Context Lenses – Document Visualization and Navigation Tools for Rapid Access to Detail

Andreas Dieberger & Daniel M. Russell

IBM Almaden Research Center
650 Harry Road,
San Jose, CA 95120, USA
{andreasd, daniel2}@us.ibm.com

Abstract: We introduce visualization and navigation tools -- Context Lenses (CL) -- which use a condensed representation of a document to provide a global overview of the distribution of key items as well as a focus view of their immediate context. The Context Lens is a focus+context dynamic information filtering tool allowing users to optimally choose how to access a particular piece of information. CLs support efficient information finding tasks by allowing rapid skimming of documents and document collections. Certain tasks can be accomplished using only the context information provided in the context lens, eliminating the need to access the actual document. Thus, use of a CL can eliminate the need to scroll within a document, or – in a distributed environment – to download the document. This hints at interesting applications for the display of search results and document collections on wireless PDAs. We present a user study about two different Context Lenses.

Keywords: Information Visualization, Navigation Tool, Information Lens, Focus and Context Techniques, Intermediary Based Computation, Information Scent.

1 Introduction

While the amount of information accessible by us grows every day, the amount of time we want to spend to look at this information gets smaller and smaller. Not too long ago, you could expect somebody to have a "look at your Web site," and generally browse around. Today, people want to see only the one snippet of information they really care about. Similarly, when searching for information on the Web it is essential to identify those snippets of information that are of interest, ideally pulling out the desired information without having to even look at the Web page the information came from.

We propose a tool, called Context Lens (CL), which allows users to quickly identify and access sections of interest within a document.

1.1 Information Scent

Russell, et al. describe sensemaking as activities in an information-rich environment, that require a certain amount of resources to accomplish an information analysis task (Russell, Stefik et al. 1993). Sensemaking involves optimizing the relative cost of activities in a given information environment. Examples of such activities are: following a link, scrolling a Web page, reading an entire paragraph of text and so forth. One of their key findings was that changing information representations enables users to perform certain tasks using different, cheaper activities. Therefore, a change in representation can reduce (or increase) the cost of sensemaking.

By definition, creating an abstract representation of an information environment eliminates a lot of information. Ideally, only information of immediate use remains in such a representation, along with enough context to perform the information processing task at hand. The key to a useful abstraction is, therefore, to show an adequate amount of residual context information. In Information Foraging theory, such residual information is defined as information scent (Pirolli and Card 1995).

2 The Context Lens

With the Context Lens we introduce a class of visualization and navigation tools, which allow users to keep an overview of a document or document collection, while being able to focus on details. Users maintain a sense of location within a document, grasp the distribution of potentially interesting information items and can study these items within their immediate context.

Navigation activities in an information space need to address the following fundamental questions of navigation (Dieberger 1994):

1. Where am I?
2. Where can I go?
3. Where is X?

Although the formulation of these questions hints at a spatial setting, they are valid for navigation in non-spatial domains, like Web pages. Unfortunately, many navigation tools, especially on the Web, ignore at least one of these questions.

Context Lenses address all three questions by providing a general overview showing all possible locations and the current focus within a document (collection). They also provide a representation of the document based on "what is of interest to the user", which helps answering the third question.

A CL permits users to scan through a document by employing an interaction technique called *brushing*. This technique shows another version of a data item while the pointing device (mouse) is in proximity of it (Card, Mackinlay et al. 1999).

Typically, brushing is realized through rollover events from a pointing device. Therefore, users do not need to click items to retrieve information about them, but simply touch them with the mouse cursor. This makes brushing is a more fluid and effortless interaction technique than pointing-and-clicking.

A CL allows brushing without having the entire document available. This is of special significance for very large amounts of data (for example for video data), as well as for devices with limited bandwidth connectivity (i.e. wireless PDAs).

2.1 What is of Interest to the User?

A key part of the CL is an abstract representation of the document, based on information about a user's interests. The representation highlights the position of interesting information items extracted from the document and provides rapid access to detailed information about these items. Assuming these items are indeed relevant, an appropriate representation can reduce the cost of information seeking activities, as explained in section 1.1.

