GROUPING ANNOTATING AND FILTERING HISTORY INFORMATION IN VKB

A Thesis

by

RAGHU CHAITANYA AKKAPEDDI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2003

Major Subject: Computer Science

GROUPING ANNOTATING AND FILTERING HISTORY INFORMATION IN VKB

A Thesis

by

RAGHU CHAITANYA AKKAPEDDI

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Frank Shipman (Chair of Committee)	Takashi Yamauchi (Member)
John Leggett (Member)	Valerie E. Taylor (Head of Department)

December 2003

Major Subject: Computer Science

ABSTRACT

Grouping Annotating and Filtering History Information in VKB. (December 2003)

Raghu Chaitanya Akkapeddi, B.E., Osmania University

Chair of Advisory Committee: Dr. Frank Shipman

History mechanisms available in hypertext systems allow users access to past interactions with the system and help users incorporate those interactions into the current context. The history information can be useful to both the system and the user.

The Visual Knowledge Builder (VKB) creates spatial hypertexts – visual workspaces for collecting, organizing, and sharing. It is based on prior work on VIKI. VKB records all edit events and presents them in the form of a "navigable history" as end-users work within an information workspace.

My thesis explores attaching user interpretations of history via the grouping and annotation of edit events. Annotations can take the form of a plain text statement or one or more attribute/value pairs attached to individual events or group of events in the list. Moreover, I explore the value of history event filtering, limiting the edits and groups presented to those that match user descriptions.

My contribution in this thesis is the addition of mechanisms whereby users can cope with larger history records in VKB via the process of grouping, annotating and filtering history information.

ACKNOWLEDGEMENTS

I would like to take this opportunity to offer my heart felt thanks to a number of people without whose help and support this thesis would never have been possible. I must start with my advisor Dr. Frank Shipman. I would like to express great gratitude to Dr. John Leggett and Dr. Takashi Yamauchi.

I have had the pleasure of working with a number of very talented and knowledgeable people at Center for the Study of Digital Libraries (CSDL). I would like to express special thanks to Dr. Hao-Wei Hsieh and Michael Moore. They have been tremendously helpful. I also want to thank all my fellow students and researchers in the department.

My parents have provided me with countless opportunities for which I am eternally grateful. Nothing I can say can do justice to how I feel about their support. I must thank a few friends without whose encouragement I might never have finished. So thank you very much - Kiran, Murali and Sarma.

Last but by no means least it gives me immense pleasure to thank my dearest friends - Anshul, John, Purna and Srikanth.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

History mechanisms available in hypertext systems allow users access to past interactions with the system and help users incorporate those interactions into the current context [Lee 1992]. The history information can be useful to both the system and the user. History mechanisms allow users to evaluate past work, to learn from past mistakes and to identify relationships between users and particular actions. History allows the system to model end-users and their tasks.

Long term use of applications that maintain history can generate a substantial amount of history information. In such cases, it can be difficult to locate relevant portions of the history. As with more general information location, filtering can reduce this problem. By reducing the quantity of information that is displayed or bringing attention to a subset of the information, filtering reduces the effort, both physical and mental, that a person spends interacting with the history.

Another approach to improving access to and use of large amounts of history data is through attached structure and meaning. Most existing interactive history mechanisms generate information that users may access but not edit or augment. They do not provide the option to record interpretations of the history. Examples of interpreting history include grouping a certain number of events, annotating events and assigning attributes to events. All of these operations assign structure and meaning to the existing history.
This thesis explores the use of filtering and attaching interpretations of history data within the context of the Visual Knowledge Builder, a spatial hypertext system.

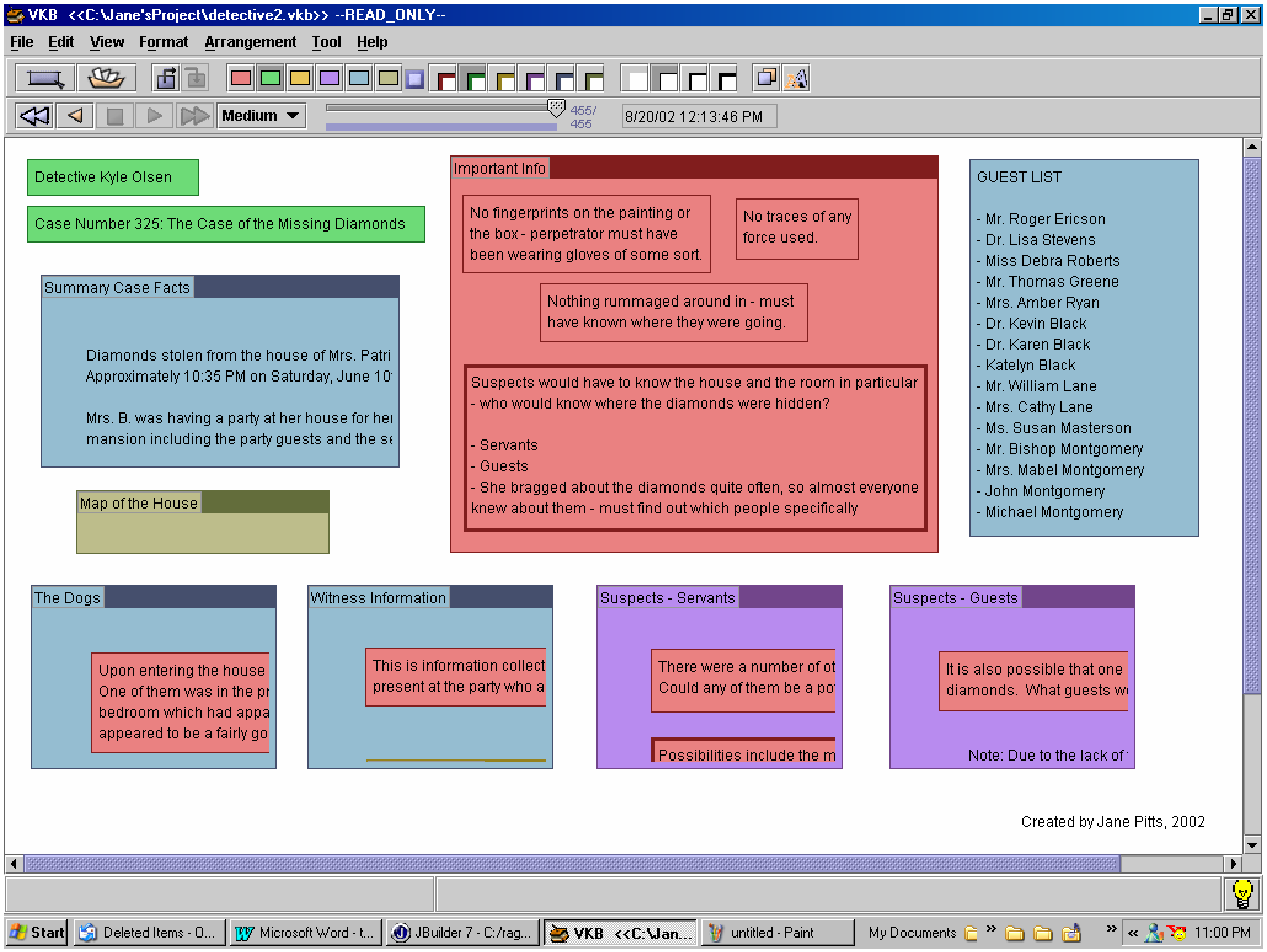
This thesis follows the style and format of *ACM Transactions on Information Systems*.

2. VISUAL KNOWLEDGE BUILDER

The Visual Knowledge Builder (VKB) is a spatial hypertext – a visual workspace for collecting, organizing, and sharing information [Shipman, Hsieh and Airhart 2001]. It is based on prior work on VIKI [Marshall and Shipman 1995]. VKB uses a hierarchy of two-dimensional *collections* that can contain other collections or visual *symbols* that represent information *objects*. Users modify visual attributes for collections and symbols to indicate their interpretation of the information. They can express relationships among various symbols by placing them near one another or by placing them in a collection. Broadly, this is a tool that aids the development of visual languages and supports collaborative knowledge-building tasks.

VKB records all edit events and presents them in the form of a "navigable history" as end-users work within an information workspace. The history mechanism in VKB is similar to the one implemented in INDY [Reeves 1993]. The interface allows users to quickly navigate through a lengthy history. VKB, like INDY, contains a feature to play forward and backward through a set of events and to return to specific events through a list of edits and through interacting with objects in the workspace. Navigable history supports the gradual interpretation of information by providing access to prior interpretive activities in a workspace and presenting them in a relevant context [Shipman and Hsieh 2000]. Such access is particularly valuable in activities where an information workspace is shared.

Basically there are four mechanisms in VKB to return to the prior state-playing the history in reverse, dragging the slider, selecting the start of a particular editing session, and returning to a specific edit event on a particular object [Shipman et al. 2001]. Figure 1 shows a sample VKB workspace.



Detective Kyle Olsen

Case Number 325: The Case of the Missing Diamonds

Summary Case Facts

Diamonds stolen from the house of Mrs. Patri
Approximately 10:35 PM on Saturday, June 10

Mrs. B. was having a party at her house for her
mansion including the party guests and the se

Map of the House

Important Info

No fingerprints on the painting or the box - perpetrator must have been wearing gloves of some sort.

No traces of any force used.

Nothing rummaged around in - must have known where they were going.

Suspects would have to know the house and the room in particular
- who would know where the diamonds were hidden?

