# COMPARISON OF EDIT HISTORY CLUSTERING TECHNIQUES FOR SPATIAL HYPERTEXT

A Thesis

by

BIKASH MANDAL

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 2005

Major Subject: Computer Science

# COMPARISON OF EDIT HISTORY CLUSTERING TECHNIQUES FOR SPATIAL HYPERTEXT

A Thesis

by

BIKASH MANDAL

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

Approved by:

| | |
|---|---|
| Chair of Committee, | Frank Shipman |
| Committee Members, | Richard Furuta |
| | Michael G Messina |
| Head of Department, | Valerie E. Taylor |

December 2005

Major Subject: Computer Science

# ABSTRACT

Comparison of Edit History Clustering Techniques for
Spatial Hypertext. (December 2005)
Bikash Mandal, B.S., Bangladesh University of Engineering & Technology
Chair of Advisory Committee: Dr. Frank Shipman

History mechanisms available in hypertext systems allow access to past user interactions with the system. This helps users evaluate past work and learn from past activity. It also allows systems identify usage patterns and potentially predict behaviors with the system. Thus, recording history is useful to both the system and the user.

Various tools and techniques have been developed to group and annotate history in Visual Knowledge Builder (VKB). But the problem with these tools is that the operations are performed manually. For a large VKB history growing over a long period of time, performing grouping operations using such tools is difficult and time consuming. This thesis examines various methods to analyze VKB history in order to automatically group/cluster all the user events in this history.

In this thesis, three different approaches are compared. The first approach is a pattern matching approach identifying repeated patterns of edit events in the history. The second approach is a rule-based approach that uses simple rules, such as “group all consecutive events on a single object”. The third approach uses hierarchical agglomerative clustering (HAC) where edits are grouped based on a function of *edit time* and *edit location*.

The contributions of this thesis work are: (a) developing tools to automatically cluster large VKB history using these approaches, (b) analyzing performance of each approach

in order to determine their relative strengths and weaknesses, and (c) answering the question, “how well do the automatic clustering approaches perform” - by comparing the results obtained from this automatic tool with that obtained from the manual grouping performed by actual users on a same set of VKB history.

Results obtained from this thesis work show that the rule-based approach performs the best in that it best matches human-defined groups and generates the fewest number of groups. The hierarchic agglomerative clustering approach is in between the other two approaches with regards to identifying human-defined groups. The pattern-matching approach generates many potential groups but only a few matches with those generated by actual VKB users.

## ACKNOWLEDGEMENTS

It is my great pleasure to offer my heart felt thanks and gratitude to a number of people without whose help and support this thesis work would never have been possible.

I am deeply grateful to Dr. Frank Shipman, my thesis supervisor, who led me in the right direction with thoughtful advice. From the very beginning he has always given me constant guidance and inspiration to make this thesis work successful. I would like to express my gratitude to Dr. Richard Furuta, Dr. Michael G Messina and Dr. Carroll Messer. Dr. Messer was my external advisor at the beginning of the thesis work. Their thoughtful comments and suggestions have made this thesis more consistent.

I had the opportunity to work with a number of very talented and knowledgeable people at the Center for the Study of Digital Libraries Laboratory (CSDL) at Texas A&M University. I would like to specially thank Dr. Haowei Hsieh. He has been tremendously helpful for a number of implementation issues during the development of this thesis work. I would also like to thank my fellow students and researchers in the CSDL Laboratory. They provided valuable suggestions and feedback during the discussions in CSDL group meetings.

Finally I thank all others who have directly and indirectly helped me in this thesis work in the department and outside the department.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# 1. INTRODUCTION

Structuring and interacting with data in freeform systems is an active area of research [Chiu 1998], [Edmonds 1988]. In freeform systems, flexibility in the organization of data is important. Data should not be structured prematurely as, during the course of the activity, perceived structures may emerge in unexpected ways and they may be subjected to multiple interpretations. When the domain of application is known, a more robust system can be achieved by designing it to dynamically interpret an anticipated set of frequently used structures appropriate for that domain. Unfortunately, when freeform manipulation of content is enabled, the resulting interpretation can become ambiguous over time. One approach to addressing the communication issues that result is to store the history of the information space so that users can access earlier versions when trying to disambiguate expression.

With the continued decrease in the cost of persistent storage, more applications are storing user history for use beyond the individual session. There are a variety of uses of such history. Lee [Lee 1992] identifies seven basic uses of information in a user's history: reuse, inter-referential input/output, error recovery, navigation, reminding, user modeling, and user interface adaptation. Greenberg [Greenberg 1993a] distinguishes three kinds of reuse facilities: adaptive systems, programming by examples (PBE) and history mechanisms. *Adaptive systems* use dynamic models of previous inputs to predict subsequent ones, which are then made available to the user. *Programming by example* is concerned with the reuse and generalization of long input sequences. *History mechanisms* allow users to manipulate a temporally ordered list of their interactions.

---

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

User history over long periods of time can become huge and difficult to manage. One example of this, and the context of this thesis, is the Visual Knowledge Builder (VKB) [Shipman 2001] developed in the CSDL Laboratory of Texas A&M University. VKB is a spatial hypertext system and provides a visual workspace for collecting, organizing, and sharing information. It is based on prior work on support for information analysis in VIKI [Marshall 1994]. It uses two-dimensional *collections* that can contain other collections and visual *symbols* that represent information *objects*. Users modify visual attributes of *collections* and *symbols* to indicate their interpretation of the information. These features facilitate the development of task-specific visual languages and support collaborative knowledge-building tasks.

VKB has been under development since 1997 and in use since 1999. Some VKB workspaces have been in use for more than five years and include two to three thousand events. Prior work on VKB explored an interface for attaching user interpretations of history via grouping and annotating history events [Akkapeddi 2003]. Groups may include other groups to form a hierarchal view of edit history. Groups of edits can be assigned meaningful labels by the user. Annotations attached to user events or groups of events can take the form of a plain text statement or metadata, one or more attribute/value pairs. The grouping and annotations allow filtering of the history to locate past states of the VKB document. Unfortunately, the grouping and annotation of large edit histories requires significant human effort and is rarely used.

This thesis work explores the potential to automatically analyze the edit history to generate initial groupings of edit events to reduce required human effort. These groupings and patterns may be valuable for: (a) accessing desired segments of long edit histories (b) identifying portions of edit history with special or similar attributes, and (c) predicting future interaction during VKB authoring. This thesis implements and analyzes following three approaches to cluster VKB edit history:

*Pattern matching approach:* This approach attempts to recognize repeated event patterns via detailed analysis of the edit history. History event sequences of a specified length range are identified using a simple pattern matching algorithm [Capelle 2002]. This approach helps develop tools similar to those used in programming by demonstration (PBD).

*Rule based approach:* In this approach spatial relationship among objects determines the grouping of history events. The edit events/actions are grouped based on the object(s) being edited and the container of objects. The objective of this approach is to generate simple and logical groups based on *spatial* and *containing* relationships of VKB objects and containers.

*Hierarchical agglomerative clustering (HAC) approach:* In this approach, *time* and *space* in VKB workspace domain are used as two different dimensions during clustering. The time when sequential events occur and position of the objects/collections in the VKB workspace that are manipulated by those events are used to determine the "inter-distance" between the two events. The inter-distance is calculated from a linear function of *time* and *space*. The objective of this approach is to find clusters of objects based on proximity in the VKB history space.

This thesis work is carried out by implementing following 3 steps:

(a) Automatic clustering tools and techniques are developed to do the clustering of a large VKB history file using three approaches mentioned above;

(b) Performance of each clustering technique is analyzed to determine their relative strengths and weaknesses and finally;

(c) Answering the question, "how well do these automatic clustering tools perform" by comparing the results obtained from these automatic tools with that obtained from the manual grouping by actual VKB users on a same history.

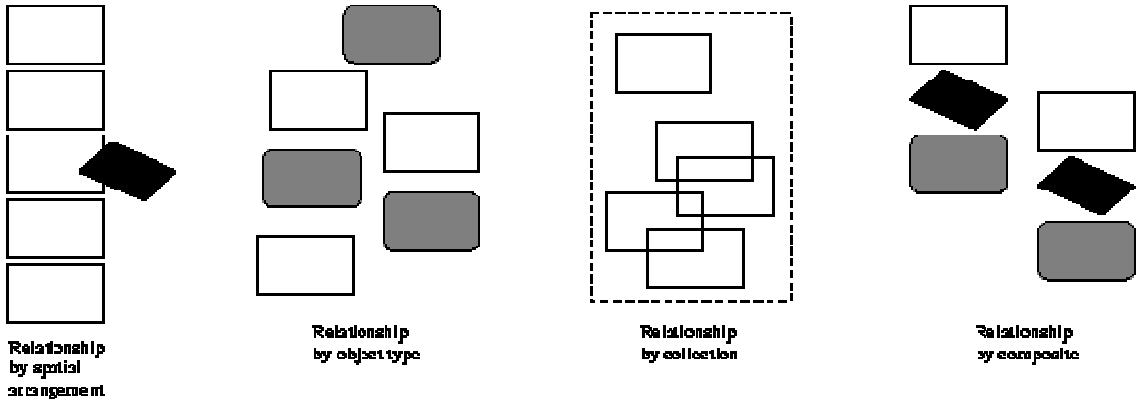
This thesis will provide an understanding of the automatic grouping techniques with advantages and disadvantages of the three different approaches implemented. This result informs the design of history clustering for other applications that store edit history. These results will also facilitate development of further enhancements to VKB's functionality and user interface.