User interest can be specified in a variety of ways. For example, users can specify keywords they are interested in. This approach is similar to electronic clipping services that scan news messages for mention of certain company names or keywords. Another possibility is to observe user behavior. An example is to monitor queries sent to search engines and to use the query terms as keywords of interest. The CL we describe in this paper uses a combination of these two techniques.

More elaborate techniques, like monitoring keywords in recently read email messages or Web pages, or to associate keywords with current projects could provide an even better handle on a user's current task context. A detailed description of these techniques is outside the scope of this paper.

2.2 Interacting with a Context Lens

A CL consists of two parts: an interactive abstract representation of the document, and the lens, which provides detailed information about items of interest. The abstract representation can be realized in a variety of ways, two of which we describe in more detail.

The *horizontal CL* was inspired by the Tilebar visualization (Hearst 1995). It consists of a regular grid where columns represents sections (for example paragraphs) of a document. Columns on the left correspond to sections at the top of the document.

Each row in the grid represents a *feature*. A feature is an "item of interest" and might indicate "keyword x found here", the presence of a phone number or of a foreign language term. The feature concept is quite flexible and can incorporate any type of information for which it is possible to build a detector and that we might find useful in the context of a certain document. Depending on the feature value, grid cells are drawn in different colors. This makes it easy to see the distribution of features within the document.

Note, that the term "section" is loosely defined. For the Web CL described in this paper we define sections based on the <P> (paragraph) tag in the page's HTML code. The optimal definition of sections depends on the document and task at hand. For certain tasks it is sufficient to represent only the top part of a longer document. In that case a section might be equivalent to a paragraph of text. Typically only 30-40 sections can be represented in a horizontal CL. If it is essential that a document be represented in its entirety, sections have to be defined as larger chunks of data to accommodate the entire document.

The main benefit of using a CL stems from dynamic interaction with it: when brushing over the CL, focus and context information about the corresponding section and feature is displayed as indicated in Figure 1a.

For a cell indicating presence of a feature (dark cells), brushing triggers display of a focus window, directly adjacent to the cell. The focus view contains detailed information about a feature in this particular section. In the case of the Web CL we show a keyword along with its immediate context.

If a section does not contain a feature (light cells) brushing does not display a focus view, but still shows certain context information under the CL. We use the start of the section for this purpose, as it often contains key information (see Figure 1a). Display of focus information needs to be instantaneous to make brushing a fluid, and dynamic process.

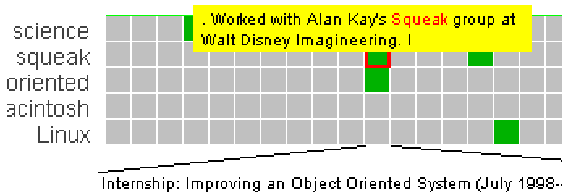

Figure 1a: Horizontal Context Lens with focus information for a keyword (squeak). The beginning of the section is shown below the CL.

Horizontal Cs are displayed at the top of the page. Users can determine from the CL whether they want to look at a section in more detail and scroll there or directly jump there by clicking a grid cell in the CL.

Our second implementation of the CL is very similar to the horizontal CL. The *vertical CL* uses a vertical grid with smaller grid cells. This design allows us to represent many more (100-150) sections in the page. This, however, makes single cells too small for accurate brushing. Therefore, the vertical CL uses brushing on entire sections (rows).

When brushing over sections, the vertical CL shows information for an entire section using a floating window on top of the Web page. This design can display focus information for several features at once and therefore is more economical in terms of screen real estate. Also the vertical CL shows the beginning of each section. This data is shown above the divider line in the focus view. Context information for the various features is displayed below the divider line (see Figure 1b.) One shortcoming of this extremely compact design is that it is very difficult to label the features of the vertical CL (see future work).

Depending on the type of document various kinds of context information could be used. Ideally, the information displayed in the focus view depends not only on the document type, but also on the current work context.