- Servants
- Guests
- She bragged about the diamonds quite often, so almost everyone knew about them - must find out which people specifically

GUEST LIST

- Mr. Roger Ericson
- Dr. Lisa Stevens
- Miss Debra Roberts
- Mr. Thomas Greene
- Mrs. Amber Ryan
- Dr. Kevin Black
- Dr. Karen Black
- Katelyn Black
- Mr. William Lane
- Mrs. Cathy Lane
- Ms. Susan Masterson
- Mr. Bishop Montgomery
- Mrs. Mabel Montgomery
- John Montgomery
- Michael Montgomery

The Dogs

Upon entering the house
One of them was in the pr
bedroom which had appa
appeared to be a fairly go

Witness Information

This is information collect
present at the party who a

Suspects - Servants

There were a number of ot
Could any of them be a po

Possibilities include the m

Suspects - Guests

It is also possible that one
diamonds. What guests w

Note: Due to the lack of

Created by Jane Pitts, 2002

Figure 1: A sample VKB space

3. PROBLEM STATEMENT

Long-term use of applications that record history creates more information than can be dealt with. Some of the VKB workspaces have been in use for more than three years and include two to three thousand events. Locating particular activities in such a history is difficult in the current interface. Also, the existing version of VKB does not support user augmentation of the history, thus any effort locating and comprehending sequence of events must be repeated each time the need occurs.

My thesis explores attaching user interpretations of history via the grouping and annotation of history events. Annotations can take the form of a plain text statement or one or more attribute/value pairs attached to events in the list. Moreover, I explore the value of history event filtering. The large amount of accumulated history information described above will be filtered and presented according to user-specified queries.

4. HISTORY FRAMEWORK

History mechanisms in hypertext systems can be divided into those that record navigation events and those that record authoring events. Navigation history mechanisms record the movement of a reader through a hypertext and are included in common Web browsers such as Internet Explorer and Netscape Navigator. Such a record can be used to enable backtracking through prior navigations, to present “already visited” clues, to create paths, and to generate a personalized list of nodes [Tauscher 1996]. Authoring history mechanisms record events that edit the hypertext. Such mechanisms are often found in editors (e.g. Microsoft Word) to revisit prior versions of a document and to allow authors to undo and redo edits.

4.1 NAVIGATION HISTORY

As already mentioned in the introduction, history information can be useful to both humans and the system. In the context of the human, navigation history allows backtracking. For example, users may want to go back to the original position where they started. This helps in avoiding one of the problems of hypertext: “Getting lost in hyperspace” [Conklin 1987]. Navigation history can also reduce the disorientation problem by allowing the user to go back over the course their course of link traversal.

Nielsen [1990] discusses a structure-oriented backtracking feature as an alternative to the single-step backtracking found in most browsers. The system used its knowledge about the structure of the hypertext to provide a direct backtrack jump to the location where the user was likely to want to return. This prevents the user from having to backtrack one step at a time through a number of intermediate screens.

The system can use the navigation history to help in identifying patterns in the way the user utilizes the system. Such an understanding allows pre-fetching data that may be accessed by the user the in the near future. Another system use of navigation history is generating models of users’ interests. These models may be used to make navigation suggestions to readers. This analysis may benefit the author of the hypertext as well since a composite model of user interest can identify what information readers desire most. Finally, analysis of navigation patterns can point out aspects of a hypertext that create confusion or common "wrong turns".

Wexelblat [1998] describes one such application. “Footprints” uses information from web server logs to see where users have gone and then processes that information into various displays. Footprints also supports a list tool that provides a scrollable list of titles for pages visited by the user. The list is also navigable. Double clicking on any title brings up the corresponding page in the browser.

Thuring et al. [1995] refers to a presentation interface that shows the history of a hyperspace reading session by chronologically listing each node that has been visited during the session. It also shows the number of hierarchical nodes of the document and reader’s current position within that hierarchy.

Kashihara et al. [2000] encourage users to reflect on what they have constructed during exploration in hyperspace. The reflection involves rethinking their exploration process that they have carried out since it has a great influence on their knowledge construction. Their “interactive history” provides users with their annotated exploration history with exploration purposes. The interactive history system allows users to view and reconstruct the exploration history in order to rethink the exploration process that they have carried out so far. Users are given an option to annotate the nodes that they are exploring so that it helps the exploration process.

4.2 AUTHORING HISTORY

Authoring history can be very informative to both authors and readers in terms of adding context to information. Edit history information helps users interpret ambiguous materials by providing access to the context of an edit. Also, reviewing a sequence of edits provides insight into the thought processes and intentions of the author. Documents that include their authoring history provide readers with a notion of the document’s

constructive time [Shipman et al. 2001]. In addition to its value for interpreting content, the record of edits also allows authors to undo and redo actions, to return to prior versions of their hypertexts, and to review edits made by collaborating authors.
The system may use authoring history to identify common patterns of edits. Using programming by demonstration techniques, these patterns can be the basis for macros, or the system offering to complete menial tasks for the author. Here, the system learns new behavior by analyzing the history information created by users working in the system over a period of time.

An example of authoring history use in hypertext comes from NoteCards [Trigg et al. 1986]. NoteCards users can see a list of history events on the screen while they are browsing through links and nodes as mentioned in [Utting and Yankelovich 1989]. Users can annotate the history list by adding text at any place in the window. From the history list, users can select an item in the list and spawn a "minibrowser" which shows that selected item in more detail.

CoVer is a hypermedia version server [Haake and Haake 1993]. It represents the various versions of the same object through multi-state objects (mobs). The "derivation history" generates an unconnected graph structure over all versions of documents independent of their structuring into mobs. CoVer offers several options for "history tracking". The application can explore the historical development according to the creation date of versions, restrict the view to versions to those created by a specific author, or use any other attribute that suggests a comprehensible view over the version. This concept of "restricting the views" to some particular versions is close to the concept of filtering as explored in this thesis.

Writing Environment (WE) is a hypertext writing environment that can be used to create both electronic and printed documents [Smith et al. 1987]. WE has an "online tracker" that can capture users' interactions with the system. That information is represented as a sequence of symbols with attributes that constitutes the history of the session. The system also has a replay function that replays all the events that occurred during the session in time proportional to the original session.

Hayashi and Sekuima [1993] demonstrated a mechanism that records users access histories as tree forms in their implementation of “Nelumbo”. Nelumbo has a WYSIWYG editor that displays not only the final version of the document but also the snapshot of various versions of the same document at any point in the writing process.

4.3 FILTERING

Filtering in history to view a subset of events can be a useful tool for information seekers trying to rebuild the thought processes of the past. The history list can be problematic when the number of items on the list is long or when people are looking for information that is not shown on the display [Tauscher and Greenberg 1997]. Being able to perform a search on the history list and its various attributes can reduce this problem. Examples of such searches may include finding an action done by a person, an outcome, or a pattern within a lengthy simulation sequence or across several such sequences. Specifying a search pattern of edits in the history is a non-trivial problem as the size of the history grows [Plaisant et al. 1999].

Determining which features a history will locate desirable information requires some understanding of the characteristics of information users will want to specify. In Reeves study of history use in design, people most often searched for information using the following access methods [Reeves 1993]:

(i) by participant: they remember person X doing some action.

(ii) by communication medium: people recalled what medium was used (eg. whiteboard, overhead transparency).

(iii) by time: people use relative time (“midway through the meeting”) and clock time (“we only got through item 1.2 by 5 o’clock”).

(iv) by relation to other events: people used events as markers before/after other events.

The filters being investigated in this thesis include:

(i) Filtering on the basis of the user, that is, the participant.

(ii) Filtering on the basis of a time that particular event has occurred.

The other two access methods as presented in [Reeves 1993] are beyond the scope of this thesis.

[Klemmer et al. 2002] present a history interface that displays a history of the design using thumbnails. Each thumbnail presents the contents of the board at the time of capture. To support the user in determining what has changed between adjacent frames, the developers highlight the elements that were altered in the most recent frame. Users can opt to see only a set of thumbnails by using a filtering interface – the various filters provided by the system include by time, by note and by author.

4.4 HISTORY APPLICATIONS

This chapter has shown that there are a variety of applications of history data in hypertexts. Table 1 summarizes the above discussion by identifying applications of authoring and navigation history for readers, authors, and the system.

Table 1: Uses of authoring and navigation history by authors, readers, and the system.

	Hypertext Author	**Hypertext Reader**	**System**
Authoring History	Return to prior edits, Replaying edits of others, Undo/Redo	Context for interpreting the hypertext content	Macro generation via programming by demonstration
Navigation History	Information about reader preferences and confusion	Backtracking, "Already visited" clues	Pre-fetching and suggestions based on models of reader interest

5. HISTORY IN VKB

The Visual Knowledge Builder (VKB) includes navigation history to provide readers a view of the writers time [Shipman and Hsieh 2000]. While a hypertext document is being authored VKB records events in the hypertext history and provides methods to access prior states of the hypertext. The reader can play forward or backward in the authoring process. Analysis of VKB workspaces indicates navigable history supports (1) learning and interpreting authors work practices, (2) recognizing patterns of activity in information space, and (3) disambiguating specific actions and context. Hypertexts that include authoring history add a notion of constructive time for their readers.

Navigable history implies the ability for the reader to move back and forth between the current and prior states of the document. The potential value of navigable history mechanism in VKB became apparent in uses of its precursor, VIKI. When a VIKI space was collaborative, being able to understand what has occurred since the last time a user has worked on the space is an important issue. In one such collaborative task, a “History” collection was created as a location for notes explaining the activities of each session of VIKI use.

There are four mechanisms in VKB to return to a prior state: playing the history in reverse, dragging the slider, selecting the start of a particular editing session and returning to a specific edit event on a particular object [Shipman et al. 2001].
The interface for two of these mechanisms is found on the history bar, located under the visual edit bar in the interface as shown in Figure 2. The left side of the history bar has controls similar to a VCR interface, providing the ability to play forward and backward through the changes to the workspace in either a step-by-step or continuous play manner. The system allows the user to select one of five speeds of playback.

