

Window Frames as Areas for Information Visualization

Staffan Björk and Johan Redström

PLAY Research Studio, Interactive Institute
 Box 620, SE - 405 30 Gothenburg, Sweden
 {staffan.bjork,johan.redstrom}@interactiveinstitute.se
 http://www.interactiveinstitute.se/

ABSTRACT

We describe how the frames of document windows can be used as areas for visualizing information about a document. A number of design examples are presented as illustrations of how users can be supported with contextual information about a document in a way that does not interfere with the visual presentation of the document itself. A *ChangeIndicator* is used as an example of how information can be mapped to basic design variables of a frame, such as its color. More complex visualizations can be achieved by mapping information about parts of documents to parts or details of the frame. This is illustrated with a *ReadabilityViewer*. Finally, the *ScrollSearcher* is presented as an example of how the results of processes and functions can be visualized in the frames of a document.

Keywords

Window frames, borders, scrollbars, contextual information

INTRODUCTION

In GUIs, a document is usually displayed in a window that separates it from the rest of what is displayed on the screen. In many applications, the frame contains icons for certain functionality, such as for formatting the text. To some extent information about the document, such as the number of pages, which page one is currently viewing, etc., is also presented.

The perhaps most elaborate features of document frames are the *scrollbars*. The scrollbar was originally designed to support navigation of pages larger than what can be displayed in the window by enabling the user to scroll up and down to reveal different parts of the document. Since there is mapping between the position of the handle in the scrollbar and what part of the text that is displayed, users

can also use the scrollbar to see what part of the document is displayed.

Over time, the scrollbar has been improved to give the user even more information that support navigation. Instead of having a handle with a fixed sized, the size is proportionally adjusted to show how much of the document is currently presented.

To summarize, window frames of documents are used for some forms of information presentation. With some exceptions, however, the focus is on representing functionality of the application rather than information about the document. In this paper, we describe examples of how the frames can be used to present contextual information about documents by means of abstract information visualization (cf. Redström et al. 2001).

MAPPING INFORMATION TO BASIC DESIGN VARIABLES

Mapping information to basic design variables such as the color of a frame, makes it possible to add a dimension of information representation to an existing design almost without increasing visual complexity (except for dynamic coloring). Other such basic design variables that we can map information to include the thickness of the frames, texture, shape, etc. Abstract visualizations based on general features of frames can be suitable when the information presented is aimed at supporting the user's awareness about something, without being too intrusive. As design example, consider the *ChangeIndicator*:

The ChangeIndicator

The *ChangeIndicator* (fig. 1) is a simple visualization that provides a measure of how much one has modified a text since it was last saved. Whenever a character is typed (including backspace and delete), the border surrounding the text slightly changes color. Such information might support the user's awareness of, e.g., when it is time to save the document.

In the *ChangeIndicator*, the border color changes from white to black by going through nuances of red, blue, and

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, and that copies bear this notice and the full citation on the first page.