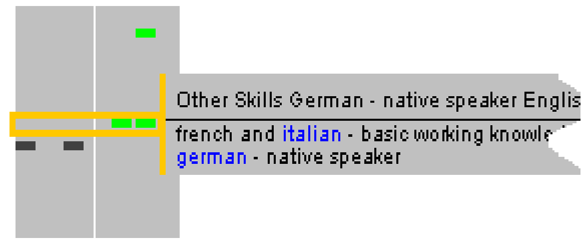

Figure 1b: Vertical Context Lens showing focus information for an entire section (paragraph). The text of the section start is shown above the divider line. The white vertical line separates permanent features from temporary ones (mined from recent search engine queries). Note: although we searched only for "italian" and "german" we also found french and english mentioned in the context.

Examples of potentially useful features are information on who accessed or modified parts of a document, the severity of spelling mistakes, or what kinds of information are embedded in a document (phone numbers, email addresses, or URLs.) For non-textual documents, such as video, entirely different features can be defined, for example the presence of background music, voice over etc., see (Ponceleon and Dieberger, 2000).

2.3 Eliminating the need to scroll

The Context Lens was designed as a navigation tool. We observed, though, that for certain tasks the CL can eliminate the need to actually navigate the document under consideration:

A CL provides access to relevant parts of a document, along with their immediate context and possibly some meta information. The CL represents this information independent of the base document. Should the information in the CL suffice to satisfy our information needs, there is no need to ever look at the base document itself.

We discovered this while experimenting with the CL. One of the authors executed a search on a person's name to find his phone number. As mentioned above, our implementation uses the query terms of recent search engine queries as temporary features. The search resulted in a typical search engine result page, displayed both as Web page, as well as in the CL. When brushing in the CL, the focus view showed the name and phone number of the person we were looking for. We had intended to look for the phone number on that person's home page, but the CL eliminated the need to access that Web page to find the phone number, see Figure 2. This works because certain types of information, e.g. names and phone numbers or email addresses tend to occur close together on certain types of Web pages. Thus, depending on the page type, a variety of information requests can be filled using the context information in a CL.

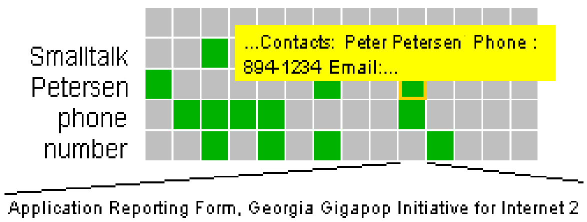

Figure 2: Finding a phone number in a CL (anonymized)

It can be argued that there are easier ways of finding a person's phone number, including submitting more specific queries. The point here is though, that the immediate context of keywords can be relevant, because certain types of information frequently occur close to keywords. Using a CL we can search for keywords and find the information we are actually looking for in the context of the keyword.

Many information-seeking tasks are based on vague information and the selection of result items can be done only with common knowledge and understanding of a search domain and the ability to recognize information even in unusual formats.

For example, when looking for people with degrees from "a well known east coast university" in a database of resumes it is impossible to specify a simple search query. However when searching for – say – a masters degree in a resume, the awarding university typically is mentioned close to the degree. In a CL we can easily find occurrences of the term "masters degree" and the immediate context will show the awarding university. This allows a CL user to brush through a list of matches and to scan the context for matching universities, thus leveraging his or her common knowledge.

2.4 Implementation

Because a CL contains much less information than the base document, the amount of information to download is sharply reduced. Instead of processing the base document on the client side, we employ a Web intermediary to perform the document analysis.

Intermediary based computation was introduced by Barrett and Maglio (Barrett and Maglio 1998; Maglio and Barrett 2000). An intermediary is a computational element located between one or more information providers and an information consumer. Intermediaries can perform transcoding, annotation, aggregation etc. In the case of the Web CL, we use an intermediary implemented using IBM's WBI toolkit (http://www.almaden.ibm.com/cs/wbi/).

In essence, the intermediary is a proxy server. It monitors requests to search engines and collects search terms as temporary keywords for the CL. It also inserts a Java applet into every Web page and analyzes pages for occurrences of keywords. Once the Java applet starts up, it sends a request for information to the intermediary. The intermediary responds with the data extracted from the page.