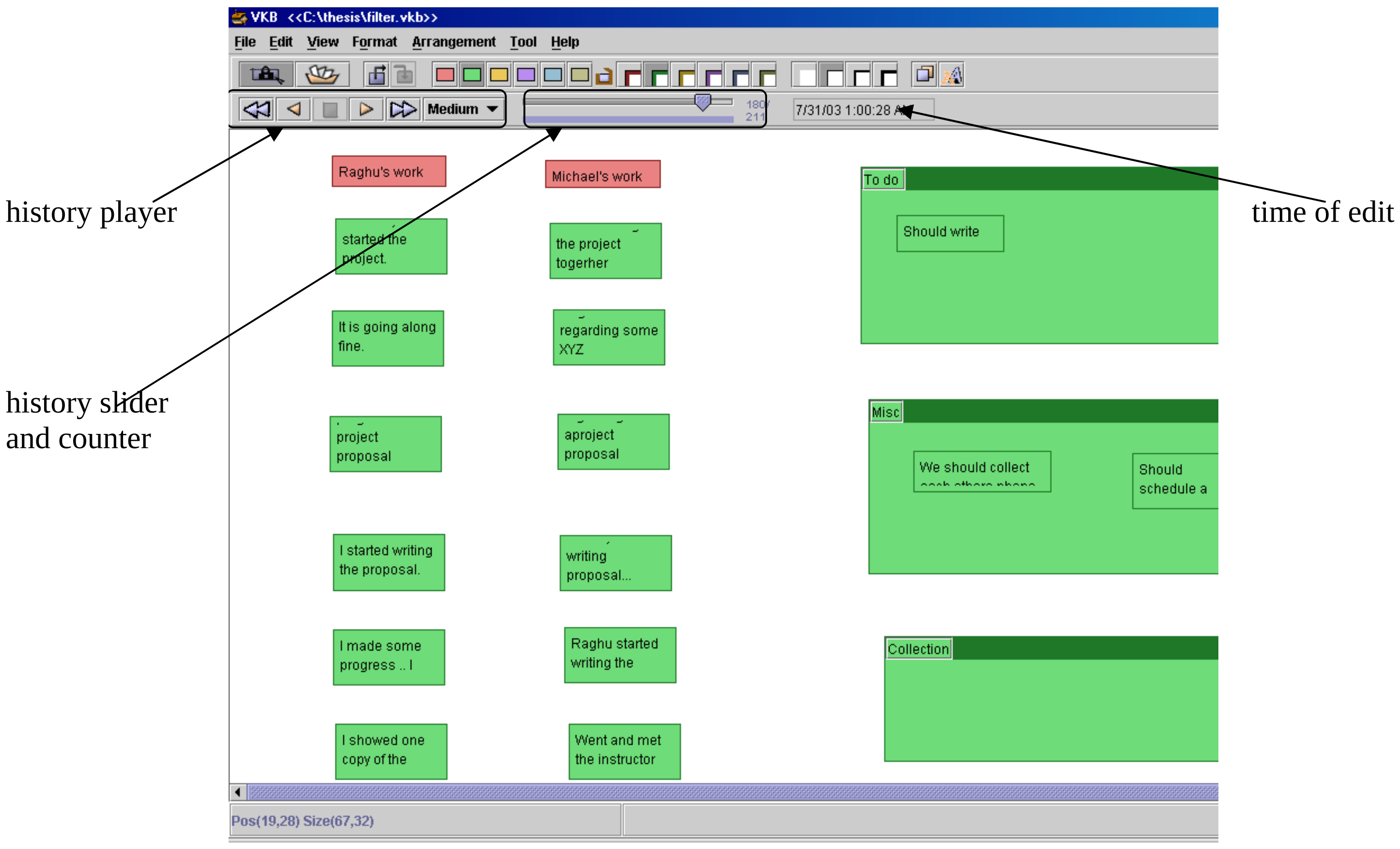
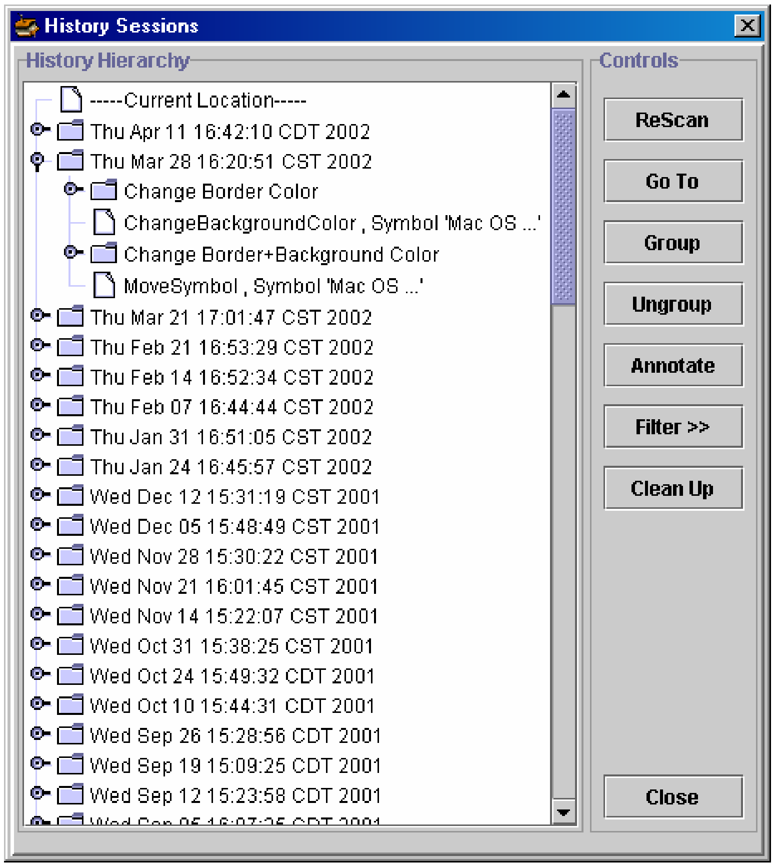


Figure 2: History Mechanism in VKB

The middle area of the history bar includes a slider for quickly navigating forward and backward through a lengthy history. The position of the slider is relative to the number of edits in the history, thus providing feedback on where the currently displayed state is relative to the cumulative effort put into developing the workspace and how fast the current play-forward/playback is moving relative to its history. The number of events in history and the number for the event that lead to the current state are shown next to the slider. Figure 2 shows the project space as it appeared after 180 edit events out of a total of 211. The time and date of the event that left the workspace in the displayed state is provided on the right of the history bar. This provides the absolute time of the modifications to the document. By noticing the date and time of changes through playback or while moving the slider, the user can determine the workspace's age and pace of changes.

Gaps between time stamps of the various events in the history also provide a sense of the length and frequency for authoring and editing sessions. VKB uses gaps in the history to recognize editing sessions. Figure 3 shows the editing sessions for a sample workspace.



```
-----Current Location-----
Thu Apr 11 16:42:10 CDT 2002
Thu Mar 28 16:20:51 CST 2002
    Change Border Color
    ChangeBackgroundColor , Symbol 'Mac OS ...'
    Change Border+Background Color
    MoveSymbol , Symbol 'Mac OS ...'
Thu Mar 21 17:01:47 CST 2002
Thu Feb 21 16:53:29 CST 2002
Thu Feb 14 16:52:34 CST 2002
Thu Feb 07 16:44:44 CST 2002
Thu Jan 31 16:51:05 CST 2002
Thu Jan 24 16:45:57 CST 2002
Wed Dec 12 15:31:19 CST 2001
Wed Dec 05 15:48:49 CST 2001
Wed Nov 28 15:30:22 CST 2001
Wed Nov 21 16:01:45 CST 2001
Wed Nov 14 15:22:07 CST 2001
Wed Oct 31 15:38:25 CST 2001
Wed Oct 24 15:49:32 CDT 2001
Wed Oct 10 15:44:31 CDT 2001
Wed Sep 26 15:28:56 CDT 2001
Wed Sep 19 15:09:25 CDT 2001
Wed Sep 12 15:23:58 CDT 2001
```

Figure 3: History Session Dialog

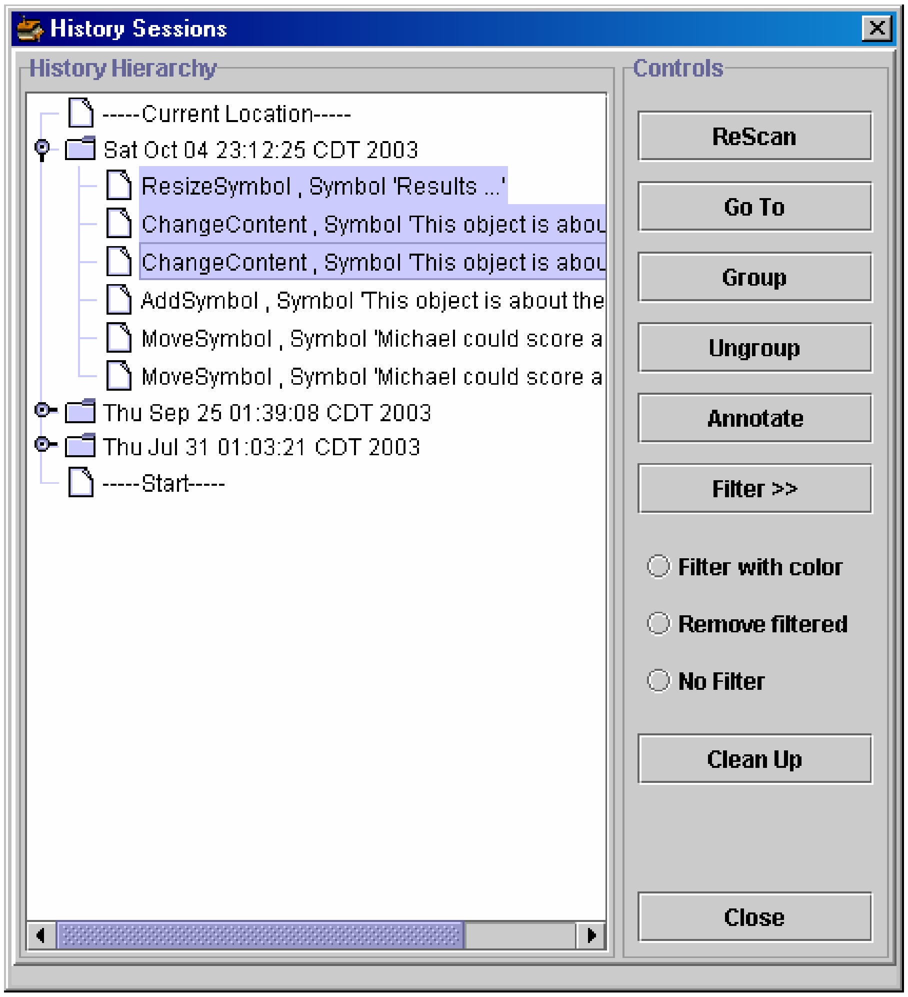
6. ENHANCEMENTS TO THE HISTORY

This section describes enhancements provided to the existing VKB history mechanism aimed at supporting users interacting with large histories. The enhancements discussed in this chapter are (1) Grouping, (2) Annotating and (3) Filtering.