The next section provides an overview of spatial hypertext. Section 3 enumerates many of the potential roles of user history in hypertext systems. Section 4 describes the history-related functionality provided in the current version of VKB and the need to automatically cluster VKB edit history. Section 5 presents an overview of clustering techniques and Section 6 describes three clustering approaches applied to VKB histories. Section 7 compares the performance of the three approaches. Section 8 discusses these results and Section 9 presents conclusions from this thesis project as a whole.

# 2. SPATIAL HYPERTEXT

Spatial hypertexts evolved from traditional navigational and node-link based hypertext systems like Xerox NoteCards [Halasz 1987] and the Virtual Notebook System [Shipman 1989]. Such systems provided views of the node and link graph structure of the hypertext to provide users with greater context about how one page of content fit within the overall hypertext. The success of such views led to map-based hypertexts that used graphical views of network as the primary interface, including Aquanet [Marshall 1991] and gIBIS [Conklin 1988]. When such tools were used for authoring, users often avoided creating explicit links between information objects and instead used spatial position and visual similarity to express relations.

Spatial hypermedia systems like VIKI [Marshall 1994] were designed to provide visual means to develop task-based interpretations of information using proximity, alignment, repeated patterns, and visual attributes of information chunks. This expression is used to indicate information object similarity, to cluster information objects into groups, and to identify relationships between information objects.




(a) (b) (c) (d)

Figure 2.1: Types of relationships found in spatial hypertext

Figure 2.1 provides examples of four forms of spatial expression found in spatial hypertext systems.

*Relationship by spatial arrangement*: This relationship is expressed by proximity between objects [Figure 2.1(a)]. It can be horizontal lists, vertical lists, matrices, stacks etc., each of which requires a certain alignment and visual uniformity of objects to express ordered groupings.

- *Relationship by object type*: As shown in Figure 2.1(b), it is expressed by visual similarities between objects. Object type relationships can express category membership that cross-cuts spatial positioning. Visual similarities can be expressed with color, font, extent, shape, border-width etc.

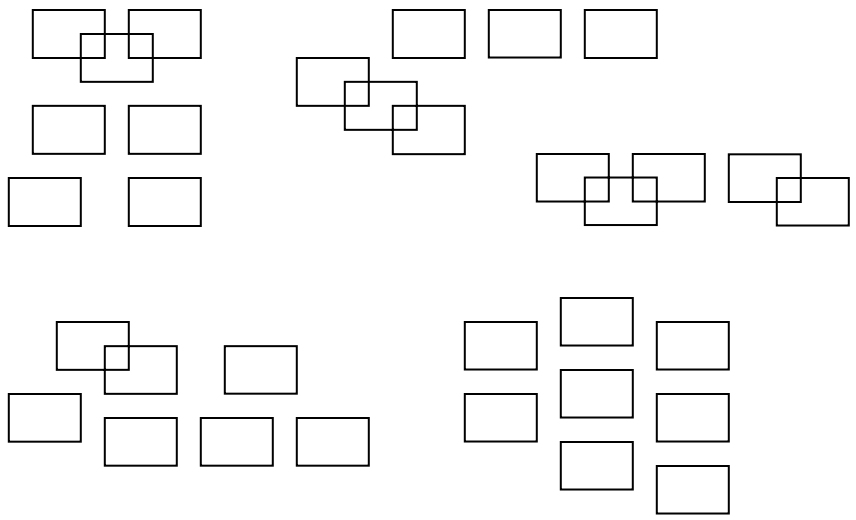


Figure 2.2: Relationship by activity-related areas

- *Relationship by collection*: This relationship is categorizes a set of objects according to some criteria [Figure 2.1(c)]. A spatial hypertext collection is a separate space containing the set of objects that are related by their membership in that collection. Subspaces can express hierarchical ordering of collections.
- *Relationship by composite*: Composites provide users with the ability to express relationships between different types of objects [Figure 2.1(d)]. They can reveal themselves as recurring patterns of objects, where the immediate constituents of the composites can be of arbitrary type.

In addition to the forms of expression shown in Figure 2.1, people group objects into activity related areas, as shown in Figure 2.2.

- *Relationship by activity-related areas*: As shown in Figure 2.2, a space is said to be organised into activity-related areas when objects are clustered (by spatial proximity) in such a way that objects in the same cluster relate to the same task. Only the clear spatial separation of objects into clusters is significant, not structure within the clusters.

## 2.1 CHARACTERISTICS OF SPATIAL HYPERMEDIA SYSTEMS

The following is a description of properties of the spatial hypermedia systems:

(1) Structure creation: In spatial hypermedia, the focus is on active creation of structure rather than reading and traversing structure [Marshall 1994].

(2) Emerging structure: Spatial hypermedia allows structure creation even when users only have a fuzzy idea about what kind of structure to create. It allows the user to create collections before naming and typing them. Thus it supports explorative development of

informal emerging structure. In [Marshall 1997], it is mentioned that *"people's organisational strategies change over the duration of a task as they become more familiar with the material to be organised. This stresses the need for emerging structure in tasks involving structuring."*

(3) Flexibility: Spatial hypermedia allows users to express a variety of interpretations of material by arranging material in multiple ways. The flexibility of spatial arrangment makes it easy to express relationships such as similar-to or different-from [Marshall 1995]. Along with spatial properties, the ability to change color, border, shape, font, etc., enables the expression of independent categorizations and information characteristics. For instance, one could express more relevance of an object by thicker borders, and less relevance by thinner borders.

(4) Common understanding: Spatial hypermedia allows people in a group setting to discuss appropriate structures and, by moving objects around in a space, suggesting new structures. In this way communication surrounding a set of information can be grounded in the visual representation, providing for a common understanding of structure to be developed, over time. Though spatial structure is one's personal reflections, that understanding and interpretation of a collection of information can be communicated to others via visual expression [Marshall 1997].

(5) Handling large structures: Spatial hypermedia allows people to structure large amounts of data in a comprehensive way. Some methods to show overview of structure fail when information sets grow large. For example, graph structures and networks indicating relationships and interdependencies get confusing when there are many links between nodes [Feiner 1988].

(6) Spatial parsing: Some spatial hypertext systems include spatial parsers that attempt to recognize the perceptual structures being created in the visual workspace. These recognized structures are used by the system to facilitate the selection and manipulation of the structures and to aid in formalizing the implicit representation of relationships into

an explicit structure. The spatial parsers also provide feedback to the users about what the system has understood and provide some indication as to the comprehensibility of the information space for others.

## 2.2 OTHER SPATIAL HYPERTEXT SYSTEMS

The use of spatial expression had been part of a number of earlier systems, such as Boxer's use of space in a programming environment [diSessa 1986]. Among hypertext systems, Storyspace used the concept of a writing space [Joyce 1991] for arranging the contents of a hypertext while authoring the links that are the structure of the resulting hypertext. There are now a large number of spatial hypertext systems or systems that have been motivated by spatial hypertext systems. Web Squirrel used the ease of spatial expression to make it easy to keep track of large numbers of URLs [Simpson 2001]. Hot Sauce is a graphical browser presenting labelled nodes in multiple colors in a three dimensional space [Guha 2005]. Nodes can link to external resources, like web pages, gopher files, ftp files, desktop files, email and database structure. A node can also hide/reveal other nodes to which it links to; thus capable of expressing hierarchical relationships. Hot Sauce presents content in a view where objects are floating freely in space.

## 3. HYPERTEXT HISTORY

A very common phenomenon in our everyday life is “repetition”. It has impact over computing and non-computing activities. In non-computing activities, people replay their favorite music; favor a subset of recipes and tools, and follow similar procedures to accomplish routine office tasks [Greenberg 1993b]. Hanson, Kraut, and Farber [Hanson 1984], and Greenberg and Witten [Greenberg 1988] report that, in computing environment, *“a few commands issued during a user session comprise a large percentage of the total commands used”*. According to Lee [Lee 1992], recurrent activities arise for several reasons. Some tasks are routine and frequent or inherently repetitive. Given that many computer-based tasks involve repetition, there is an opportunity to collect a record of user’s interactions—called a *user history*—and make it available for future use.

Lee also identifies seven basic uses of the information in a user’s history: reuse, inter-referential input/output, error recovery, navigation, reminding, user modeling, and user interface adaptation. Greenberg [Greenberg 1993a] distinguishes between three kinds of reuse facilities: adaptive systems, programming by example, and history mechanisms. *Adaptive systems* use dynamic models of previous inputs to predict subsequent ones, which are then made available to the user. *Programming by example* is concerned with the reuse and generalization of long input sequences. *History mechanisms* allow users to manipulate a temporally ordered list of their interactions.

There are two major types of history mechanisms in hypertext systems. The first type of history mechanism records navigation events, such as the traversal of links from page to page, and is appropriately called *navigation history*. The second type of history mechanism records the edit events that make up the authoring process, and is called *authoring history*. In navigation history, the movement of a reader through a hypertext is

recorded. This recording can be used in a number of applications like backtracking, already visited clues, generating personalized list of nodes and paths etc. Navigation history and some of the features it makes possible are included in most Web browsers. In the authoring history, events that edit the hypertext are recorded. Authoring history, at least at the session level, is found in editors like MS Word. They allow revisiting prior versions of a document and undo/redo edits.

Histories, whether of navigation or authoring, can either be linear or branching. In a linear history, returning to a prior state (e.g. using the “back” button in the browser to go to a previously visited page) and then making an alternate choice clears the forward history from the point of change. Branching history provides more information as each navigational path or editing sequence is recorded as a branch with one being indicated as being that for the current state of the hypertext. This type of history mechanisms has been proposed by Cockburn and Jones [Cockburn 1996].