The current CL backend is very simple. The intermediary does not modify any requests for Web pages. As the page data comes in from the server, the intermediary inserts the applet tag and passes the page on to the browser. It also analyzes the page for the occurrence of keywords, highlights each occurrence, and marks it using an html anchor so that it is possible to link to the occurrence. The intermediary keeps a list of link anchors and also stores the immediate context of each keyword occurrence. Once the CL applet starts, it sends a request for information to the intermediary. The intermediary answers this request by sending the collected information about the page to the applet and the CL becomes active.

Currently we display both the page and the CL in the browser. Alternatively, the intermediary could create a minimal Web page containing only the CL applet. Selecting a section in the CL would then request just that section from the intermediary. This implementation would be especially useful for navigating large Web pages on devices with limited bandwidth, for example on wireless PDAs. The intermediary is directly connected to the Internet and easily can retrieve large documents, but the PDA with its low-bandwidth connectivity needs to retrieve only relevant parts of the document.

Currently the focus view shows only immediate context of keywords. In order to make the focus view even more useful, the intermediary could analyze the page to determine whether additional nearby information might be of interest, based on the user's profile. For example, if the focus contained a person's name, a close-by phone number or address, even if not in the immediate context, could be relevant. In this case, the focus view would display also this type of information. It is possible to carry this idea a step further and retrieve additional information from separate sources, like from a corporate phone book. A system of that kind was described in (Maglio and Farrell 2000).

2.5 Related Work

The grid representation of horizontal CL was inspired by Hearst's TileBar interface (Hearst 1995). Like the horizontal CL, a tilebar represents sections of documents as columns, and query keywords as rows. However, Tilebars appear in groups in the result set of a search query. Each result is associated with one Tilebar indicating where in the document search terms are located. A CL, on the other hand, is closely associated with only one document the user is focusing on. It provides access to focused context information through brushing and permits users to immediately navigate to that section.

SeeSoft used an abstract representation of long text documents, in particular source code. In this representation it indicated, for example, the author of a section by color (Eick, Steffen et al. 1999). Selecting a section displays the corresponding code fragment. Apparently, SeeSoft did not indicate other types of context information. As it shows simply the corresponding part of the base document, it misses out on the opportunity to reduce the amount of information the user has to process.

Further related work can be found in other lens techniques, like the TableLens (Rao and Card 1994), abstract representations like the Perspective Wall (Mackinley, Robertson et al. 1991) and the Information Mural (Jerding and Stasko 1995).

3 Evaluation

As described in section 2.3, one of the CL's strengths is to help answering vaguely defined queries based on the immediate context of search items. Such queries are common in a hiring scenario. Therefore, we chose a task in a resume evaluation scenario for our evaluation. We used a corpus of 255 resumes mined from the Web. The resumes were on a Web site which a search engine.

In this scenario the CL leverages common writing practice for resumes, which places, for example, years of experience close to a skill. Thus, the immediate context of the word "Java" is likely to mention years of experience or skill level. Similarly, resumes typically show the university at which a degree was achieved next to the degree. We compared using a search engine, which highlights keywords with a combination of search engine and either the horizontal or the vertical CL.

3.1 Hypotheses

We tested 4 hypotheses. 1: A horizontal CL is faster than just highlighting keywords in the search results. 2: The vertical CL is faster than the horizontal CL (because it shows more context at the same time.) 3: Using the CL people do not need to scroll. 4: We can answer vague queries without looking at the base document.

3.2 Procedure

We designed three conditions: Highlight: The search engine results with the keywords highlighted. This control condition represents typical search engine results. Horizontal: Highlighting plus a horizontal CL on top of the result page. Vertical: Highlighting plus a vertical CL at the top left of the result page.

12 subjects, researchers and visiting researchers at our lab participated in the study. Five participants were female; all participants had extensive computer and Web experience.

We prepared 3 query sets. Each consisted of 3 training queries and 6 queries for the study itself. The study was designed as a within subject test. To eliminate effects of ordering, we defined 6 groups of 2 subjects each, based on permutations of the ordering of the three conditions. We also permuted the query sets over the tools in the 6 groups to reduce the effect of the query set on the results.