6.1 GROUPING A SET OF HISTORY EVENTS

As shown in figure 3, the set of history of edit events is shown as a vertical list with various editing sessions visually marked and differentiated. One way the user can add contextual information to this list of events is by grouping a set of those events and providing a meaningful name to the group. The user created groups can further be grouped. The system generated session nodes are treated like any other group of events and can also be grouped.

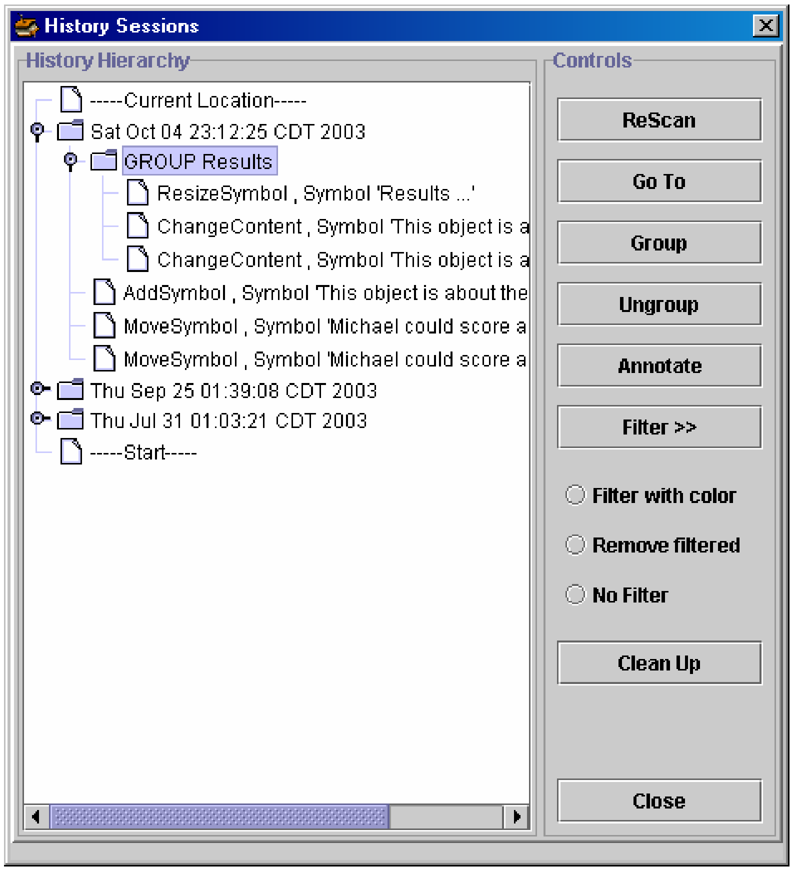
In this operation, a set of contiguous history events is selected by the user and the "Group" button to the right is clicked. Then the user is prompted to enter the name for the newly created group. After the grouping operation is performed the newly created "group event" replaces the list of grouped events that become a subset of the new group event. Figures 4a and 4b show before and after such grouping. The process of creating the group is treated as a single event. The newly created group event automatically attains some information that might be useful to the end-user and this will be discussed later in the "Annotating history events" section.


History Sessions

History Hierarchy

-----Current Location-----
Sat Oct 04 23:12:25 CDT 2003
ResizeSymbol , Symbol 'Results ...'
ChangeContent , Symbol 'This object is abou
ChangeContent , Symbol 'This object is abou
AddSymbol , Symbol 'This object is about the
MoveSymbol , Symbol 'Michael could score a
MoveSymbol , Symbol 'Michael could score a
Thu Sep 25 01:39:08 CDT 2003
Thu Jul 31 01:03:21 CDT 2003
-----Start-----

Figure 4a: Set of events to be grouped selected



-----Current Location-----
Sat Oct 04 23:12:25 CDT 2003
GROUP Results
ResizeSymbol , Symbol 'Results ...'
ChangeContent , Symbol 'This object is a
ChangeContent , Symbol 'This object is a
AddSymbol , Symbol 'This object is about the
MoveSymbol , Symbol 'Michael could score a
MoveSymbol , Symbol 'Michael could score a
Thu Sep 25 01:39:08 CDT 2003
Thu Jul 31 01:03:21 CDT 2003
-----Start-----

Figure 4b: Those events grouped and new node named as Results

The option of “ungrouping” the set of grouped nodes is also provided to the user. The ungrouping operation is performed by selecting the group node in the tree structure and clicking on the “Ungroup” button in the same interface.

The process of grouping and its use can be better understood through the following example.

Figure 4a and Figure 4b represent the change in the VKB History Interface when the user groups a set of history events. The first three highlighted events in the history interface presented in Figure 4a represent a set of events that are related to the result of the story being authored in the VKB workspace. The whole list indicates the various operations the author performed in building the story being told by the workspace. Now, the user can use the grouping operation implemented and group the events that are related as shown in figure 4b. This adds context to the cryptic history events and also adds readability to the interface.

6.2 ANNOTATING HISTORY EVENTS

The various edit events in the history session interface can appear so repetitive and cryptic to the end user that events cannot always associate with the context or task in which the event occurred. Thus annotating history events will help the end user to explicitly add context. This allows the addition of personal interpretations to the various events and in turn helps the user understand the VKB workspace better.

The user can annotate events and groups of events in the history interface in two ways:

(1) As shown in figure 5 the user selects the appropriate event in the interface and by right clicking opens a popup menu. Selecting Annotate from the menu opens a dialogue for free form text entry that is associated with that event (figure 6). This text can be anything that the user thinks is appropriate and useful in understanding the context or in comprehending the history of a workspace.

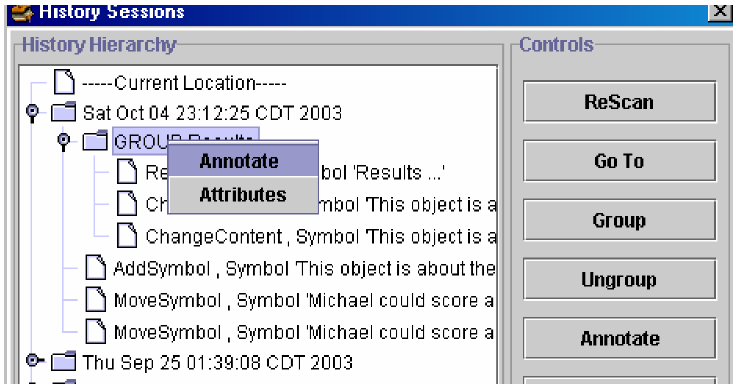
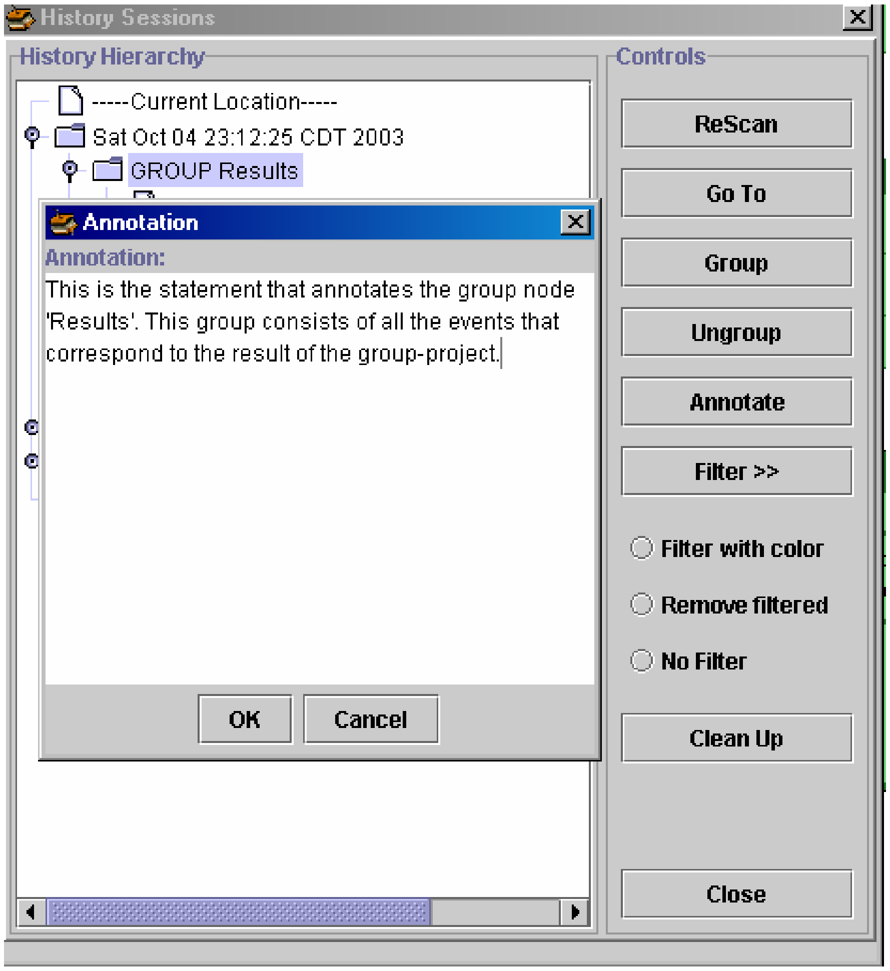

History Sessions
History Hierarchy
-----Current Location-----
Sat Oct 04 23:12:25 CDT 2003
GROUP Results
Annotate
Attributes
Re
bol 'Results ...'
Ch
mbol 'This object is a
ChangeContent , Symbol 'This object is a
AddSymbol , Symbol 'This object is about the
MoveSymbol , Symbol 'Michael could score a
MoveSymbol , Symbol 'Michael could score a
Thu Sep 25 01:39:08 CDT 2003
Controls
ReScan
Go To
Group
Ungroup
Annotate

Figure 5: Popup menu for annotating history events



This is the statement that annotates the group node 'Results'. This group consists of all the events that correspond to the result of the group-project.