## 3.1 NAVIGATION HISTORY

The primary uses of navigation history found in Web browsers are backtracking, history lists, and already-visited cues. Additional applications of navigation history include paths and guided tours, visualizations of and reflections on navigation, and the recognition of access patterns.

Backtracking and history lists enable the user to go back to a previously visited page. Backtracking reduces the disorientation problem, also known as the “getting lost in hyperspace” problem [Conklin 1987], by allowing the user to go back over the course of link traversal. There are a number of alternatives to single step backtracking. One approach is a structure-oriented backtracking [Nielsen 1990], where systems provide a

direct backtrack jumps based on their knowledge about the structure of the hypertext, to locations where the user is likely to want to return.

History lists tend to take one of two forms. The first representation is a sequential text-based list of the titles of nodes visited. Nielsen [Nielsen 1990] implemented the later method in a HyperCard-based hypertext system. The Symbolics Lisp [Walker 1987] environment contains a window-based hypertext system called the *Document Examiner*. This was designed for its extensive online manual. It provides two types of historical lists, a *command* history and a history of *topics* examined. The second representation is a "cache" of graphical miniatures of the most recently visited nodes. The National Museum of Denmark museum used a visual "cache" approach in their information system [Nielsen 1990]. The screen in the information system includes a graphic of a museum artifact. It is displayed horizontally along the bottom of the screen to provide the user with a quick method for returning to the eight most recently visited nodes.

*Visual indicators such as checkmarks, asterisks, or the plus sign serve as "footprints" in overview diagrams and help users to avoid returning to nodes that have been recently visited* [Wexelblat 1998]. Bernstein's Hypergate system displays a small marker called a "breadcrumb" next to hyperlinks. This leads to material the reader has already seen. A checkmark next to nodes that have been visited within the overview map is placed in Nielsen's HyperCard-based hypertext system [Nielsen 1990].

A persistent form of navigation history is Vannevar Bush's path [Bush 1945]. In today's hypertext systems, *paths* are typically associated with the idea of a guided tour, where author determines an appropriate order of presentation for a given audience. It may even include annotations explaining the items on the path. Trigg [Trigg 1986] extended this concept to NoteCards and called *stops* along the tour *tabletops*.

Similar to paths, “Footprints” [Wexelblat 1998] uses information from web server logs to figure out where users have gone and then processes that information into displays. It also provides a navigable scrollable list of titles for pages visited by the user. Thuring et al. [Thuring 1995] refers to a presentation interface that shows the history of a hyperspace reading session. It shows by chronologically listing nodes visited during the session. It also shows reader’s current position within that hierarchy.

Paths do not have to be shared. Kashihara, et al. [Kashihara 2000], encourage users to reflect on their *“interactive history”*, providing users with a view of their exploration process. It allows users to view and annotate the exploration history in order to rethink the exploration process they have carried out so far.

Another area of use of navigation history is identifying *access patterns* and thus pre-fetching data that may be accessed by the user in the near future. Navigation history can also be used to generate models of *users’ interests* in order to make navigation suggestions to readers. The access metrics and access patterns available from server-side navigational history (server logs) can identify what information readers desire most and can help locate confusing links or common “wrong turns”.

## 3.2 AUTHORING HISTORY

Authoring or edit history can be very informative to both authors and readers. Authoring history is perhaps most valuable to authors as it allows them to undo and redo actions, to return to prior versions of their hypertexts, and to review edits made by collaborating authors. Edit history information helps readers interpret ambiguous materials by providing access to the context of an edit. Reviewing a sequence of edits provides insight into the thought processes and intentions of the author. Authoring history provides readers with a notion of the document’s constructive time [Akkapeddi 2003].

A variety of hypertext systems have included some form of edit history. CoVer [Haake 1993] represents various versions of the same object through multi-state objects (mobs). The “derivation history” generates an unconnected graph structure over all versions of documents. The application can explore historical development according to a number of attributes like creation date of versions, specific author and comprehensible view over the version.

Writing Environment is an environment that can be used to create both electronic and printed documents [Smith 1987]. It has an “online tracker” to capture users’ interactions. The captured information is represented as a sequence of symbols with attributes that constitute the history of the session. It also has a replay function that replays all the events that occurred during the session.

Hayashi and Sekuima [Hayashi 1993] in their “Nelumbo” system, implemented a mechanism that records users access histories in tree forms. Its WYSIWYG editor, displays not only the final version of the document but also the snapshot of various versions of the same document.

Additional forms of support based on authoring history can be provided by the system. Visual cues can be added to frequently modified portions of content [Edit wear]. Also, the system may use authoring history to identify common patterns of edits. These patterns can be the basis for macros, or where the system offering to complete menial tasks for the author. More ambitious generalizations of such patterns can be the basis for programming by demonstration (PBD) techniques. In such scenarios the system learns new behavior by analyzing the edits made by users working in the system over a period of time.

## 3.3 HISTORY APPLICATIONS

As mentioned in this section, there are a variety of applications of history data in hypertexts. Table 3.1 summarizes the above discussion by identifying applications of authoring and navigation history for readers, authors, and the system.

Table 3.1: Uses of authoring and navigation history by authors, readers, and the system [Akkapeddi 2003]

| | Hypertext Author | Hypertext Reader | System |
|---|---|---|---|
| Authoring History | Undo/Redo , Return to prior document states, Replaying edits of co-authors | Provides context for interpreting the hypertext content | Visualization of edit patterns, Macro generation, Programming by demonstration |
| Navigation History | Information about reader preferences and confusion | Backtracking, “Already visited” cues, Paths | Pre-fetching, Suggestions based on models of reader interest |

# 4. HISTORY IN THE VISUAL KNOWLEDGE BUILDER

The potential value of capturing edit history in spatial hypertext became apparent during use of VIKI [Marshall 1994], as users in collaborative spaces would leave notes to one another to explain the changes they had made. As such, the design of the Visual Knowledge Builder (VKB) included the ability to record the edits to the space and make this history persistent across sessions.

VKB provides four access mechanisms whereby users can return to prior states of or replay edits to the spatial hypertext [Shipman 2001]. The first is via play forward and backward buttons on the left side of the history toolbar (shown in Figure 4.1). Using these buttons, the user can step forward and backward one edit at a time or play through edits continuously in either direction. A control is provided for modifying the speed at which continuous playback occurs. The second access mechanism is a slider on the same toolbar that indicates the number of edits in the overall history and position of the state currently being displayed within that history. Users can drag the slider back and forth to get to prior or later periods of activity. As the user drags the slider, or as the slider changes position due to the displayed state changing based on one of the other history access mechanisms, the time indicator on the right side of the history toolbar changes to indicate the time of the edit currently indicated by the slider.

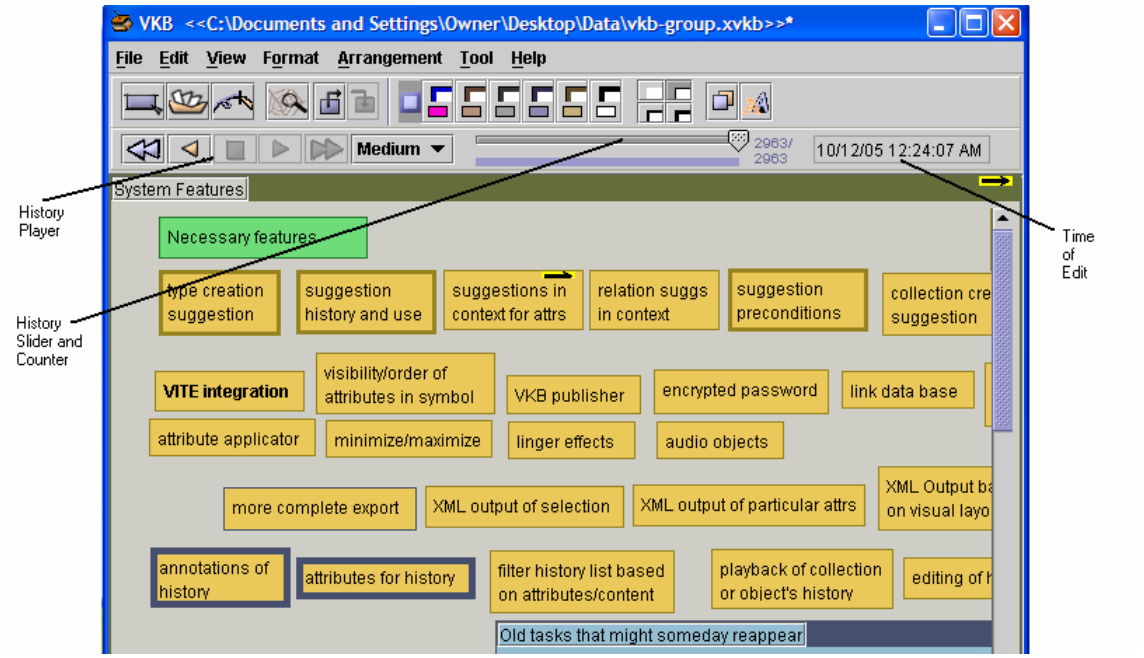



Figure 4.1: VKB history mechanisms

The two other history access mechanisms provided are selecting the start of a particular editing session in the History Sessions Dialogue (see Figure 4.2) and returning to a specific edit event on a particular object via a popup menu in the workspace. Together, the record of history and the variety of access mechanisms provide readers with a view of the writers', or constructive, time [Shipman 2001]. This supports learning and interpreting authors' work practices, recognizing patterns of activity in information space, and disambiguating specific actions and context.