During each section of the experiment we first introduced the general setup, described and demonstrated the tool and permitted users to familiarize themselves with the tool till they felt comfortable with it. We then asked participants to execute the training queries. We pointed out possible issues, shortcuts and the most effective way to use this particular tool for the task at hand. Typically the training sessions lasted about 5 minutes each. All participants rapidly grasped how the CL works.

We then ran the subjects through the queries. We measured the time from the moment where the search results appeared on the screen till subjects selected a candidate by clicking on him or her.

In case the user scrolled or performed navigational actions (e.g. hitting the back button) we recorded these actions as one navigation action each. We defined a scroll action as continuous scrolling till participants started reading again.

The queries were formulated vaguely and designed such that often there was more than one valid choice (participants needed to find just the first match.) When participants selected a candidate that did not satisfy the query, we instructed them to keep looking and continued measuring time and navigation / scroll actions.

3.3 Results and Discussion

One striking result was the almost complete lack of scrolling activity when using a Context Lens. We observed a total of only 12 scroll actions for both conditions involving a CL. In most of these cases participants had selected a wrong person and began searching again by scrolling to the top of the page.

In the remaining cases, a peculiarity of Web browsers caused people to scroll: when linking to a section of a Web page, this section normally gets scrolled to the very top of the browser window. However, if the section happens to be at the very end of the page, many browsers do not scroll that item to the top. This confused participants, because they were confronted with a different section than they had selected in the CL.

We observed that most participants made their choice using the CL alone: they navigated to their selection only because we asked them to do so. We recognized this from the fact that participants did the final click on a person (to indicate the choice) without any hesitation.

In condition highlight (no CL), we occasionally observed very long execution times in the range of up to 2.5 minutes, although the average search time for this condition was only 23 seconds. These extreme times occurred only when participants had prematurely dismissed a correct result. We decided to eliminate search times above 60 seconds (approximately mean + 2 standard deviations).

It is interesting to note that we had to eliminate 8 data points in condition Highlight, 9 from condition Horizontal, but only 2 from condition Vertical. As the Vertical condition provides a larger amount of context in the lens, we can see this as support for the utility of context in our task.

In the post-experiment interviews, almost all participants stated, that using a CL appears to be "obviously faster and more efficient". The average execution times confirm this observation, but the data were only marginally significant, as determined by a one way ANOVA, $F(2)=2.90$, $p=0.076$.

Means and standard deviations for the three conditions are shown in Table 1. A pair-wise T-test between conditions "Highlight" and "Horizontal" was significant $t(11)=2.56$, $p < 0.05$, which indicates that participants solved their tasks faster using the Horizontal CL than using only Highlighting. However a pair-wise T-test between condition Highlighting and Vertical was not significant. We also found that the average execution times for the Vertical condition were slightly larger than for Horizontal (see below).

	Highlight	Horizontal	Vertical
Mean	23.00	17.63	19.36
SD	13.75	11.74	12.55

Table 1: Average execution times (60s cutoff)

We also looked at the number of scroll actions and navigation actions in the three conditions (see Table 2). Note that in almost all cases there is one navigation action for each query in condition Horizontal and Vertical because of our study setup. This means, that in order to make a fair comparison, we need to compare the average 3.15 scroll actions in condition Highlight with an average of 1.01 in Horizontal and an average of 0.39 in Vertical, which represents a dramatic reduction in scrolling activity.

	Highlight	Horizontal	Vertical
Mean	3.15	2.01	1.39
SD	2.81	2.39	0.85

Table 2: Average numbers of scroll and navigation actions. Note that conditions Horizontal and Vertical have a minimum of one navigation action

In the horizontal CL tool the context is shown in black text on a yellow background and the term of interest is highlighted in red. This makes the context information very easy to read. As a vertical CL displays more context information in a much larger focus view, we believed that a large yellow rectangle would be awkward to look at. Therefore we used a light gray background and blue highlighting for the vertical CL. Unfortunately this made the Vertical CL harder to read. The fact that the vertical CL is about as fast as the horizontal one, although it is less readable and although there is much more text to read, indicates that the vertical CL might actually be a lot faster. This hypothesis needs to be tested in a follow-up study.