To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

NordiCHI 10/02 Århus, Denmark

© 2002 ACM ISBN 1-1-58113-616-1/02/0010...\$5.00

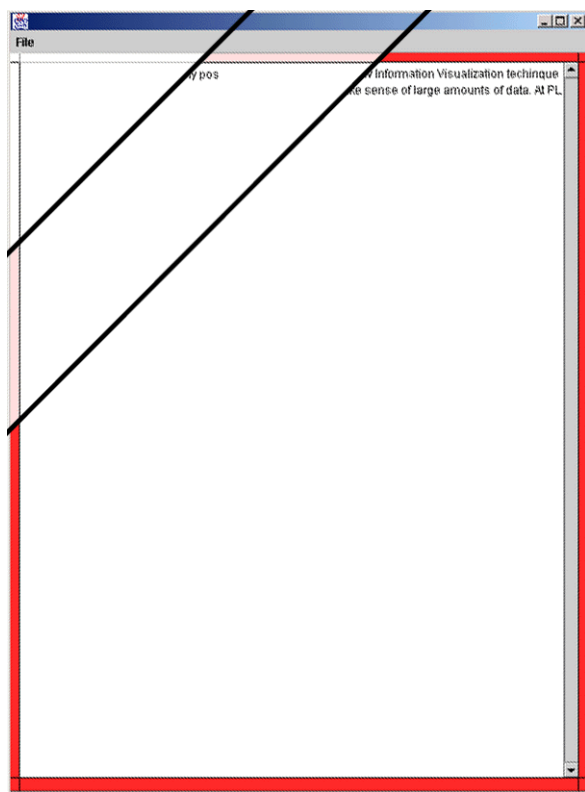


Fig. 1: The ChangelIndicator. Compound figure showing the color of the frame at three different steps as the color changes from white, via pink, to red.

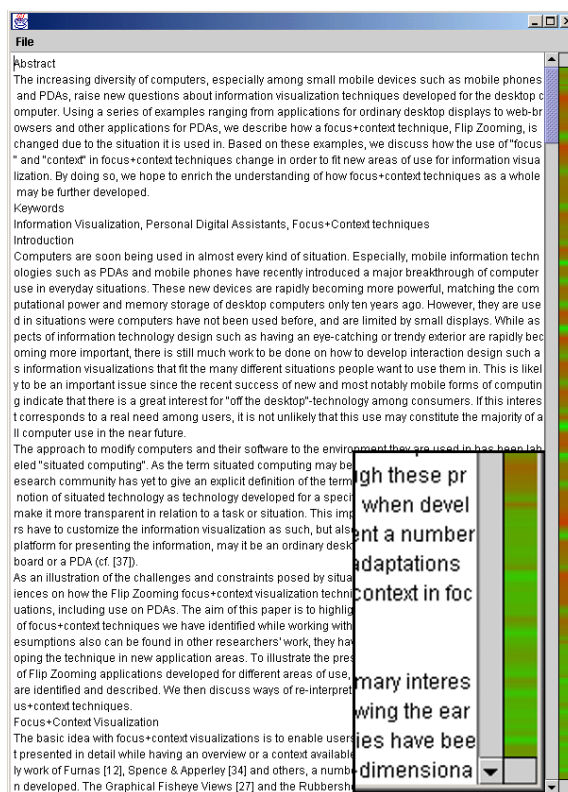


Fig. 2: The ReadabilityViewer. Green/light grey is mapped to easy paragraphs while red/dark grey is mapped to difficult paragraphs. (Close-up on the bottom right corner.)

green. The color sequences are consciously designed not to have “highlights” to suggest user actions, e.g., to urge the user to save the file.

In comparison to information about modifications of a document presented as numbers, text and icons, this is a more ambient visualization that the user do not have to pay explicit to. Instead, the user can learn to interpret this abstract information visualization over time by relating the changing colors of the frame to his or her working method and to the material being manipulated.

MAPPING INFORMATION ABOUT PARTS OF DOCUMENTS TO PARTS OF FRAMES

We can also use slightly more complex visualizations to present information about a document. Instead of mapping a property of the document as a whole to the general appearance of the frame, we can map information about details in, or parts of, the document to details in the appearance of the frame. Following the basic idea behind the scrollbar, we can map the vertical sides of the frame to the length of the document and represent information about

different parts of the document in the corresponding sections of the frame. Consider the following design example:

The ReadabilityViewer

The *ReadabilityViewer* (fig. 2) automatically provides the user with an overview of how “easy” or “difficult” it is to read different parts of a document according to some automatic readability measurement. This information is presented as shifting nuances in the coloring of the frame.

The presentation of the “readability” of the document is constructed by first calculating the readability index of each paragraph using a simple formula based on average word and sentence length and total number of sentences in the paragraph. (For the algorithms of a more advanced readability measurement, the Flesch Index, see Talburt 1984). This value is mapped to a color lying between pure green and pure red. The middle of the paragraph (in the scrollbar’s frame of reference) is given this color, and the space between these points are filled with a linear transgression from the two colors. The primary motivation for having smooth transitions between the paragraphs was that

the readability index itself should not be seen as an exact measurement, but rather as a support for getting an overview of how different parts of the document are related to each other in terms of readability.