Gaps between time stamps of the various events in the history provide a sense of the length and frequency for authoring and editing sessions. VKB uses gaps in the history to automatically recognize the editing sessions presented in the History Sessions Dialogue. This Dialogue also allows users to express interpretations related to the edit history by selectively grouping edit events together into meaningful clumps and annotating these groupings [Akkapeddi 2003].

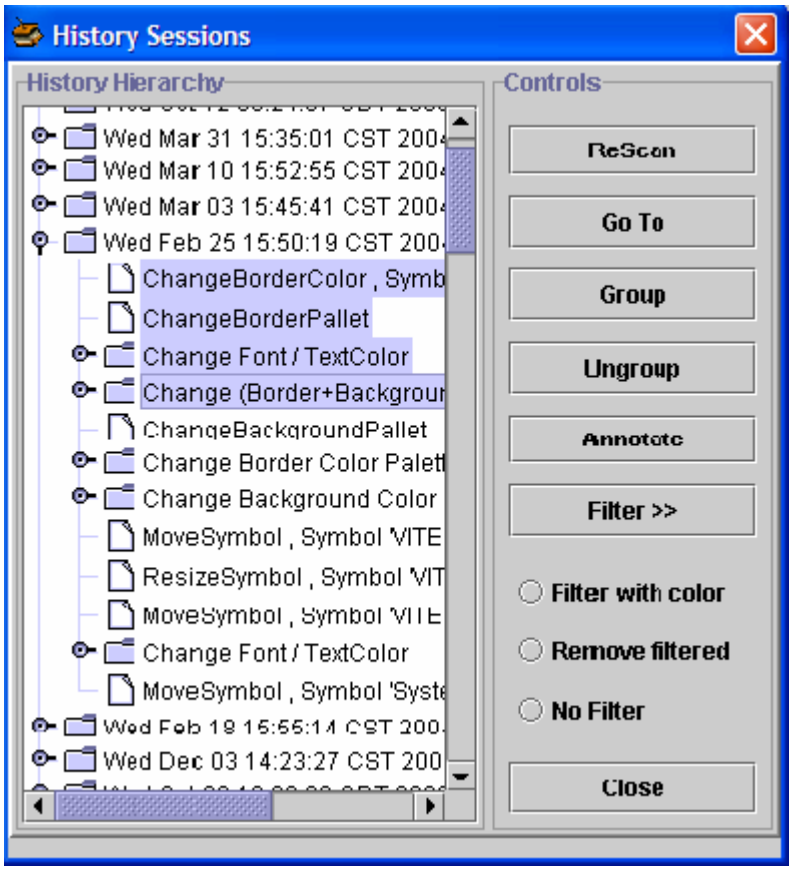



Figure 4.2: History sessions dialogue

Groups may include other groups to form a hierarchal view of edit history. Groups of edits can be assigned meaningful labels by the user. Annotations attached to user events or groups of events can take the form of a plain text statement or metadata, e.g., one or more attribute/value pairs. The grouping and annotations allow filtering of the history to locate past states of the VKB document. Unfortunately, the grouping and annotation of edit histories requires significant human effort and is rarely used.

## 4.1 THE PROBLEM OF LONG HISTORIES

VKB has been under development since 1997 and in use since 1999. Some VKB workspaces have been in use for more than five years and include two to three thousand events. Manipulating the edit history in these cases becomes difficult. Having meaningful groupings within the edit history can help in accessing desired segments of long edit histories, identifying portions of edit history with special or similar attributes, and predicting future interaction during VKB authoring sessions. For a large VKB history, the current interface and support for grouping edit events is too difficult and time consuming.

The problem is two-fold. The first issue is that the level of detail of user activity captured in edit event is too low level for humans to evaluate and care about. The second issue is that it is only once the history becomes large that users start to perceive value to grouping and annotating the user history. By then, the history is too long for the current author-centric mechanisms to be used.

The rest of this thesis describes the exploration of techniques for automatically clustering VKB edit history to solve this problem. The next section provides an overview of clustering methods. This is followed by a description of the three methods implemented and evaluated with VKB history data.

# 5. AUTOMATICALLY CLUSTERING EDIT HISTORY

The thesis has three major objectives. The first objective is developing tools to automatically cluster large VKB history. The second objective is analyzing the results of different clustering approaches to determine their relative strengths and weaknesses. The third objective is evaluating what roles the clustering approaches can fill in VKB by determining how well they match the manual grouping by VKB users.

## 5.1 CLUSTERING APPROACHES

Clustering is an unsupervised learning method. In a clustering process there is no specific target value to be predicted. The objective of the clustering process is to find common patterns or groups of similar examples. There are a number of models and algorithms available for clustering. They can be classified and compared according to following dimensions:

(a) Conceptual (model-based) vs. partitioning clustering

(b) Exclusive vs. overlapping clustering

(c) Hierarchical vs. flat clustering

(d) Incremental vs. batch clustering

*Conceptual clustering* groups elements based on a preexisting model while *partitioning clustering* groups elements without using a preexisting model (e.g. based on their similarity to one another). If each element is placed in at most one cluster then it is *exclusive* while an element can be simultaneously placed in multiple clusters when using an *overlapping* approach. Clustering algorithms that produce clusters of clusters are *hierarchical* while those that do not are considered *flat*. Finally, if a clustering approach requires all the elements to be clustered prior to execution then it is a *batch* approach

while those that can cluster new elements without revisiting previously clustered elements are *incremental*.

There are many different approaches for clustering data. Some frequently used approaches are: (a) Cluster/2, (b) K-Means partitioning, (c) probability-based partitioning, (d) hierarchical agglomerative clustering, and (e) hierarchical divisive clustering [Everitt 1981]. Due to the goal of generating a hierarchy groups in edit history, this thesis will focus on hierarchical approaches.

Hierarchical clustering techniques build clusters step by step. There are two main approaches in hierarchical clustering: the Hierarchical Divisive Clustering technique and the Hierarchical Agglomerative Clustering (HAC) technique [Everitt 1981].

Divisive methods start with the assumption that all objects are part of a single cluster. The algorithm splits this large cluster step by step until each object is a separate cluster. On the other hand, agglomerative methods start inversely. Initially each object forms a separate cluster. The clusters are combined step by step. In each step, the two clusters with the highest similarity or the smallest dissimilarity are merged. Iteration continues until all objects are in one single cluster. Thus, hierarchical agglomerative clustering methods are based on a similarity or dissimilarity matrix of objects. Different methods differ in the way similarities or dissimilarities are calculated after two clusters are joined. Table 5.1 summarizes different calculations of similarity or dissimilarity used. This table uses the following notations:

$d^{new}_{(p+q),i}$ = dissimilarity between the new cluster $(p+q)$ with cluster $i$

$d_{pi}$ = dissimilarity between the cluster $p$ and cluster $i$

$s^{new}_{(p+q),i}$ = similarity between the new cluster $(p+q)$ with cluster $i$

$s_{pi}$ = similarity between the cluster $p$ and cluster $i$

Table 5.1: Dissimilarities and similarities for different hierarchical clustering techniques [Everitt 1981]

| *Method* | *Calculation of dissimilarities and similarities* |
|---|---|
| Complete linkage | $d^{new}_{(p+q),i} = \max(d_{pi,} d_{qi})$ *res.* $s^{new}_{(p+q),i} = \min(s_{pi}, s_{qi})$ |
| Single linkage | $d^{new}_{(p+q),i} = \min(d_{pi}, d_{qi})$ *res.* $s^{new}_{(p+q),i} = \max(s_{pi}, s_{qi})$ |
| Average linkage | $d^{new}_{(p+q),i} = \left(d_{pi} + d_{qi}\right)/2$ |
| Weighted average linkage | $d^{new}_{(p+q),i} = \left(n_p \cdot d_{pi} + n_q \cdot d_{qi}\right)/\left(n_p + n_q\right)$ |
| Within average linkage | $d^{new}_{(p+q),i} = \left(n_p \cdot d_{pi} + n_q \cdot d_{qi}\right)/\left(n_p + n_q\right)$<br>$d^{new}_{(p+q),(p+q)} = \frac{\left(\frac{2}{n_p \cdot (n_p - 1)} \cdot d_{pp} + \frac{2}{n_q \cdot (n_q - 1)} \cdot d_{qq} + n_p \cdot n_q \cdot d_{pq}\right)}{\left(n_p + n_q\right) \cdot \left(n_p + n_q - 1\right)/2}$ |
| Median linkage | $d^{new}_{(p+q),i} = \frac{1}{2} \cdot d_{pi} + \frac{1}{2} \cdot d_{qi} - \frac{1}{4} \cdot d_{pq}$ |
| Centroid linkage | $d^{new}_{(p+q),i} = \frac{n_p}{n_p + n_q} \cdot d_{pi} + \frac{n_q}{n_p + n_q} \cdot d_{qi} - \frac{n_p \cdot n_q}{(n_p + n_q)^2} \cdot d_{pq}$ |

The following is a characteristic description of different methods:

*Complete Linkage* uses a max function and results in a strong definition of the homogeneity of clusters: “The largest dissimilarity between all objects of one cluster should be less than a certain value”. The farthest neighbor of each object should have a distance less than a certain value. In contrast to complete linkage, *single linkage* only requires that the nearest neighbor is located within a certain distance. Thus, the method is called nearest neighbor. Both single and complete linkage methods can be used for similarity or dissimilarity measures.

The advantage of single and complete linkage is their invariance against monotonic transformation of the dissimilarities or similarities. If the dissimilarities or similarities

are squared or log-ed, the results do not change. A disadvantage of single linkage is the potential for *chaining*, which causes too few large and heterogeneous clusters. In contrast, complete linkage may produce too many small clusters.