Figure 6: Textual Augmentation of a history event

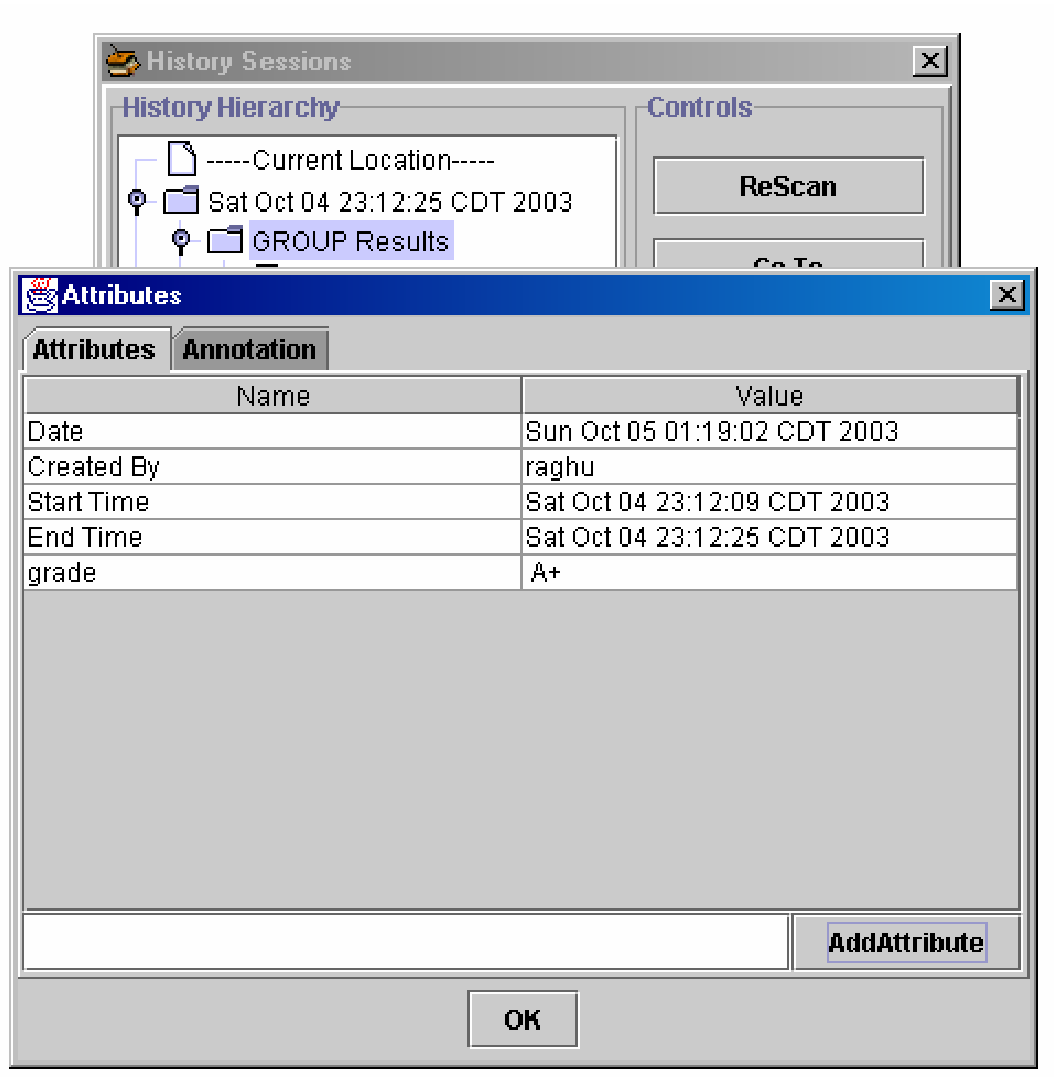
This process of annotating is also referred as textual augmentation.

(2) The second mechanism of annotating history events is to add attributes to the various history events. An attribute is defined in VKB as a name-value pair. When the option “Attributes” is selected in popup menu shown in figure 5, the interface to add attributes appears (figure 7). The user can add attributes for a single history event or a group of events.

As already discussed in “Grouping the history events” section the newly created group node has some attributes implicitly added to it like the time of creation, the person who created it, the timestamps of the first and last nodes among the “grouped nodes”.

The following example illustrates the use of history annotations in VKB.

As shown in figure 3 the user is displayed an interface that has various history events and the label associated with each such history event is very cryptic and is usually very difficult to comprehend the context in which the event occurred from that label. The user might want to add some text to help him understand the context in which the event occurred. This can help me comprehend the workspace better when he is looking at how that particular workspace evolved. For the event labeled “GROUP Results” the user may add, “This is the statement that annotates the group node 'Results'. This group consists of all the events that correspond to the result of the group-project.” This helps him understand the context better and clarifies his thought process.



Name	Value
Date	Sun Oct 05 01:19:02 CDT 2003
Created By	raghu
Start Time	Sat Oct 04 23:12:09 CDT 2003
End Time	Sat Oct 04 23:12:25 CDT 2003
grade	A+

Figure 7: An interface to add attributes to a history event

6.3 FILTERING IN THE HISTORY

Filtering in history can be a useful tool for information seekers trying to rebuild the thought processes of the past. Sometimes, the user can be overwhelmed by the amount of history information presented to him. By specifying certain criteria the user can see only the events and group of events that satisfy the criteria.

The interface implemented in VKB allows users to filter the information in the following ways (figure 8).

(1) The user can specify a person's name and view only those history events performed by that person.

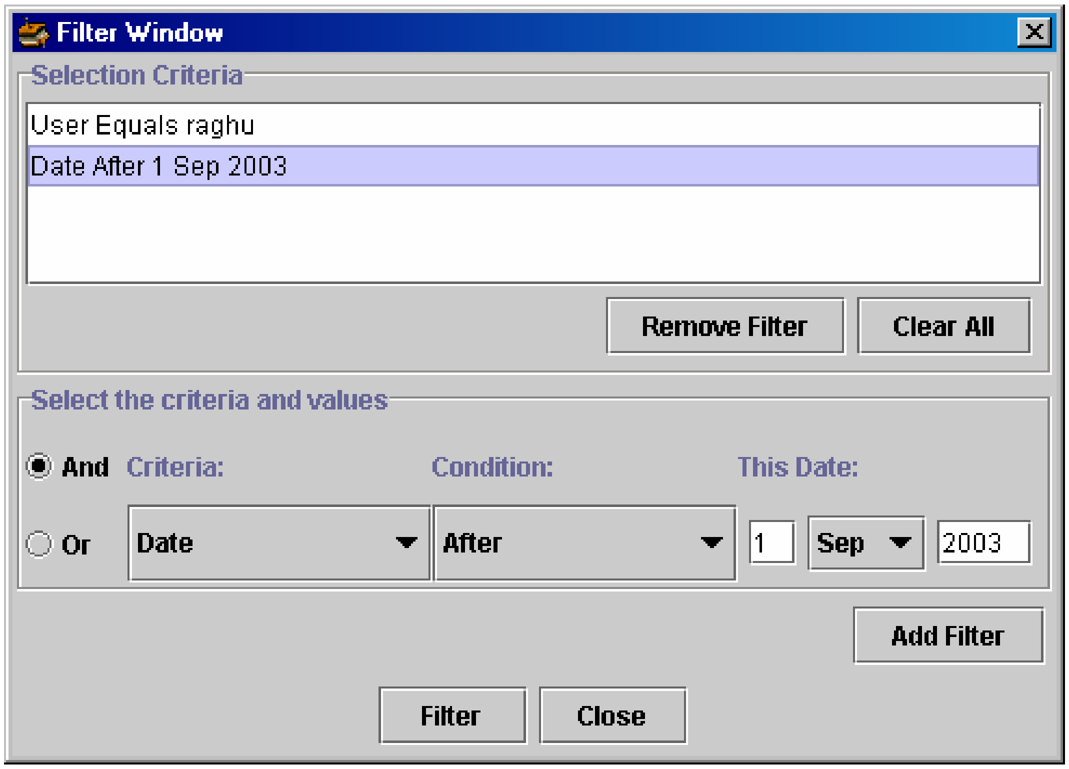
(2) The user can specify a date and can view all the history events that occurred on that day.

(3) The user can specify a particular date and view all the events that occurred before or after that day.

The user can apply more than 1 of the aforementioned filters simultaneously and filter the history accordingly. The unfiltered history list can be seen in figure 9a.

Filtered history information is viewed in two ways.

The user can see the entire history list with events matching the filter highlighted or visually marked (figure 9b). This alternative identifies matching events in the context of the total history. This context helps the user understand the filtered history better especially when the history information is not overwhelmingly large.



User Equals raghu
Date After 1 Sep 2003

(● And) Criteria: Date Condition: After This Date: 1 Sep 2003

Figure 8: History Filtering Interface

Alternatively, the user may choose the history list to show only the matching events and groups (figure 9c). This option is especially useful when the history information is very large since scrolling through thousands of events takes time and generates overhead for the end user.

