential order. Another solution is to show a time stamp for each island or to numerate them, but in this way it is more difficult to comprehend the whole search process.

- *Visualization of the same documents on different islands in case the document state has meanwhile changed:* It it possible that islands share a subset of result documents. Furthermore, in some scenario a user can decide to visit a result document from this subset or to bookmark the document somewhere later in the search session, while the document is already an element of previous islands. In our opinion, the best solution here is to show a map where a document is visualized on each island based on its current state. In this way, it is easy for a user to comprehend which documents he or she has already visited, bookmarked and which documents have not been opened yet. In other words, it prevents user's looping behavior, i.e. a user won't accidentally open a document he or she has already visited.

- *Visualization of loops:* In case a user has issued the same query multiple times the corresponding islands are the same and have the same position. As an indicator for such a loop we can use the time line information. In addition, recurrent queries can be visualized by a loop-icon (e.g. flag) placed on the island.
- *Interaction:* In order to overcome scalability problems, we propose to extend the functionality of the treasure map by making each island zoomable using a magnifying glass. This will make the map exploration possible even if the number of user queries is large. Furthermore, a user can open each document placed on an island by clicking on it.

We implemented the first prototype of the treasure map. A hexagonal grid is used to position islands on the map. The position of a new island is calculated based on the similarity to previous islands and mapped to the grid field. In case the field is already occupied, the new island is placed in the alternate location from the neighborhood, i.e. the nearest grid where no island yet exists.

4. CONCLUSION

In this paper, we introduced a novel concept of a search history visualization in form of a treasure map for children as a primary user group. Our search history uses the metaphor of a treasure hunt that is familiar to young users. The treasure map contains a landscape of islands, where each island represents the context of one user query: Visited and unvisited relevant results and bookmarked documents for an issued query are shown on an island. Furthermore, the text documents are arranged on an island so that more similar documents are placed closer to each other. In this way users are provided with additional information on how diverse the results are. We discussed design decisions that are important to build the whole map and implemented a rudimentary prototype of the map.

We plan a user study with children to measure the usability of the map. We are going to test if, given a particular treasure map, children can construct a chronological order of search queries issued within a search session. Moreover, we plan to compare the treasure map with a standard browser history and measure the time for a user to find a particular document.

Acknowledgments

We are grateful to Severin Orth for his support in the implementation of the island generator and Steffen Metzner for his support in the implementation of the map visualization.

5. REFERENCES

[1] D. Bilal and J. Kirby. Differences and similarities in information seeking: children and adults as Web users. *Information Processing & Management*, 38(5):649–670, 2002.

[2] S. Cousins, A. Paepcke, T. Winograd, E. Bier, and K. Pier. The digital library integrated task environment (DLITE). In *Proceedings of the second ACM international conference on Digital libraries*, pages 142–151. ACM, 1997.

[3] V. de Silva and J. B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. pages 705–712, 2002.

[4] V. de Silva and J. B. Tenenbaum. Sparse multidimensional scaling using landmark points. Technical report, Stanford University, 2004.

[5] T. Gossen, J. Hempel, and A. Nürnberger. Find it if you can: usability case study of search engines for young users. *Personal and Ubiquitous Computing*, 17(8):1593–1603, 2013. DOI: 10.1007/s00779-012-0523-4.

[6] T. Gossen, T. Low, and A. Nürnberger. What are the real differences of children's and adults' web search. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information*, pages 1115–1116. ACM, 2011.

[7] T. Gossen, M. Nitsche, and A. Nürnberger. Knowledge journey: a web search interface for young users. In *Proceedings of the Symposium on Human-Computer Interaction and Information Retrieval*, HCIR '12, pages 1–10, New York, NY, USA, 2012. ACM.

[8] T. Gossen, M. Nitsche, and A. Nürnberger. Knowledge journey: A web search interface for young users. In *Proceedings of the Sixth Symposium on Human-Computer Interaction and Information Retrieval (HCIR 2012)*. ACM, 2012.

[9] T. Gossen, M. Nitsche, J. Vos, and A. Nürnberger. Adaptation of a search user interface towards user needs - a prototype study with children & adults. In *Symposium on Human-Computer Interaction and Information Retrieval (HCIR'13)*, 2013.

[10] A. Gutnick, M. Robb, L. Takeuchi, J. Kotler, L. Bernstein, and M. Levine. Always connected: The new digital media habits of young children. The Joan Ganz Cooney Center at Sesame Workshop, 2011.

[11] M. Hearst. *Search user interfaces*. Cambridge University Press, 2009.

[12] S. Kaasten and S. Greenberg. Integrating back, history and bookmarks in web browsers. In *CHI'01 extended abstracts on Human factors in computing systems*, pages 379–380. ACM, 2001.

[13] R. Kail. *Children and their development*. Prentice Hall Upper Saddle River, NJ, 2001.

[14] A. Komlodi, G. Marchionini, and D. Soergel. Search history support for finding and using information: User interface design recommendations from a user study. *Information Processing & Management*, 43(1):10 – 29, 2007.

[15] J. Kruskal and M. Wish. *Multidimensional Scaling*. Sage, 1986.

[16] C. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

[17] Medienpädagogischer Forschungsverbund Südwest. KIM-Studie 2010. Kinder+ Medien. *Computer+ Internet. Stuttgart*, 2011.

[18] M. Twidale and D. Nichols. Designing interfaces to support collaboration in information retrieval. *Interacting with computers*, 10(2):177–193, 1998.