The large heterogeneous clusters resulting from using single linkage and the many small clusters produced using complete linkage resulted in average linkage approaches. Average linkage, weighted average linkage, and within average linkage try to avoid these effects by using an average of the distances rather than a minimum or maximum function. The remaining methods characterize the cluster by selecting an existing object as a median or centroid, or generating a new object that then represents the cluster in further calculations.

Agglomeration can be visualized by “dendrogram”, as shown in Figure 5.1. The dendrogram shows which objects are combined in which step. Objects $O_1$ and $O_2$ are combined in the first step. In the second step $O_3$ is merged into the cluster built by $O_1$ and $O_2$. In the third step $O_4$ and $O_5$ are combined. Finally all objects are combined into one cluster.

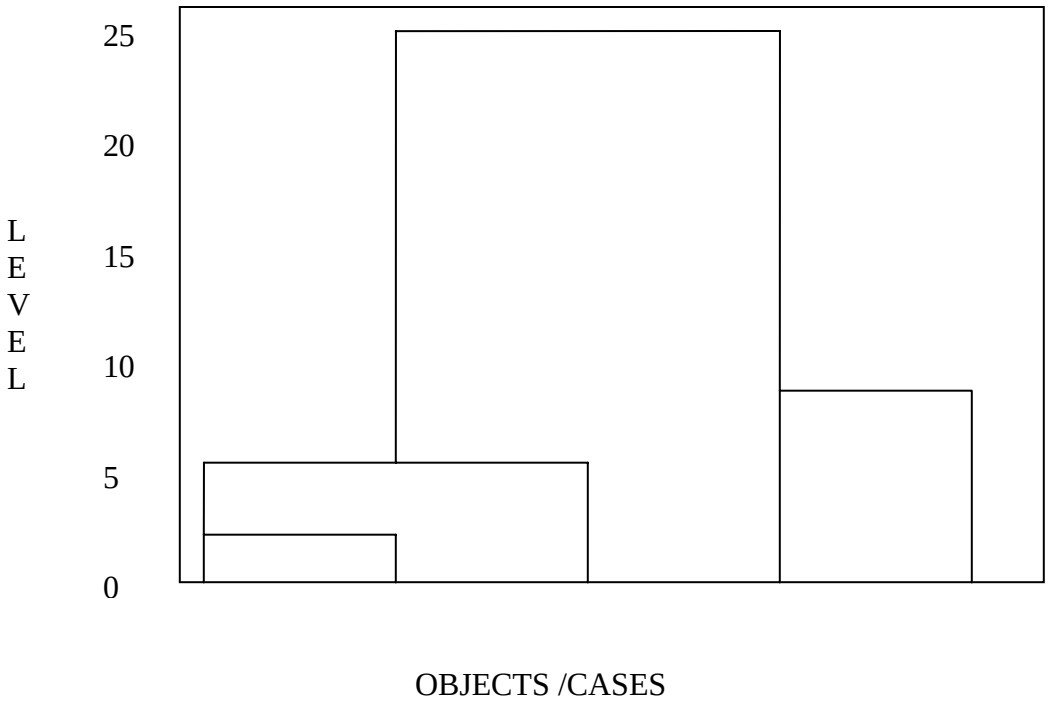



Figure 5.1: Dendrogram showing the agglomeration of objects based on distance

# 6. CLUSTERING VKB HISTORY

Three different clustering algorithms were designed and implemented for grouping VKB edit history. The three implementations were developed independently and represent significantly different approaches. The first approach is a pattern matching approach that identifies repeated patterns of edit events in the history. The second approach is a rule-based approach that uses simple rules, such as “group all consecutive events on a single object” or “group all consecutive events within a collection”. The third approach uses hierarchical agglomerative clustering (HAC) where edits are grouped based on a function of edit time and edit location.

The three different approaches used in this thesis are supported by the work of Ping-Herng Denny Lin [StatSoft 2005]. His comparison of data mining methods for classification purposes overlaps this efforts goal of grouping into meaningful edit sequences. Aggrawal & Yu’s [Aggarwal 1998] survey of data mining techniques suggests a successful process for identifying meaningful co-occurrences that can combine association rules, clustering and then classification.

## 6.1 PATTERN MATCHING APPROACH

This approach finds repetitive patterns of events in the VKB edit history. It uses techniques similar to string matching [Capelle 2002, Garofalakis 1999] and to the pattern identification algorithms used for macro recognition in programming by demonstration (PBD). To identify patterns, each type of event is encoded as a unique element. An example is “ACRM” where A is an “Add Symbol” event, C is a “Change Symbol” event, R is for “Resize”, and M is for “Move”. In this approach, a sequence of events appearing consecutively, in the real chronological order of events, is termed a “*token*”.

The *length* of this token is defined as the number of events in it. This approach searches for tokens between 4 and 20 elements in length that appear frequently within the event history. Considering patterns with length smaller than four will generate too many instances of repeated events, most of which will either have no meaning to users or represent an incomplete activity in the eyes of users. Considering patterns over length 20 is very unlikely to be useful in real applications. The objective of this approach is to find tokens of such a length that they conform to activities that users view as coherent.

In developing this approach, it was decided to identify patterns of events on *objects* and *collections*. This produces patterns of events that represent manipulations of objects, manipulations of collections and manipulations of combinations of objects and collections. An example is "*after creating a new collection and changing its size and background color, the user creates two objects and modifies their textual content, shape and color*". When such a pattern identification algorithm is applied to a long edit history, lots of repeated patterns can be found. Whether these patterns make sense to users depends on whether the groupings indicate a sequence of low-level actions that together instantiate a sub-task or sub-activity in the workspace (e.g. creating an new object or list of objects with certain content). To reduce this problem, user-defined constraints can be used to focus the mining task on interesting patterns [Capelle 2002].

A limitation of the current pattern matching implementation is that sequences of events must be identical in length and composition to be considered equivalent. The inclusion of wildcards or regular expressions would address this but makes the generation of potential patterns problematic. Then another algorithm must be designed to decide what to generalize and abstract into a wildcard or regular expression? Another limitation is that time gaps between edit events and the context of the event, e.g. the collection in which the event occurs, do not impact the grouping.

## 6.2 RULE-BASED APPROACH

The rule-based approach groups VKB history events using rules based on observations of how users interact with VKB. This approach is somewhat similar to the rule-based approach to word clustering used by Hui, Han, et. al [Han 2003] in their text classification system. The implementation of this approach for this thesis is very simple and consists of two rules. The first rule groups contiguous events that modify the content, position or visual attributes of a particular object or collection in the workspace. This rule groups local actions on an entity in the workspace. The second rule groups all consecutive events that occur within a particular collection. This rule runs after the first rule and can generate groups that include groups generated by the first rule. The result of this process is groups of events in a particular subspace in the VKB workspace.

Figure 6.1 shows a scenario in VKB history, where $C_i$ represents collection i, $O_j$ represents object j, and '<>' is the top-level collection (or desktop) of a VKB workspace. The results of applying the first rule cause events $e_2$, $e_3$, $e_4$, $e_5$, applied on object $O_2$, to be grouped together. $O_2$ is in the top-level collection (shown as <>). Another 4-event grouping is $e_7$, $e_8$, $e_9$, $e_{10}$ as they are all manipulation of object $O_1$.

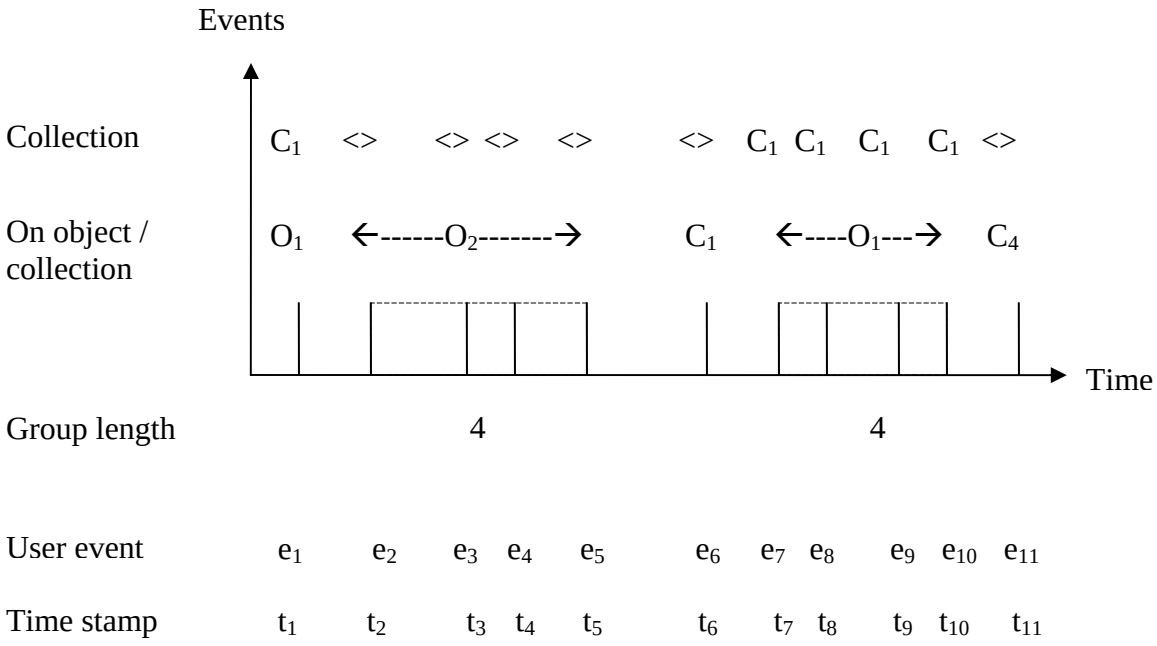



Figure 6.1: Grouping events on single element

Figure 6.2 shows the application of the second rule. Object $O_1$ has 3 events applied on it, then object $O_2$ has 2 events, and then object $O_1$ has 3 more events. After a manipulation of the Collection $C_1$, there are edits to Objects $O_4$, $O_5$, and $O_6$ in Collection $C_2$. The first rule will group together the events on a single object as previously described. Events $e_1$ through $e_8$ are applied on different objects contained in the collection $C_1$, and so the three groups of events generated by the first rule will be grouped together creating a two-level hierarchic grouping of the events. In this example, $e_9$ , which is an event on $C_1$ is left out of any groups and then the groups and individual edits in $e_{10}$ to $e_{17}$ are grouped as they all occur in collection $C_2$.