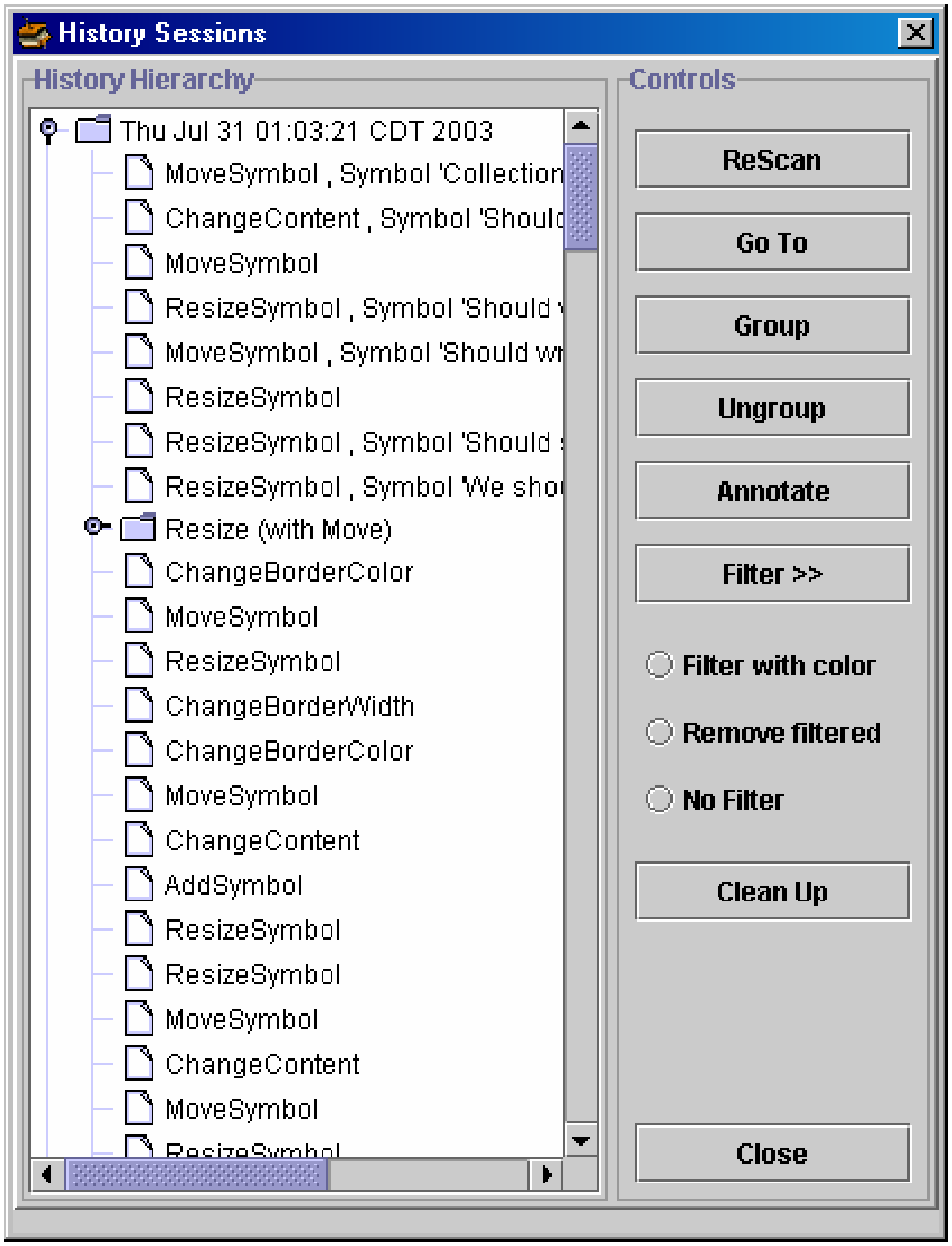
The user can toggle between filtered and unfiltered views using the radio buttons (figure 2). There are three options (1) either to see the complete set of events (2) Or to see complete list along with the filtered events visually marked/highlighted (figure 9b) and (3) the option allows the user to see only the filtered events (figure 9c).

A scenario to describe filtering history information is as explained below.

Consider 2 students working on a course project and using VKB workspace for sharing notes that describes what each person has done. At the end of the project, the instructor wants to see what each student has done and how the students worked together. By using a filter for each student with the highlight option, the instructor can see the pattern of collaboration among the students. The instructor can then see only a single student’s work by switching to the "Remove Filtered" view.

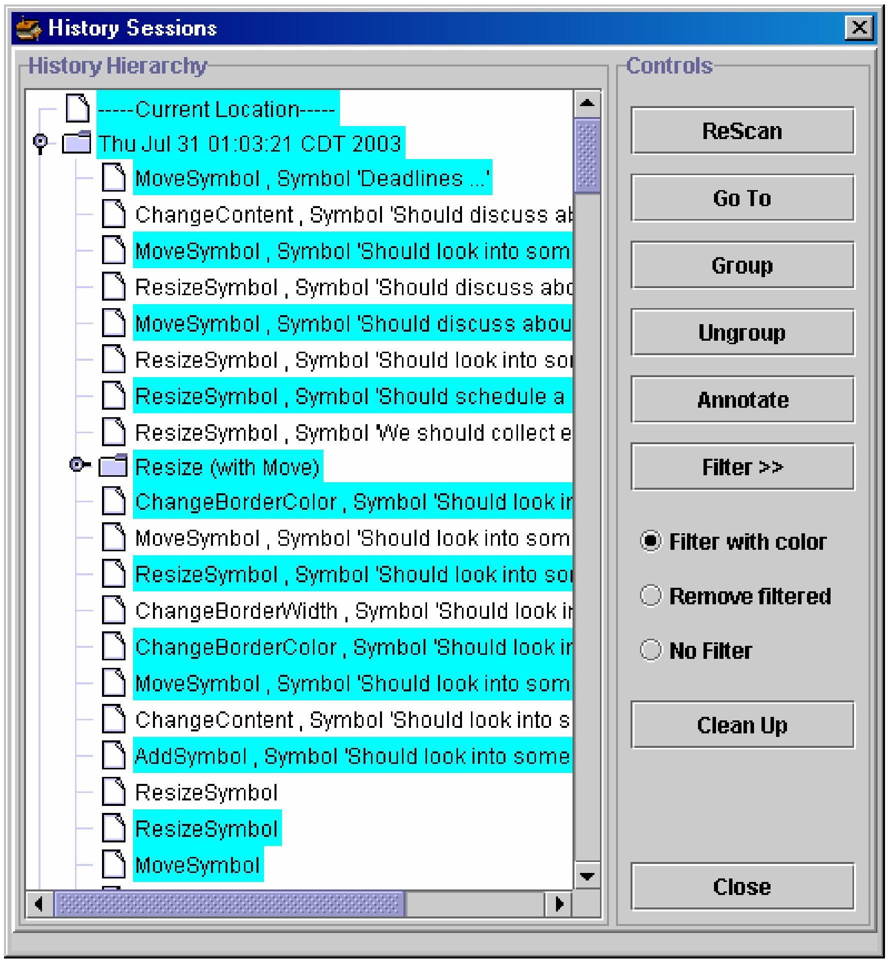
If the instructor wants to see the work done in the last 2 weeks before the submission deadline he can use the date filtering mechanism and set the filter to the date corresponding to the last 2 weeks. Then he can see how much effort the students spent in the last 2 weeks.

(1)The instructor can use the combination of both the filtering mechanisms. For example, if the instructor wants to know what each particular student did across various periods he can do so by applying more than 1 filter.



Thu Jul 31 01:03:21 CDT 2003
MoveSymbol , Symbol 'Collection
ChangeContent , Symbol 'Should
MoveSymbol
ResizeSymbol , Symbol 'Should
MoveSymbol , Symbol 'Should wr
ResizeSymbol
ResizeSymbol , Symbol 'Should
ResizeSymbol , Symbol 'We shou
Resize (with Move)
ChangeBorderColor
MoveSymbol
ResizeSymbol
ChangeBorderWidth
ChangeBorderColor
MoveSymbol
ChangeContent
AddSymbol
ResizeSymbol
ResizeSymbol
MoveSymbol
ChangeContent
MoveSymbol
ResizeSymbol