Table 2 shows that the amount of scrolling and navigating is sharply reduced in conditions Horizontal and Vertical. It also shows that although condition Vertical is slightly slower in our study, people are less likely to select a wrong entry and then have to go back to try again.

3.4 Informal Observations

We noticed that participants put a lot of trust into the tools we provided and came to rely strongly on them. Even when looking at the correct solution already, participants sometimes went back to the CL just to use it to navigate to the location they had just been at rather than scrolling and scanning the page.

Several participants suggested to include a filter, such that sections that do not match all keywords would not be shown at all. We disagree with this idea because one of our key goals is to support vaguely defined queries. Due to context a result that does not match all keywords can be a better match then a result matching every keyword perfectly. In the resume scenario, for example, when looking for people with C++ and Java skills, a C++ only specialist with many years of experience might be a better candidate than a student who programs both C++ and Java for only a year. A filter would hide this match, as the first resume does not mention both C++ and Java skills.

Side effects of keyword highlighting and formatting

We provided keyword highlighting in all three conditions, both on the Web page itself, as well as within the CL. We were surprised to observe participants look at the correct solution (indicated by them pointing with the mouse, a reading aid many people seem to use while scanning text), but then dismissing that item. An example was a query where we asked for a person who had been at a “Fairbanks Center”. Several participants looked at the correct item and sometimes even pointed at the words “Fairbank Center” with the mouse cursor, but dismissed the entry, although it was the only match.

We came to believe that keyword highlighting can cause premature rejection of correct results because a highlighted phrase can overshadow information directly adjacent to it. In the Highlight condition this likely is the main cause for the large number of extreme search times mentioned above. These extreme times occurred only when a correct solution was prematurely rejected (8 times in 72 queries). We observed similar behavior in the Horizontal CL, which uses strong highlighting of keywords (9 times), however premature rejection was practically absent in the vertical CL, which used a more subtle color scheme (2 times).

This prompts an provocative question: Highlighting key terms in search results is a well accepted technique and it works well when users are looking for exactly these key terms. However if the key terms act as guides towards context, highlighting appears to be more of a distractor and can hide information people are trying to find.

We believe that a detailed study of this observation is indicated, especially for tasks that require scanning of context information.

The current implementation of the Context Lens turns all context information into lower case text. For certain tasks, our subjects found this to be very confusing. US state abbreviations (e.g. AL for Alabama) are always written in upper case. If the same abbreviation “al” is written using lower case characters people tend to overlook it as meaningless. Similarly people overlooked the phrase “us patent” when looking for a “US patent”.

Learning effects

During the design of our study we repeatedly tested the experimental setup on one of our colleagues. Although we did not time these trials, we noticed that the subject got more proficient and noticeably faster over time. While all of our participants easily grasped how to use the CL, true mastery of the tool quite obviously requires practice.

Out of curiosity we asked two of our test subjects to perform two additional study sessions, each with one day in between. We varied the sequence of test conditions in the three tests but decided to use the same queries in the three sessions, because we had observed that the query sets feel very different depending on what tool is used to run them. Therefore we felt confident that a learning effect from the queries would be minimal – or at least small enough so we could ignore it for sake of the pilot study.

The data from the three sessions indicates that the execution times for the horizontal CL indeed tended to drop over the course of the three sessions. One of the two participants halved task reduction time from the first to the third session. At the same time, though, we found that the execution times for condition Highlight increased slightly for both subjects. We think that as the participants got used to the CL they became less confident in their search behavior using highlighting alone. A sharp increase in scroll actions in the Highlight condition (almost twice as many scroll actions) supports this theory. As the number of navigation actions dropped for both Context Lenses we think that participants got more confident using a CL over time.

4 Future Work

We introduced the CL concept in the context of Web searching, based on text context. We call such a CL a Text Context Lens. However, the CL idea is easily applicable to other media domains and media types. For example, we also built a Video Context Lens to navigate large amounts of video data, see (Ponceleon and Dieberger 2000). This system uses a vertical CL, sometimes dubbed a movieDNA because of its similarity to DNA prints.

