Visualisation and Categorisation of Cached Hypermedia Data*

F. Dridi, T. Hülsbusch, G. Neumann
Information Systems and Software Techniques
University of Essen, D-45141 Essen, Germany

Abstract: We argue that categorization and visualization of the explored information space (e.g. the web) is desired in order to improve the re-access of information. Traditional browsers support only mechanisms such as bookmarks and/or history lists in order to revisit and to reload documents. This paper presents an innovative tool (CineCat) that uses the client side cache to categorize and visualize the explored web space. CineCat supports filtering techniques based on categories to provide simplified views of the information space.

1 Introduction

The bookmark and history list mechanisms are available in most web browsers since they were introduced by Mosaic (NCSA (1998)). Bookmarks provide a means for the users to remember visited web pages in order to reload (recall) these pages at some later time. The history mechanism allows within a session to re-inspect (revisit) previously loaded web pages based on a stack model. It is interesting to note that many users have the misconception that the history list refers directly to the temporal ordering, in which pages have been previously loaded (see e.g. JONES, S., COCKBURN, A. (1996)). This paper argues that the persistent client side cache of a browser is – among its other purposes – a powerful instrument for both, revisiting and recalling of web pages.

Navigation in the web may be an arduous task because of the absence of a global categorization and visualization scheme for the enormous amount of available information. The client side cache of a browser represents the explored web space and is a good starting point for a categorization and visualization scheme.

This paper presents a tool called CineCat (HÜLSBUSCH, T. (1997)) that enables the user to define categories, and to categorize and visualize the cached documents and the link structure between these documents. The user

*Published in: Gaul, W. and Locarek-Junge, H. (ed), Classification in the Information Age, 387–394, Springer Verlag, 1999.

can navigate easily within the explored web space and revisit or recall web documents. CineCat includes several view options that can tailor efficiently the display of large and strongly interlinked information spaces.

The paper is structured as follows: In Section 2, we give a short introduction into web caching works and the cache hierarchy in the web. In Section 3 the functionality of the history list and bookmark mechanisms are discussed. Section 4 describes CineCat. Finally, Section 5 offers concluding remarks and a brief description of future work.

2 Cache Hierarchy in the Web

In the current Web, documents are identified by uniform resource locators (URLs) that refer to the location (host and directory) where a document is stored. This physical addressing mechanism requires the heavy use of intermediary caches to avoid the hopelessly repetitious transfer of interesting documents. Figure 1 shows a typical setup where documents from a distant server (Internet) are transfered through possibly multiple proxy servers (e.g. one at the access provider, one in a company or department).

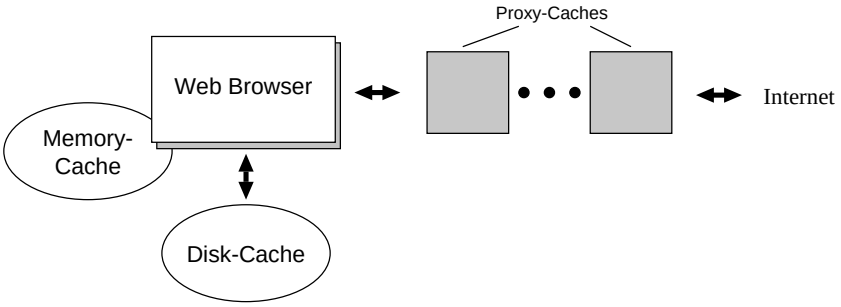

Figure 1: Cache Hierarchy in the Web

Usually there are two cache levels on the client side (Web Browser) as well: The highly volatile cache (memory cache) is primarily used for the history mechanism, where the user can use e.g. the *Back* button to see exactly the same document he saw when the resource was retrieved the last time (Fielding, R. et al. (1997)). The memory cache contains static and computed pages.

The browser side disk cache is a persistent cache that contains full or partial cache-able documents together with some meta data (URL, last modification date, content type, current size of cache entry, time of last retrieval, etc.). When the user requests a document the web client checks for an entry in the local cache and verifies the validity of the entry before the request leaves the client machine. Typically the size of the client side disk cache is bounded to a maximum size and cache entries are removed using a LRU (least recently used) mechanism based on the last access time. The client side disk cache

contains the most recently explored web space. For most browsers this important set of web pages is more or less invisible to the user.