Figure 9a: The unfiltered history list



History Sessions

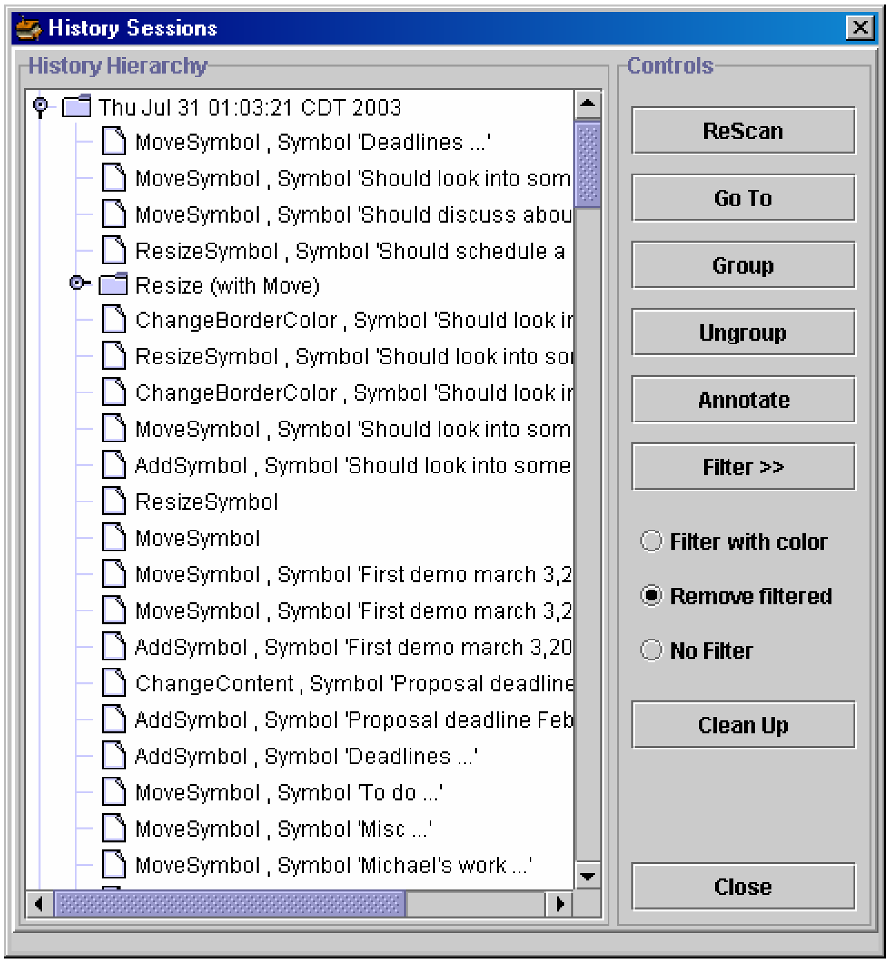
History Hierarchy

-----Current Location-----
Thu Jul 31 01:03:21 CDT 2003
MoveSymbol , Symbol 'Deadlines ...'
ChangeContent , Symbol 'Should discuss ab
MoveSymbol , Symbol 'Should look into som
ResizeSymbol , Symbol 'Should discuss abo
MoveSymbol , Symbol 'Should discuss abou
ResizeSymbol , Symbol 'Should look into so
ResizeSymbol , Symbol 'Should schedule a
ResizeSymbol , Symbol 'We should collect e
Resize (with Move)
ChangeBorderColor , Symbol 'Should look in
MoveSymbol , Symbol 'Should look into som
ResizeSymbol , Symbol 'Should look into so
ChangeBorderWidth , Symbol 'Should look in
ChangeBorderColor , Symbol 'Should look in
MoveSymbol , Symbol 'Should look into som
ChangeContent , Symbol 'Should look into s
AddSymbol , Symbol 'Should look into some
ResizeSymbol
ResizeSymbol
MoveSymbol

Controls

[● Filter with color] ○ Remove filtered ○ No Filter

Figure 9b: History list with events matching the filter highlighted


History Sessions

History Hierarchy

Thu Jul 31 01:03:21 CDT 2003
MoveSymbol , Symbol 'Deadlines ...'
MoveSymbol , Symbol 'Should look into som
MoveSymbol , Symbol 'Should discuss abou
ResizeSymbol , Symbol 'Should schedule a
Resize (with Move)
ChangeBorderColor , Symbol 'Should look ir
ResizeSymbol , Symbol 'Should look into so
ChangeBorderColor , Symbol 'Should look ir
MoveSymbol , Symbol 'Should look into som
AddSymbol , Symbol 'Should look into some
ResizeSymbol
MoveSymbol
MoveSymbol , Symbol 'First demo march 3,2
MoveSymbol , Symbol 'First demo march 3,2
AddSymbol , Symbol 'First demo march 3,20
ChangeContent , Symbol 'Proposal deadline
AddSymbol , Symbol 'Proposal deadline Feb
AddSymbol , Symbol 'Deadlines ...'
MoveSymbol , Symbol 'To do ...'
MoveSymbol , Symbol 'Misc ...'
MoveSymbol , Symbol 'Michael's work ...'

Controls

Figure 9c: History list, which shows only events matching the filter

7. HEURISTIC EVALUATION

7.1 THE PROCESS

The interface was evaluated by three computer science graduate students who have basic theoretical and practical expertise in the field of User interface Design and Usability. Each evaluator was given a set of nine actions to perform using the new history features and interface (see Appendix 1). After performing each of the activities they were asked to answer "Where do you think this interface failed or which part of this interface you liked?" and "Any other comments." The nine actions were chosen to cover common activities across grouping and ungrouping, annotating, and filtering the history. The complete comments and suggestions they provided are found in Appendix 2. The following discussion explores the lessons learned from this evaluation in terms of the general history interface, grouping of events, annotating events, and filtering the event list. Each of the evaluators took approximately 45 minutes to perform the evaluation.

7.2 GENERAL HISTORY

Regarding the history interface in general there were problems with the representation of elements in the history list. Two of the evaluators commented on the difficulty of locating events and that an efficient search feature should be implemented to search for specific events, since all the events look alike. The interface developed requires the users to manually scan the entire list. Another suggestion was to color code the events for easier and quicker identification.

7.3 GROUPING OF EVENTS

The evaluators felt it was easier to group the events but suggested that they would prefer to right click on the group and get all the options rather than having to click on another button. Another interesting suggestion was to be able to group non-consecutive events so that all events that refer to a particular topic are grouped even though they are not consecutive in the history list.

7.4 ANNOTATING HISTORY EVENTS

One of the evaluators commented that there is no visual indication to distinguish events that are annotated from those that are not. It was also suggested that it would be very useful if annotations associated with each event are displayed as tool-tip text when the mouse rolls over that particular event.

When asked to comment about the interface that add attributes to an event, all the evaluators felt the use of colon as a delimiter to enter name-value attribute pairs was very confusing and suggested other methods to accept the attributes from the user. One of the suggestions that refers to a possible alternative to this interface advocates the use of 2 different textboxes, each one to enter name and value separately. It was also suggested that the button to add the attribute to the event be initially disabled till the user actually enters some appropriate values in the text box. This lets the system validate the user input before the option to add the attribute is provided. One of the evaluators felt that the add attributes option should also be present as a button. In the current implementation, it is present only as an option in the pop-up menu when the specific event is selected. Another evaluator suggested adding an escape character to give the user the ability to add a colon part of the name or value in the attribute pair. It was also suggested that the user should be given an option to modify an already existing attribute.

7.5 FILTERING IN HISTORY

One of the evaluators suggested that the date value field that is used while specifying the date filter should be accompanied by a visual indication of the format in which the values are to be entered. An interesting suggestion refers to the incremental creation of a "cumulative filter". This cumulative filter would apply a new filter on the events that were generated by previous filter. It was also recommended that the interface should provide a summary of the results of the various actions that were performed pertaining to the filtering the history events. An evaluator observed that the name of the user whose events are being filtered is case-sensitive. One of the users opined that opening the filter dialog when a filter is already applied should show the already existing filters applied so that they can be modified or new filters added.

7.6 DISCUSSION

While all three evaluators were able to complete the nine actions without intervention, there are a number of valuable suggestions that should be part of the VKB history interface to make it more efficient and effective. The suggestion to be able to group non-contiguous but related edits brings up the issue that user activities are intermixed and a tree view for browsing a history may not be the best visualization. Thus, the evaluation results provide both low-level fixes to the current interface design and high-level issues for the history interface and representation as a whole.

8. CONCLUSION

History mechanisms in hypertext systems include navigation history mechanisms that record navigation events and authoring history mechanisms that record authoring events. History information can be useful to both humans and the system. Applications of navigation history include providing information about reader preferences, backtracking and rendering suggestions based on models of reader interest. Applications of authoring history include undo/redo operations, providing context for better interpretation of the context and macro generation.

This thesis investigates methods for coping with large edit histories. The grouping of events mechanism implemented adds higher-level task information to the various events in the history. Looking at the large list of history events with associated activity grouped helps the user in comprehending the evolution of the workspace better. Annotations and attributes added by the user events and groups of events enhance the interpretation of the history and its communicative utility for others. Filtering in history can be a useful tool for information seekers trying to rebuild the thought processes of the past. The user who can sometimes be overwhelmed by the amount of history information presented finds it useful to specify certain criteria by which only the events and group of events that satisfy the criteria are identified.

A heuristic evaluation of the interface and mechanisms indicated that the features are usable but there is much room for improvement. The enhancements suggested in the heuristic evaluation can be implemented to make the interface more usable and effective as part of future work. My contribution in this thesis is the addition of mechanisms to help users cope with larger history records in VKB via the process of grouping, annotating and filtering history information. All of these operations assign structure and meaning to the existing history.

REFERENCES

CONKLIN, J. 1987. Hypertext: An introduction and survey. In *IEEE Computer, 20*, 9, 17-41.

HAAKE, A. AND HAAKE, J.M. 1993. Take CoVer: Exploiting version support in cooperative systems. *Proceedings of InterCHI'93 - Human Factors in Computer Systems*, Amsterdam, Netherlands, 406-413.

HAYASHI, K. AND SEKUIMA, A. 1993. Mediating interface between hypertext and structured documents. [Electronic Publishing] *Volume 6*, 4, 423-434, December 1993.

KASHIHARA, A., HASEGAWA, S. AND TOYODA, J. 2000. An interactive history as reflection support in hyperspace. *Proceedings of World Conference on Educational Multimedia and Hypermedia* (ED-MEDIA 2000), 467-472, Montreal, Quebec, Canada.

KLEMMER, SCOTT R., THOMSEN, MICHAEL, PHELPS-GOODMAN, ETHAN AND LANDAY, JAMES A. 2002. Where do web sites come from? Capturing and interacting with design history. *Proceedings of CHI 2002, ACM Conference on Human Factors in Computing Systems*, *CHI Letters*, MN, 4(1), 1-8.

LEE, A. 1992. *Investigations into History Tools for User Support*. Ph.D. thesis, Computer Systems Research Institute, University of Toronto, Canada.

MARSHALL C.C., AND SHIPMAN, F.M. 1995. Spatial hypertext: Designing for change *Communications of the ACM, 38,* 8, 88-97.

NIELSEN, J. 1990. The art of navigating through hypertext**.** *Communications of the ACM, 33*, 3, 296-310.

PLAISANT, C., ROSE, A., RUBLOFF, G., SALTER, R. AND SHNEIDERMAN, B. 1999. The design of history mechanisms and their use in collaborative educational simulations. *Proceedings of the Computer Support for Collaborative Learning*, CSCL' 99, Palo Alto, CA, 348-359.