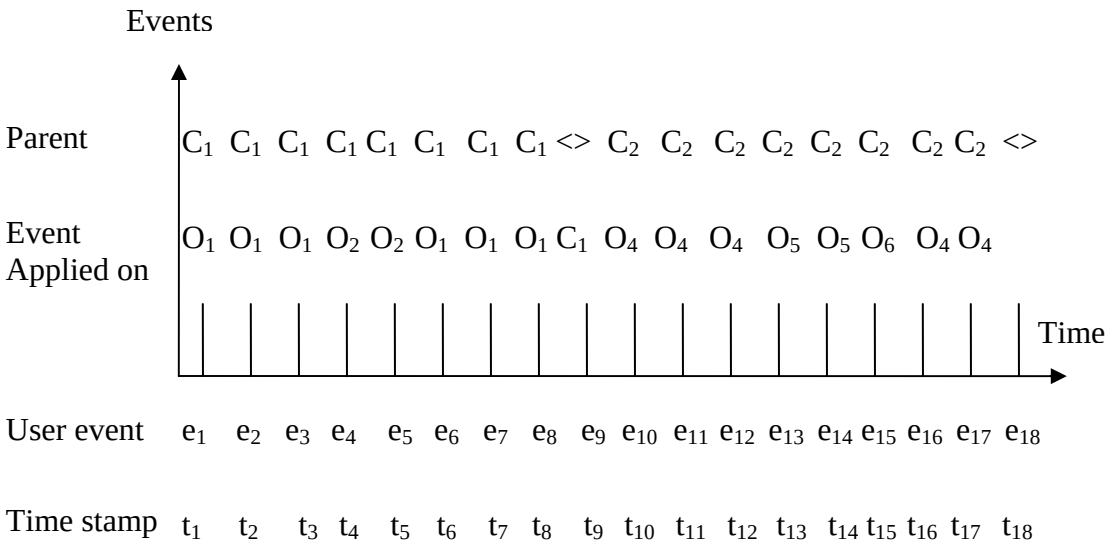



Figure 6.2: Grouping events in collection

While not part of the implementation in this thesis, additional layers of grouping could be added based on whether consecutive edits were to a particular spatial structure recognized by the VKB spatial parser. As the spatial parser is meant to recognize the hierarchy of perceptual structures created by users to express relationships about the information in the workspace, automatic grouping based on this structure would provide further levels of granularity in the edit history.

The primary weakness of this approach is the limited forms of groups recognized. Unlike the pattern-matching approach, there is no impact on the type of edits being performed on an object or within a collection. As long as two edits are consecutive and are to an individual object or within a particular collection, they will be grouped. Similar to the pattern-matching approach, time is also not included in this particular instantiation although it would be easy to add a cutoff to the rules such that edits with a certain gap in-between would not be grouped.

## 6.3 HIERARCHICAL AGGLOMERATIVE CLUSTERING (HAC)

The last approach for grouping history explored in this thesis uses hierarchical agglomerative clustering (HAC). HAC groups elements based on a similarity function. In this implementation, event dissimilarity is calculated based on a linear function of the time and space (position) of the VKB history. This is somewhat like Chiu and Wilcox's use of *time* and *space* for hierarchical agglomerative clustering in dynamically grouping user strokes in a pen-based and audio system [Chiu 1998]. Hierarchical agglomerative clustering (HAC) methods have also been used in [Yeung 1998], [Gatica-Perez 2001], and [Zhao 2001] in video analysis. Work in [Zhong 1996], [Yeung 1998], [Rui 2000] also proposes *visually-based* and *time-constrained* clustering.

The dissimilarity function is based on the time and space differences between history events. Events in the VKB history include an absolute timestamp. For the HAC analysis, a relative time of appearance for each event is derived from the actual timestamps. For each event there is one an associated workspace element (e.g. object or collection). These workspace elements have a spatial location/position within. Spatial distance between two events is calculated based on the distance between the workspace elements manipulated in the two events.

To be specific, consider the following example where $e_1$ and $e_2$ are events in VKB. Event $e_1$ appears at time $t_1$ and $e_2$ appears at time $t_2$. Thus $\Delta T = t_1 - t_2$ (where $t_1 > t_2$). Event $e_1$ was applied to symbol $S_1$ and $e_2$ was applied to symbol $S_2$. $P_1(x_1,y_1)$ and $P_2(x_2,y_2)$ are the positions of the symbols $S_1$ and $S_2$ respectively in the VKB space. The distance between symbols $S_1$ and $S_2$ is $\Delta S$, which is calculated as described in the next paragraph. The dissimilarity between event $e_1$ and $e_2$ is $d = A*\Delta S + B*\Delta T$, where $A$ and $B$ are coefficients.

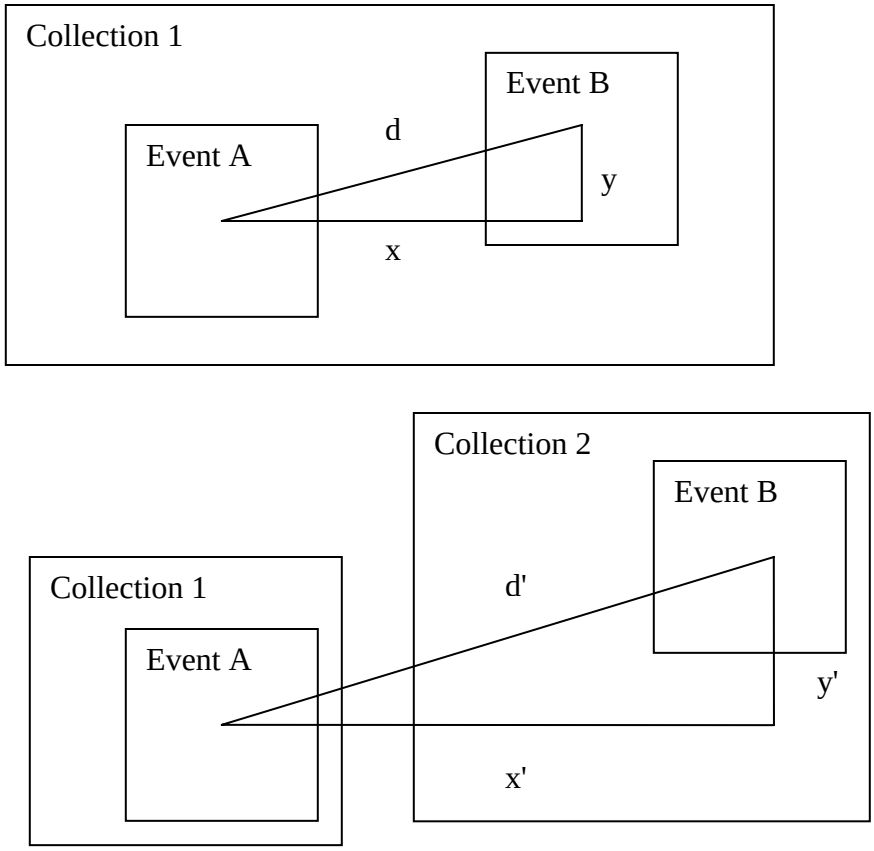



Figure 6.3: Calculation of spatial distance among nodes

For calculating spatial distance between two events, different strategies are applied based on whether the two events occur within the same collection. If the two events manipulate symbols within the same collection then their spatial distance is calculated to be the Euclidean distance between those symbols based on their x and y coordinates. If they belong to different collections, the spatial distance is calculated by adding 2000 to the Euclidean distance between the symbols. This value is chosen based on the maximum value of all spatial distances between two symbols in a single collection. In this current dataset that maximum distance is 1057. Hence the addition of 2000

guarantees that the events are treated as if they were further away than they would have been if they had been in the same collection. Figure 6.3 shows how spatial distance is calculated for two events in same and different collections. Here $d = \sqrt{(x^2 + y^2)}$ and $d' = 2000 + \sqrt{(x'^2 + y'^2)}$.