One motivation of this paper was to experiment with the potential of the most recently explored web space and to examine how to categorize and to visualize this information space in order to improve navigational transparency. Visualization can help to develop a spatial understanding of the explored information space in order to locate information in the already-seen web space quickly. Filtering mechanisms based on categories can be used to provide different views on the information space (e.g. the information space relevant to one or more research topics). The local cache can assist off-line browsing and cooperative browsing when the cache is shared.

3 Support for Navigation and Categorization in Browsers

In the following we describe the functionality of the bookmark and history list mechanisms:

History List

The web pages visited during a session are kept in the linear history list, which can be used to *recall* these pages at some later time. The history list contains the shortest navigation path from the session start to the actual document. The *Back* and *Forward* buttons of a browser can be used to navigate backward or forward in this navigation history.

Note that in general the *Back* and *Forward* buttons do not control browsing in the temporal ordering of previously visited pages, but determine the currently displayed page in a stack of pages (JONES, S., COCKBURN, A. (1996)).

This stack of pages in the history list does not necessarily contain all previously visited pages: e.g. hitting twice back and activating a new link deletes two recently read pages from the stack. Entries are also lost, when the browser is terminated. The volatility of the history list entries entails that it is not feasible to provide additional information (such as meta-data) for these entries. Therefore the history list cannot support persistent categorization well.

Bookmarks

Bookmarks can be used to save document locations (URL) in order to *revisit* these pages at some later time. When a bookmark is added, the URL of the current document is placed into a file with other bookmarks. Note that these entries are typically as well in the cache. Entries in the bookmark file can be categorized hierarchically. Each entry belongs to one category. When an entry should be placed in more than one category, it must be duplicated.

In the remainder of the paper we will investigate how the functionalities for recalling and revisiting of pages can be provided through a single, more powerful mechanism that exploits the local cache. Every local disk cache provides already meta-data so categorization is a conservative extension. However, the local cache with all links is a very complex structure which has to be visualized to become explorable. The next section describes the innovative tool (CineCat) that analyses and extends the client side disk cache and gives the user the ability to categorize the contained documents and to visualize the visited information space.

4 CineCat – Cineast Cache Analysis Tool

CineCat is implemented in the Wafe (Neumann, Nusser (1993)) environment which includes support for several libraries like OSF/Motif and OTcl (Wetherall, Lindblad (1995)). The user interface of CineCat is implemented using OSF/Motif, its application logic is implemented using OTcl. CineCat uses the client side cache of the Cineast web browser (Köppen et al. (1997)). The Cineast browser, CineCat, and the Wafe software package are free available (WAFE (1998)). In order to visualize and categorize the cached data and its link structure we have to deal with the following aspects:

- Which objects can be visualized and categorized?
- How are these objects linked (kinds of links)?
- How can these objects be visualized, i.e. how can they be presented to the user in a clear way?
- Which icon set can be used to visualize the categories in a clear way?
- How should the visited web space be graphically displayed (layouting)?
- How can cache entries be categorized in a user friendly way?

In the following we will discuss these aspects and finally we will describe the CineCat user interface and its functionality.

4.1 Visualizing documents

A client side cache includes the visited web pages, images and other hypermedia data like audio files. A cache includes also meta data which is of minor importance for this paper. In order to give the user a suitable view of the cached documents and their link structure we only consider the web pages (HTML documents). We distinguish between server start documents (Hosts) and other documents (Pages) that are linked from the server start document.

Figure 2: Symbols for Server start documents and others

Figure 2 shows, how we visualize these two types of documents (two server start documents and a document). The icon depicted in the middle of each symbol designates the category of this document. Documents that are not already categorized can be visualized by using predefined icons. For example, in Figure 2 we use the icon in the middle to indicate an uncategorised server start document. For a document (Page) we use a different icon. The user can define his own categories and his own icons. All predefined and user-defined categories can be assigned to any document.