REEVES, B. 1993. *Supporting Collaborative Design by Embedding Communication and History in Design Artifacts*. Ph.D. Dissertation, Department of Computer Science, University of Colorado at Boulder.

SHIPMAN, F. AND HSIEH, H. 2000. Navigable History: A Reader's View of Writer's Time. *The New Review of Hypermedia and Multimedia, 6*, 147-157.

SHIPMAN, F., HSIEH, H., MALOOR, P. AND MOORE, J.M. 2001. The visual knowledge builder: A second generation spatial hypertext. In *Proceedings of Hypertext 2001*, Aarhus, Denmark, Aug 14-18, 2001, ACM Press: New York.

SHIPMAN, F., HSIEH, H. AND AIRHART, R. 2001. Analytic workspaces: Supporting the emergence of interpretation in the Visual Knowledge Builder. *Proceedings of Interact 2001*, Tokyo, Japan, 132--139.

SMITH, J.B., WEISS, S.F. AND FERGUSON, G.J. 1987. A hypertext writing environment and its cognitive basis. *Proceedings of the ACM conference on Hypertext*, Chapel Hill, North Carolina, USA. 195 – 214. ACM Press, New York.

TAUSCHER, L.M. 1996. *Evaluating History Mechanisms: An Empirical Study of Reuse Patterns in WWW Navigation*. M.S. thesis , University of Calgary, Canada.

TAUSCHER, L. AND GREENBERG, S. 1997. How people revisit web pages: Empirical findings and implications for the design of history systems. *International Journal of Human-Computer Studies, 47*, 1, 97-137.

THURING, M., HANNEMANN J. AND HAAKE, J. 1995. Hypermedia and cognition: Designing for comprehension**.** *Communications of the ACM, 38,* 8, 57-66. ACM Press, New York.

TRIGG, R. H., SUCHMAN, L. A. AND HALASZ, F. G. 1986. Supporting Collaboration in NoteCards. *Proceedings of CSCW'86*, 153-162. ACM Press, New York.

WEXELBLAT, A. 1998. History-rich tools for social navigation. In *Proceedings of the CHI 98 Conference*, Los Angeles, California.

APPENDIX 1. HEURISTIC EVALUATION

Please fill in your comments in the appropriate space provided along with each question.

(1) Group the first consecutive "Move and resize events" in the list.

They are :

(a) MoveSymbol, Symbol 'Should look into some related … …'

(b) ResizeSymbol, Symbol 'Should discuss about writing t … …'

Where do you think this interface failed or which part of this interface you liked?

Any other comments:

---

(2) Ungroup them.

Where do you think this interface failed or which part of this interface you liked?

Any other comments:

---

(3) Annotate the event:

MoveSymbol, Symbol 'Deadlines… …'

Where do you think this interface failed or which part of this interface you liked?

Any other comments:

---

(4) Add attributes to the event :

ChangeContent, Symbol 'Should discuss about writing t … …'

Where do you think this interface failed or which part of this interface you liked?

Any other comments:

---

(5) Use pop-up menu and add attributes and also annotation to the same event:
ResizeSymbol, Symbol 'Should schedule a meeting with … …'

Where do you think this interface failed or which part of this interface you liked?

Any other comments:

---

(6) Use filtering mechanism to know how many events are performed by Michael. [name filter]

Where do you think this interface failed or which part of this interface you liked?

Any other comments:

---

(7) How many events occurred in total after Sep 1, 2003 [date filter]

Where do you think this interface failed or which part of this interface you liked?

Any other comments:

---

(8) How many events were performed by Raghu after Sep 1,2003 [both the filters.]

Where do you think this interface failed or which part of this interface you liked?

Any other comments:

---

(9) How many events were performed by User whose name starts with Mic [name filter - different interface

----

Where do you think this interface failed or which part of this interface you liked?

Any other comments:

APPENDIX 2. HEURISTIC EVALUATION RESULTS

(1) Group the first consecutive "Move and resize events" in the list.

They are :

(c) MoveSymbol, Symbol 'Should look into some related … …'

(d) ResizeSymbol, Symbol 'Should discuss about writing t … …'

Evaluator 1: It was easy to group the items.

Evaluator 2: The interface requires the users to manually scan the entire list.
All events look alike, other than specific text that describes them.
Its like dealing with dollar bills, the only way to tell the difference is by reading the text.
Comments :
It may be better to allow the users to allow to search the event list for the events with specific properties such as annotations, names, types, text altered etc.
The events may be color-coded (background, foreground), font faces may be changed etc. for quicker easier identification.

Evaluator 3: Easy to group events by using shift-select.
Comments:
Would like to group non-consecutive events based on topic.

(2) Ungroup them.

Evaluator 1: I would prefer to right click on a group and get all the options I can do with it rather than having to click on the ungroup button.

Evaluator 2: For both grouping and ungrouping it might be easier if the option is located on a menu that appears upon a right click (or shift + right click).

Evaluator 3: Once I selected the group, items were ungrouped into leaves. I tried to ungroup a leaf (not belonging to the group and the leaf disappeared?!)

(3) Annotate the event:

MoveSymbol, Symbol 'Deadlines… …'

Evaluator 1: It was hard to find that item. Probably adding a searching option would be useful.

Evaluator 2: The menu that appears on right click is useful. There is no visual indication to distinguish events that are annotated from those that are not. Along with visual indication it may be interesting to explore the possibility of displaying annotations as tool-tips.

Evaluator 3: Very easy to annotate either by using "controls-box" or by right clicking.
Comments:
Worked really well.

(4) Add attributes to the event:

ChangeContent, Symbol 'Should discuss about writing t … …'

Evaluator 1: It was a little confusing to figure out what to do with ";". Probably separating attribute and value in 2 different text boxes would be less confusing. Also the <add attribute> button should be initially disabled and not enabled until the user has entered the pair <name: value>

Evaluator 2: Erroneous indicator. <name: value> for me, means type in this by replacing "name" and "values" in the actual attribute information, while retaining the greater than and less than signs. This action also added these signs as part of my attribute name and value. The correct indicator for your desired action is <name>: <action>.

- How would I enter an attribute that contains a colon
  The interface assumes that the first colon separates name and value.
- How do I change or modify an attribute?
- It might be useful for users to type name and value in 2 separate boxes, that way the colon character will still be available for use.
- <Return key> should act like pressing <add attribute> button.
- Title of Add Attribute button needs a space "Add Attribute".

Evaluator 3: Would be nice to have "Add attribute" button in the "controls" box but it worked very well.

Comments :

Easy to use.

(5) Use pop-up menu and add attributes and also annotation to the same event: ResizeSymbol, Symbol 'Should schedule a meeting with … …'

Evaluator 1: It was easier to work with.

Evaluator 2: The "attribute" menu item allows the user to add both attribute and also annotations. But the annotation interface does not. These interfaces should be consistent.

The attribute interface does not have a cancel button. The menu is inconsistent: "Annotate is an action". The attributes menu item should should probably read "Add Attributes" or "Annotation and Attributes".

Evaluator 3: Really simple to use. Just right click to annotate and add attributes.

Comments :

I even added a colon within my attribute. Seems like the default is to take the fist colon as the delimiter. Is there an escape character for adding colon within an attribute.

(6) Use filtering mechanism to know how many events are performed by Michael. [name filter]

Evaluator 1: It does not give me the number of items that match the filter condition, so I have to go through counting them. It would also help to indicate filter results as 0/0.

Comments:

The filtering option should have one of the presentation options as default. For example after applying the filter I forgot to select the "filter with color" option.

Evaluator2:

- Filtering interface should read "And" instead of "Add".

- The criteria, condition, value drop boxes should always ensure the consistency of items displayed.
- The date value field should contain visual queues(a text string like " for eg.1985").
- For a single filter there is no need to perform the filter. They could be combined for easier use.
- Opening the filter when the filter is already opened should show the existing filter so that it can be modified or additional filters added. Currently displays nothing in the "Selection Criteria".

Evaluator3: Really easy to use the filter. Interface even told me to add my filter to the list. Worked great!

(7) How many events occurred in total after Sep 1, 2003 [date filter]

Evaluator 1: Same as #6. Plus it would help to indicate the format of the year.

Evaluator 2: The interface should provide a summary of results of actions performed say print something like "Filtering results for "<condition>" provided xyz results.

Evaluator 3: I assume it brought the right results. Returned the values.

Comments:

Not sure whether to count the items also in the group. Would be nice if filter returned count for me.

(8) How many events were performed by Raghu after Sep 1,2003 [both the filters.]

Evaluator 1: Same as #6 and #7. What about a cumulative filtering option i.e. apply a filter and allow the user to explore the results, then apply the filter again to the previous results. Right now the user applies all the filters at once.

Evaluator 2: The filter dialog should display the relationships between all filters applied instead of simply listing them.

Filter 1 Filter 2 is less informative than Filter 1 AND Filter 2.

Evaluator 3: Did not know how to use 2 filters. Plus my previous filters disappeared beforehand.

(9) How many events were performed by User whose name starts with Mic [name filter - different interface

Evaluator 1: Same as #6 and #7.

Evaluator 2: My answer is based on exact search string (He found there are 0 results that match the criteria.) "mic" probably yields different results ☺ Check for case sensitivity.

Evaluator 3: Very easy to use "begins with"

Comments:

Not sure if results were right though. They looked right! Simple interface that's self-explanatory. Could add extra "status" info like count of number of results etc.

VITA

Name:	Raghu Chaitanya Akkapeddi
Permanent Address:	C/O Branch Manager. State Bank of Hyderabad Dubba Branch Nizamabad – 503 001. Andhra Pradesh. INDIA.
Education:	M.S. Computer Science, Texas A&M University, 2003 B.E. Computer Science and Engineering, Chaitanya Bharathi Institute of Technology, Hyderabad, Osmania University, Andhra Pradesh, India, 2001.