This prototype only uses the right border. However, it would be easy to use also the left border, e.g., to present information about what is presently displayed within the window. Thus, it would provide more detailed information about the part currently in view (cf. the *ScrollSearcher* below).

Visualizations based on similar strategies for information representation could be used to support collaborative work with shared documents by means of presenting information about where and by whom changes have been made. Another possibility would be to present information about the history and development of the document by visualizing information about when the different parts were created or modified.

PRESENTING THE RESULTS OF PROCESSES AND FUNCTIONS

Still another possibility is to map information obtained from a process or function, such as performing a search for text string in a document, to the appearance of the frames. Here, the visualization can become a support for inspecting the results. The *ScrollSearcher* was developed to illustrate this:

The ScrollSearcher

The *ScrollSearcher* application (Björk 2001) is a simple text editor with a scrollbar that initially is indistinguishable from a normal scrollbar (fig. 3). However, when a search is performed, a result bar is shown next to the scrollbar. This bar shows the location of all the hits found using the scrollbar's representation of the document.

The *ScrollSearcher* technique couples the presentation of results with the navigation provided by the scrollbar. Users move to hit locations by simply dragging the scroller so that it is aligned with the indicator for the hit. This navigation style allows users to move between different hits using a direct manipulation technique, and allows users to choose in what order to view the hits.

As all hits are presented in relation to the document as a whole, users are provided with a view of the distribution of hits. This allows users to identify hits with characteristics usually hard to notice using simple search functions, e.g. groups of hits, hits that are isolated, and hits that are located in certain regions of the document such as the beginning or the end.

Additional searches are shown by placing more result bars next to the first, using color-coding and pop-up views (so-

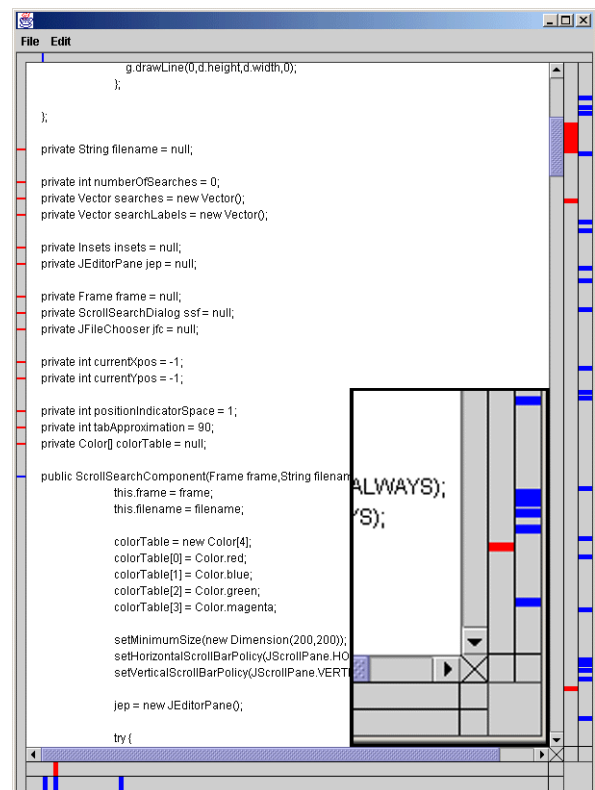


Fig. 3: The *ScrollSearcher* text editor after a search for the words *private* and *public* has been performed on a file containing java code. Searches are added to the right of previous searches and the results of each search is shown using a different color. (Close-up on the bottom right corner.)

called tool tips) to help users distinguish the different result bars. As all searches are visualized next to each other, they can easily be compared. This allows users to see where hits from different searches coincide; effectively showing the result of compound searches using Boolean expressions based on the search keys. Further, other information such as absences of hits, sequences between different searches, etc., can be found by visually comparing the different result bars.