4.2 Visualizing the link structure

CineCat distinguishes between three kinds of links (see Figure 3):

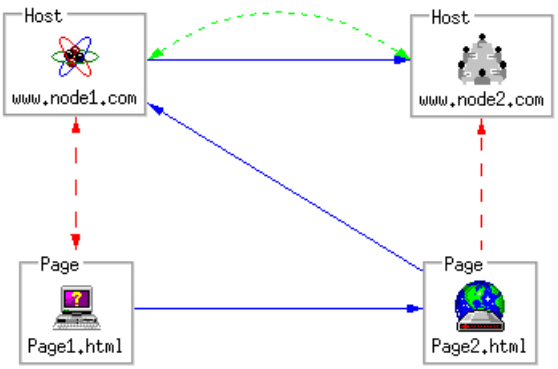

Figure 3: Links Types

- *Internal links.* An internal link references a document that exists on the same web server (link is displayed with a red dashed arrow).
- *External Links.* An external link references a document that exists on a different web server (link is displayed with solid blue arrow).
- *Help Links.* A help link binds two server start documents in order to indicate that there is a document on a web server that references another document on a different web server. This help link is needed when the web space of a web server is not completely visible in the current view, but it contains currently not displayed documents referring to other documents (ink is displayed with a dotted green arrow).

4.3 User Interface Structure

The user interface of CineCat contains four major areas (see Figure 4):

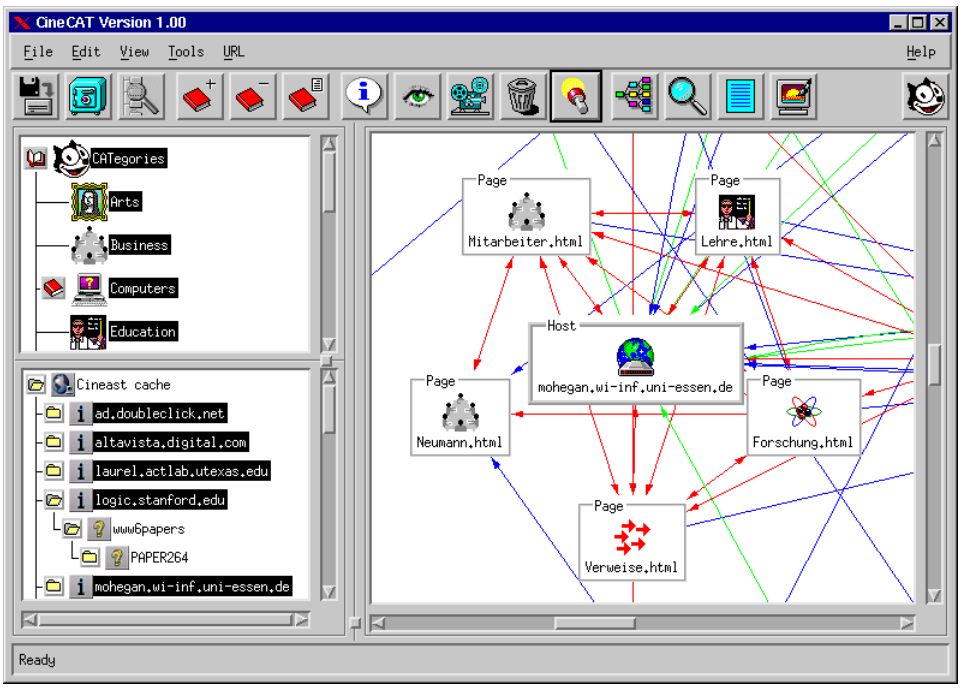

Figure 4: The CineCat User Interface

- *Menu bar and Tool bar.* These components in the top area allow the user to select functions depicted as icons or as pull down menus.
- *Category area.* This area is depicted in the top of the left hand side of the user interface. All user defined categories will be presented in this area. The user can define new categories and assign user preferred icons to them. Categories can also be deleted.
- *Cache content area.* This area is depicted in the bottom of the left hand side of the user interface. In this area the URLs every server start documents available in cache will be displayed. If an entry is selected the contents of the corresponding web server are displayed hierarchically.
- *Display area.* This area is depicted in the right hand side of the user interface. In this area a graph will be displayed that represents the link structure of the documents that belong to the selected categories and selected web servers. The user can assign a document in the display area to several categories by drag and drop. As result the icon of the first category of this document will be shown as part of the document symbol.

In Figure 4 the black entries in the "categories" and "cache contents" areas are currently selected. If the user selects a category from the category area then all corresponding documents and server start documents will be displayed in the display area. These documents will also disappear if the category is deselected. In the same way the user can select certain web servers to be visualized. This mechanism allows the user to reduce the number of visible items in the view area. This is a powerful search mechanism that shows the categorized documents in context.

In the view area a link can be selected to get e.g. its origin and destination. The user can select a document to view it with a simple HTML widget.