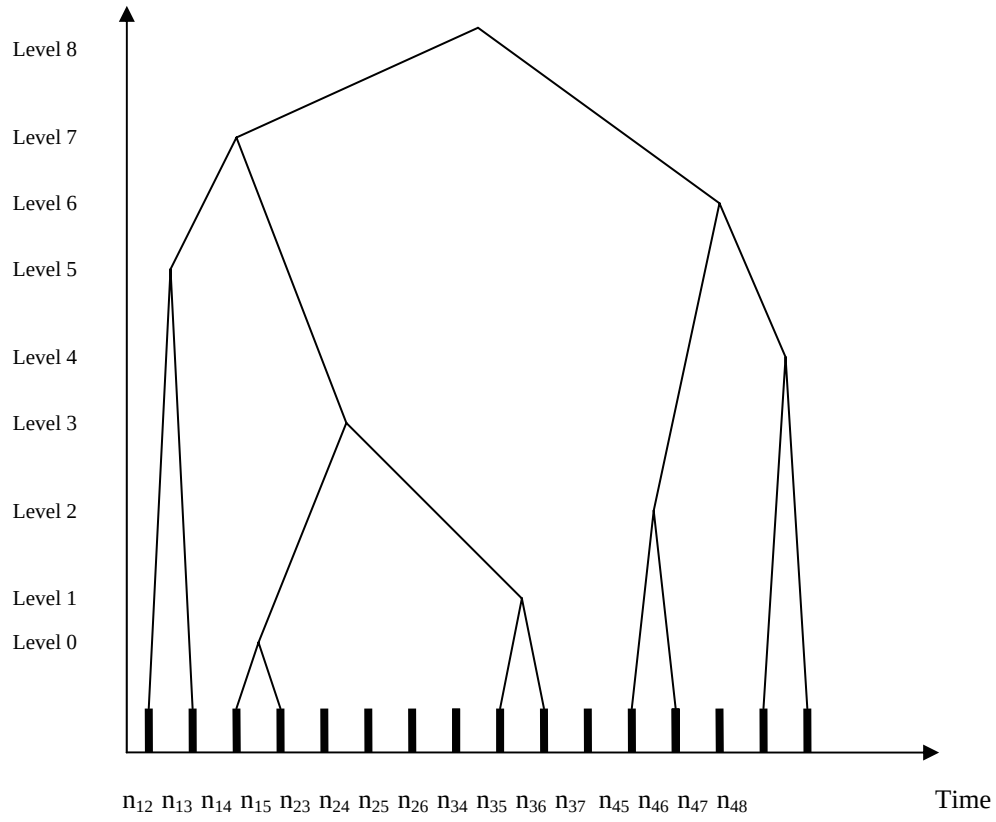



Figure 6.4: Non-threshold clustering process

Two different methods are taken in the implementation of the bottom-up clustering. In the first method, objects are clustered by comparing the nearest neighbors. The process begins in such a state that each item in the event set forms a cluster of one element. In each step, a search is made to find the best clusters to merge (e.g. those that are most

similar to one another) and combines them. The procedure gradually creates larger clusters with possibly increasing number of events in it. At the end of the process a single cluster with all the events in it is created – this is the root node of a tree with depth *n-1* for *n* original elements. In order to generate potentially meaningful groupings, once the clustering is complete, levels of similarity must be identified at which groups will be produced. Figure 6.4 shows a partial clustering tree with eight levels. A brute-force implementation of this approach is $O(n^2)$ in time, where *n* is the number of events in the history.

In the second method as shown in Figure 6.5, clustering is done based on *threshold values*. As before, all events initially form a separate cluster of one element. At each stage, an indicative *threshold value* is chosen to form larger clusters at the next higher level in the hierarchy. By increasing the *threshold values* gradually, more events are included in different clusters and larger clusters are formed at each level. The last level of hierarchy includes all of the events. Figure 6.4 show this method. A simple implementation of this algorithm takes $O(n*m)$ time, where *m* is the number of thresholds considered.

In this thesis, the clustering method using *threshold values* has been used. The initial *threshold value* is obtained by considering the inter-distance among events at the lowest level. In the lowest level, after calculating inter-distance among all the pair-wise neighboring events, the minimum and maximum among them are found. The initial threshold value starts from this minimum and reaches up to the maximum value by increasing 10% at each step.

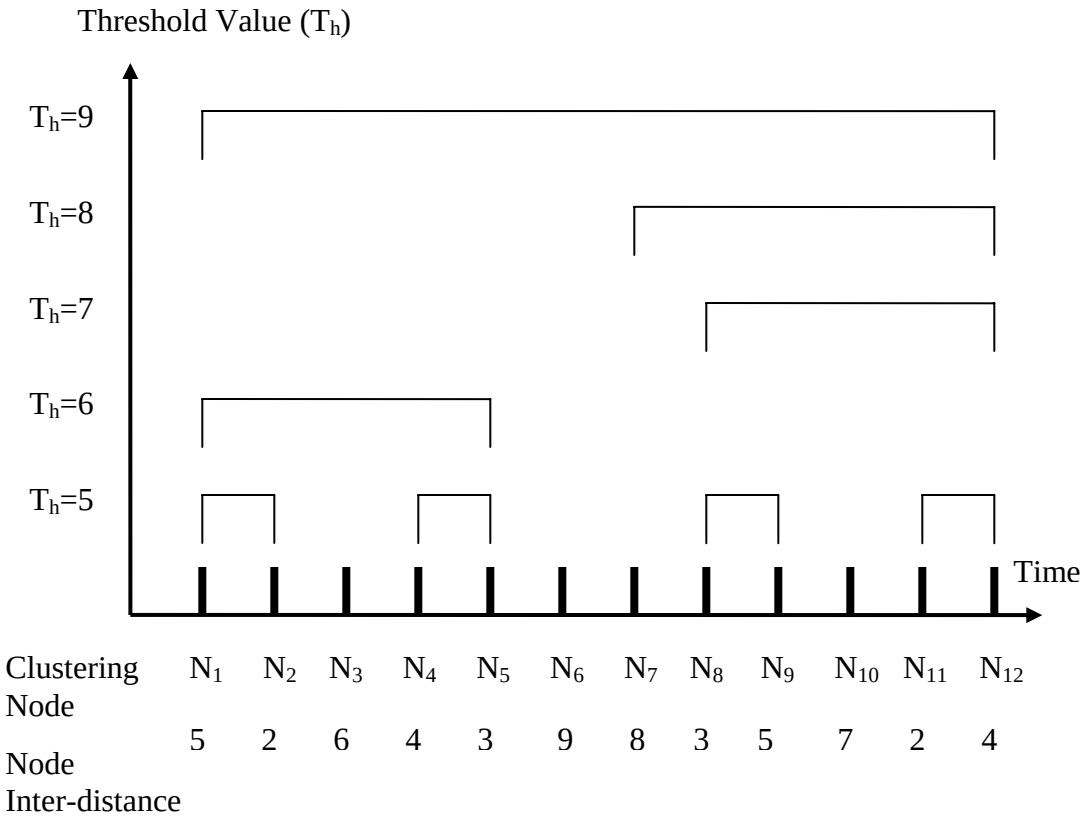



Figure 6.5: Clustering process using threshold method

# 7. COMPARISON OF CLUSTERING ANALYSIS

The objective of this thesis work is to group VKB edit history automatically in such a way that they will be similar to if the user had done the grouping manually. By implementing and comparing the three approaches, this thesis attempts to discern the strengths and weaknesses of the three approaches.

Hey-Chung Kum, et. al [Kum 2002] examine the problem of mining sequential patterns and propose a comprehensive evaluation method to assess the quality of the mined results. They propose four evaluation criteria. The methods are: (a) recoverability (b) the number of spurious patterns (c) the number of redundant patterns and (d) the degree of extraneous items in the patterns, to quantitatively assess the quality of the mined result from a wide variety of synthetic datasets with varying randomness and noise levels.

## 7.1 EXPERIMENTAL DATA

The different approaches described in the previous section to analyze VKB history were applied to a real-world dataset generated over a long period of time in the Center for the Study of the Digital Libraries (CSDL) at Texas A&M University. This file contained the event history captured during weekly meetings over a period of over four years. The following is a discussion of different observations about the CSDL VKB data file.

There are a total of 4138 events applied over 371 symbols (both objects and collections) in this VKB workspace. Out of 371 symbols, there are 21 collections and 350 objects in the file. Out of 7134 events, 433 events are applied on collections and 3699 events are applied on objects. Remaining events are applied on desktop or other entity in the VKB workspace.

Events in the history include a timestamp to indicate the actual time the event occurred. The value of the timestamp when the first event appeared is ‘941663618480’ and that for last event is ‘1080768901707’. These values represent number of milliseconds elapsed after 12:00:00 PM January 1, 1900. For the ease of analysis, this real-time value was converted into a relative timestamp for computing the time gap between events. Table 7.1 shows the distribution of events during the multiple year (230 week) authoring process.

Table 7.1: Distribution of events across the multi-year authoring process

| Serial number | Relative time of appearance range (in weeks) | Events sequence id | Number of events appearing during the time slot |
|---|---|---|---|
| 1 | 0-50 | 1-1579 | 1579 |
| 2 | 51-100 | 1580-2848 | 1268 |
| 3 | 101-150 | 2849-3323 | 474 |
| 4 | 151-200 | 3324-3540 | 216 |
| 5 | 200-230 | 3541-4137 | 596 |

One aspect of this distribution is that there is almost no activity or event occurrence except at these weekly meetings. Most of the events occur on a specific day of the week and within a one or two hour period. Figure 7.1 shows the appearance of events over a period of 30 weeks. From this graph it is evident that events appear as a cluster once in a week. This graph shows some discontinuity due to meetings not occurring every week or meetings where the data file was not edited. Figure 7.2 shows the distribution of events for one meeting over a period of approximately one hour. It is clear that there are periods

of activity in the meeting involving edits to the data file and other periods of activity where no edits occur.