The movieDNA shows key features of video segments. One possibility is to again indicate presence of keywords, for example in the transcript of a video voice over. Other features that can be represented are speaker changes, presence of music, camera angle, stress patterns in the speaker's voice etc. In the case of the video CL the focus view shows a poster frame of the video segment under consideration along with other context data, for example a transcript, if available. Current work on the Video Context Lens focuses on optimally representing focus information and on how to label features in the movieDNA.

Additional work is done to visualize and navigate event streams and chat room logs using CLs. In order to scale the CL to very large collections of documents we are developing hierarchical CLs, as described in (Ponceleon and Dieberger 2000). A key research issue in these systems is how to generate and represent information scent, when these features are not directly visible in a top level CL but buried in lower levels of the hierarchical CL.

5 Conclusions

We presented a focus+context visualization and navigation tool, called the Context Lens and described several possible forms of this class of tools. Context lenses are useful because they allow users to keep an overview of the locations of interesting items in a document while simultaneously providing access to very detailed context information.

The amount of information needed to visualize a document using a CL is much smaller than the information contained in the document as a whole. Combined with an intermediary performing document analysis, this allows previewing of large documents without downloading them as a whole.

In our user study we demonstrated that another advantage of the CL is to allow answering very general or vague queries that cannot be formulated well using Boolean search queries. Often the solution to these queries can be determined by studying the immediate context of search items, which is exactly what the CL provides.

Acknowledgements

We'd like to thank our study participants for their efforts and suggestions. We further would like to thank Team WBI, Chris Campbell, Eser Kandogan, Paul Maglio, Teenie Matlock, and Dulce Ponceleon. Christine Cheung provided the resumes.

References

Barrett, R. & Maglio, P. P. (1998). Intermediaries: New Places for Producing and Manipulating Web content. *Proc. of WWW7*.

Card, S. K., Mackinlay, J. D., et al., Eds. (1999). *Readings in Information Visualization*. San Francisco, CA, Morgan Kaufmann.

Dieberger, A. (1994). Navigation in Textual Virtual Environments using a City Metaphor, Ph.D. Thesis, Vienna University of Technology.

Eick, S. G., Steffen, J. L., et al. (1999). SeeSoft -- A Tool for Visualizing Line Oriented Software Statistics. *Readings in Information Visualization*. San Francisco, CA, Morgan Kauffman: 419-430.

Hearst, M. A. (1995). TileBars: Visualization of Term Distribution Information in Full Text Information Access. *CHI'95*. Dever, CO, ACM Press: 59-66.

Jerding, D.& Stasko J. (1995). The Information Mural: A Technique for Displaying and Navigating Large Information Spaces, *InfoVis95*, Atlanta, IEEE Press: 43-50.

Mackinley, J., Robertson, G., et al. (1991). Perspective Wall: Detail and Context Smoothly Integrated. *CHI'91*. New Orleans, Addison Wesley: 173-179.

Maglio, P. & Barrett, R. (2000). "Intermediaries Personalize Information Streams." *Communications of the ACM* 43(8): 96-101.

Maglio, P. P. & Farrell, S. (2000). LiveInfo: Adapting Web Experience by Customization and Annotation. *Adaptive Hypermedia and Adaptive Web based Systems*, Trento, Italy, pp. 144-154.

Pirolli, P. & Card, S. (1995). Information Foraging in Information Access Environments. *CHI'95*. Denver, CO, ACM Press: 51-58.

Ponceleon, D. & Dieberger, A. (2001). Hierarchical Brushing in a Collection of Video Data. *HICSS'34* (Hawaii International Conference on Systems Science), Maui, HI.

Rao, R. & Card, S. K. (1994). The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus+Context Visualization for Tabular Information. *CHI'94*. Boston, ACM Press: 318-322.

Russell, D, Stefik, M. et al. (1993). The Cost Structure of Sensemaking. *InterCHI'93*. Amsterdam: 269-276.