4.4 Layout of Interlinked Documents in the View Area

The layout of the documents and the link structure is a complex problem, since it is not trivial to present a large amount of documents that are strongly interlinked. We have integrated various view options in CineCat that focus on aspects like global view, detailed host info etc. The user can select these options from the menu bar (view menu) to obtain another graph that represents the link structure of the cached documents or those documents that belong to the selected categories and web server in the display area (see Section 4.3). In the following we describe these view options:

- *Detailed View.* This option allows the user to view the internal, external and help links (see Figure 4).
- *Selection Link types.* The user can select whether all links i.e. internal, external or help links (see Section 4.2) should be displayed (or any subset). This option can be used only with the detailed view option e.g. to visualize all the external links (or any subset) starting from a selected web server.
- *Global overview.* In order to reduce the complexity of the graph we use small icons for this option (see Figure 5). All server start documents will be displayed without links in a matrix form. All documents that are directly referenced from a server start document will be depicted in circular fashion around this document. The other indirectly referenced documents appear in a bigger circle around the server start document. By passing a matrix node with the mouse all the links related to this node will be temporary presented. This enables the user to explore the link structure of the cached data.

5 Conclusion

In this paper we have described CineCat which is a tool that gives the user the ability to visualize and categorize the explored web space. This web space is identical to the client side cache. CineCat can visualize the link

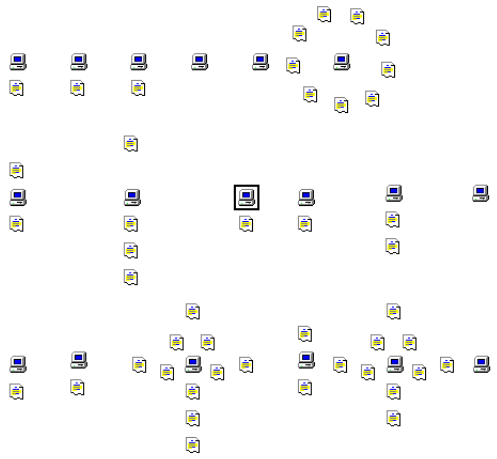

Figure 5: Matrix and Circular Layout

structure of the cached documents. By using the integrated view options CineCat is able to visualize a large and strongly interlinked cache (e.g. 2000 documents).

Since a cache can be used by several users, CineCat can be extended to support multi user visualization and (cooperative) categorization. This extension is useful in order to achieve cooperative browsing where a user can contribute URLs to shared Cache and the URL base.

From the security point of view this extension proposes some challenging problems. For example a user can have sensitive data in his local cache that should not be added to the shared space. The one-to-one correspondence between cache and explored web space is questionable in this case.

Another interesting extension would be to extend CineCat with automated classification methods in order to classify web documents in a (semi-) automated way.

References

FIELDING, R., GETTYS, J., MOGUL, J., FRYSTYK H., BERNERS-LEE T. (1997): *Hypertext Transfer Protocol HTTP/1.1*, RFC 2068.

HÜLSBUSCH, T. (1997): *Entwurf und Realisierung eines Cache-Visualisierungs-Programms für einen WWW-Browser*, Diplomarbeit, Universität GH Essen.

JONES, S., COCKBURN, A. (1996): *A Study of Navigational Support Provided by Two World Wide Web Browsing Applications* , Proc. of Hypertext '96, Washington DC.

KÖPPEN, E., NEUMANN, G., NUSSER, S. (1997): *Cineast – An extensible Web Browser*. Proc. WebNet 97, Toronto.

NCSA (1998): *NCSA Mosaic Home Page*.

http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/NCSAMosaicHome.html

NETSCAPE COMMUNICATIONS CORPORATION (1998): *Netscape Communicator*, http://home.netscape.com/

NEUMANN, G., NUSSER, S. (1993): *An X Toolkit Based Frontend for Application Programs in Various Programming Languguages*, USENIX Winter 1993 Technical Conference, San Diego, California.

WAFE Home Page (1998): *Wafe - A Programming Environment for X Toolkit Based Applications*. http://nestroy.wi-inf.uni-essen.de/wafe/

WETHERALL, D., LINDBLAD, C.J. (1995): *Extending Tcl for Dynamic Object-Oriented Programming*, Proc. of the Tcl/Tk Workshop '95, Toronto.