RELATED WORK

Several techniques that expand the functionality of scrollbars and sliders have been developed, and many of them could easily be regarded, or modified to function, as visualization techniques that make use of the frame areas of applications.

TileBars (Hearst 1995) is a compact visualization of relative document length, query term frequency and query term distribution supporting information retrieval. Value bars (Chimera 1992) uses bars in the frame region of a list to show numerical value (e.g. file age or size) of elements.

The Alphaslider (Ahlberg and Shneiderman 1994) allows rapid selection of elements from enumerable data collections using a small user interface widget, and is natural to place in the border region of an interface. As an example of Instrumental Interaction (Beaudouin-Lafon 2000), a “search and replace” widget is introduced that presents search results inside a scrollbar.

The SeeSoft system (Eick 1992) uses a categorical slider that represents multiple files as thin rows and Data Visualization Sliders (Eick 1994) uses generalized sliders to present distributions and disconnected selections, both which could be coupled with document views by placing them in border regions of the application.

CONCLUDING REMARKS

There are clearly a number of design opportunities for using document frames to present information. Here, our focus has been on rather abstract forms of information visualisation in situations of use where contextual information can support the user’s general awareness about a document without interfering with the presentation of the document itself. The design examples described are not meant to be well-developed applications. Rather, they are demonstrators that, hopefully, can serve as inspiration for further development. Future work in this area therefore include both investigating this design space further, as well as developing more elaborate applications that can be put in use.

REFERENCES

- Ahlberg, C., and Shneiderman, B., 1994. The Alphaslider: A Compact and Rapid Selector. *In: Adelson, B., Dumais, S., Olson, J., eds. Proceedings of CHI'94 Conference on Human Factors in Computing Systems, Boston, Massachusetts, USA 24-28 April 1994.* ACM Press, pp. 365-371.
- Beaudouin-Lafon, M., 2000. Instrumental Interaction: An Interaction Model for Designing Post-WIMP User Interfaces. *In: Turner, T. et al. eds. Proceedings of CHI 2000 Conference on Human Factors in Computing Systems, The Hague, The Netherlands 1-4 April 2000.* ACM Press, New York, NY, pp. 446-453.
- Björk, S. 2001. Using Scrollbars to Explore Search Results. *In: Hirose, M. Ed. Proceeding of Human-Computer Interaction Interact 2001, Tokyo, Japan 9-13 July.* IOS Press, pp. 745-746.
- Chimera, R. 1992. Value bars: an information visualization and navigation tool for multi-attribute listings. *In: Bauersfeld, P. et al. Eds. Proceedings of CHI'92 Conference on Human Factors in Computing Systems, Monterey, California, USA 3-7 May 1992.* ACM Press, pp. 293-294.
- Eick, S.G. Data Visualization Sliders, 1994. *In: Szekely, P. Ed. UIST '94 Conference Proceedings, Monterey, California, USA 6 November 1994.* ACM Press, pp. 119-120.
- Eick, S.G., Steffen, J.L., and Sumner, Jr. E.E. 1992. Seesoft - a tool for visualizing line oriented software statistics. *IEEE Transactions on Software Engineering, 11 (18), pp. 957-968.*
- Hearst, M.A. 1995. Tilebars: Visualization of term distribution information in full text information access. *In: Katz, I.R. et al. Eds. Proceedings of CHI'95 Conference on Human Factors in Computing Systems, Denver, Colorado, USA 7-11 May 1995.* ACM Press, pp. 56-66.
- Redström, J. Skog, T. and Hallnäs, L. 2000. Informative Art: Using Amplified Artworks as Information Displays. *In: Mackay, W.E. et al. Eds. Proceedings of DARE 2000 (Designing Augmented Reality Environments), Elsinore, Denmark 12-14 April 2000.* ACM Press, pp. 103-114.
- Talburt, J. 1985. The Flesch index: An easily programmable readability analysis algorithm. *In: Proceedings of the Fourth International Conference on Systems documentation, Ithaca, NY, USA 18-21 June 1985.* ACM Press, pp. 114 - 122.