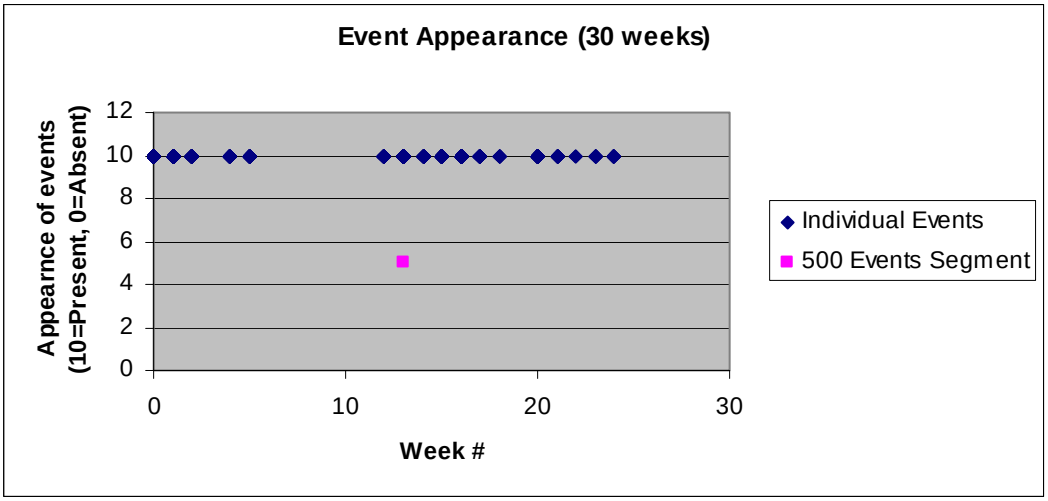



Figure 7.1: Event appearance -30 weeks

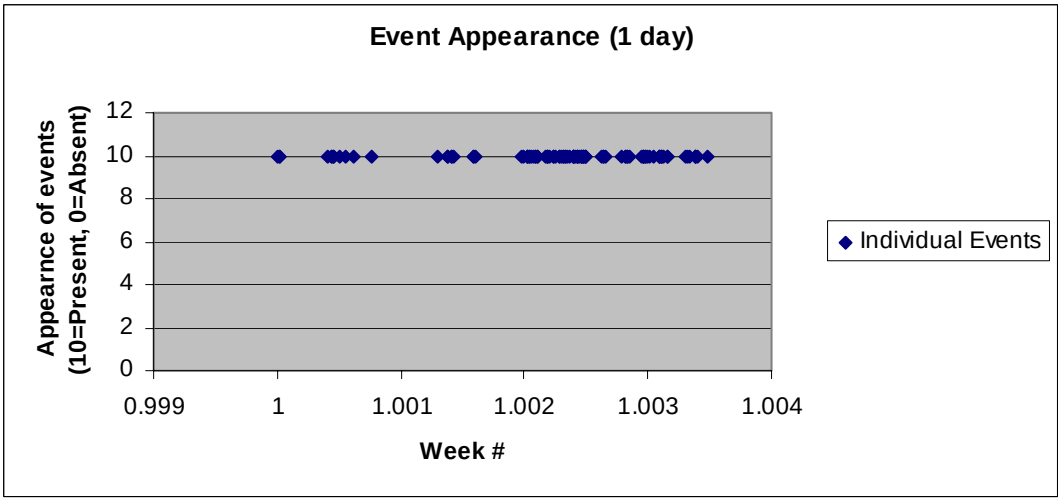



Figure 7.2: Event appearance during part of one day

For the ease of handling data and efficient implementation of pattern based analysis, entire event set has been grouped into 8 segments. Each segment contains 500 events in it except the last segment which contains 638 events. The segmentation was necessary for the pattern recognition based approach, as comparing and analyzing the entire event set is time consuming. The disadvantage of this segmentation of the original event set is that it can skip some event sequence (at most 8 chunks at 8 segment boundaries) due to this discontinuity.

Table 7.2: VKB operation summary

| Serial Number | Operation Type | Operation code |
|---|---|---|
| 1 | Add Symbol | A |
| 2 | Change Border Color | B |
| 3 | Change Content of item | C |
| 4 | Delete Symbol | D |
| 5 | Change Font | F |
| 6 | Change Background Color | G |
| 7 | Change Color index | L |
| 8 | Move Symbol | M |
| 9 | Minimize | N |
| 10 | Change Border Pallet | P |
| 11 | Resize Symbol | R |
| 12 | Change Transparency | T |
| 13 | Change Attribute | U |
| 14 | Change Border Width | W |
| 15 | Maximize | X |
| 16 | Change Font Color | l |
| 17 | Change Background Pallet | p |

The CSDL data file used in this thesis does not contain all the operations/activities defined/available in VKB. The history recorded in this data file contains 17 different operations applied over the VKB objects/collections. The operations are coded using alphabetic codes for ease of identification. Table 7.2 displays the operations included in this VKB history file and their associated codes.

## 7.2 RESULTS AND OBSERVATIONS FOR THE PATTERN-BASED APPROACH

One observation of event sequences is the repetition of a single event type. This common phenomenon is observed in almost all methods. Different repetition sequences with length up to 20 were found. An example repetitive sequence is "MMMMMMMMMMMM". In this sequence the same operation M ("Move Symbol") is applied again and again. Most likely, these represent a period of reorganization on the part of the user, although this information alone cannot distinguish between activity within a single collection and activity that happens across multiple collections. The Move (M), Add (A), Create (C), and Change Attribute (U) events are found with highest sequence length. Repetition sequence for operations Change Transparency (T), and Change Background Pallet (p) are found with smaller sequence length.

Another observation is that many of these repetitions had a distinct beginning or ending operation. For example, "CMMMMM", "MMMMMC", "MMMMMA", "WMMMMM" were found in many places. Other patterns found with almost similar characteristics are <x>MMMMMMM<y>. Here, an event is repeated with a very small subset of events at the beginning and the end. Another interesting observation is a sequence like "AC AC AC", "ACR ACR ACR". The sequences AC and ACR are repeated 3 times. Finally, repetitions like "MMMMACRMMMMM", where Move (M) is repeated with sufficiently high appearances before and after a small subset of other activity, in this case "ACR". In general this type of repetition may involve the subset in any place within

the entire sequence. Some of the sequences belonging to this category are MMMMACMMMMMM, MMCRMMMMMM, and MMMMMMCRMM.

Table 7.3: Token with maximum length from pattern based method

| Token | Length | Seg. 1 | Seg. 2 | Seg. 3 | Seg. 4 | Seg. 5 | Seg. 6 | Seg. 7 | Seg. 8 | Total |
|---|---|---|---|---|---|---|---|---|---|---|
| MMMMMMMMMMM MMMMMCMM | 19 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 2 |
| MMMMMMMMMMM MMMMMCRM | 19 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| MMWMMMMMMMMMMM | 15 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 2 |
| AUUUUUUUUUUU | 13 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| MMMACRMMMMMM | 13 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 2 |
| MMMMMMMACRMMM | 13 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 2 |
| MMMMMMMMMMRMM | 13 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 2 |
| MMMMRMMMMMMMM | 13 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| MMMMACMMMMMM | 12 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 2 |
| MMMMRMMMMMMM | 12 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 2 |
| MMMMACRMACR | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 |
| MMMMACRMMMM | 11 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 2 |
| MMMMMMMMMMF | 11 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| RMMMMMMMMMM | 11 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| WMMMMMMMMMM | 11 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 2 |
| ACMMMMMMMM | 10 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 3 |
| ACRACRACRM | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 |
| CRMMMMMMMM | 10 | 0 | 0 | 0 | 2 | 0 | 2 | 0 | 0 | 4 |
| MACRMMMMMM | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 |
| MCCCCCCCCM | 10 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 2 |
| MMACMMMMMM | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 |
| MMMACRMMMM | 10 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 2 | 5 |
| MMMMACRMMM | 10 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 3 |
| MMMMMACRRG | 10 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 2 |
| MMMMMCMMMM | 10 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 2 |

The pattern-based approach found the following repetitive patterns with the above characteristics: RMMM, MMMR, RMMR, TTTTT, RMMMMMMMMM, MMMMMMMMAC, RMMMMMMMMM, CMMMMMMMM, MMMMMMMAC, RMMMMMMMM, CMMMMMMM, MMMRMMMM, ACRACRACRM, and CMCMCM.

The pattern-based method produced a total of 403 patterns of length 4 or higher. On the higher end patterns found with maximum length of 19 are MMMMMMMMMMMMMMMMMCMM and MMMMMMMMMMMMMMMMCRM. On the lower end, five example tokens with a length of 4 are MGAC, MARC, GMAC, BACW and WMAC. Table 7.3 lists the 25 longest tokens.

Table 7.4: Token with maximum frequency of appearance from pattern based method

| Token | Length | Seg. 1 | Seg. 2 | Seg. 3 | Seg. 4 | Seg. 5 | Seg. 6 | Seg. 7 | Seg. 8 | Total |
|---|---|---|---|---|---|---|---|---|---|---|
| ACRM | 4 | 0 | 5 | 0 | 5 | 3 | 0 | 0 | 54 | 67 |
| RMMM | 4 | 2 | 18 | 8 | 9 | 0 | 6 | 9 | 8 | 60 |
| MMAC | 4 | 2 | 0 | 0 | 16 | 12 | 13 | 4 | 9 | 56 |
| RMMMM | 5 | 0 | 13 | 4 | 7 | 7 | 5 | 0 | 5 | 41 |
| MACR | 4 | 0 | 4 | 0 | 0 | 3 | 13 | 4 | 15 | 39 |
| MMRM | 4 | 2 | 15 | 2 | 8 | 0 | 0 | 9 | 0 | 36 |
| CRMM | 4 | 0 | 6 | 0 | 5 | 4 | 7 | 5 | 8 | 35 |
| MRMM | 4 | 0 | 12 | 2 | 7 | 4 | 0 | 6 | 0 | 31 |
| MMMR | 4 | 0 | 14 | 3 | 7 | 6 | 0 | 0 | 0 | 30 |
| MMMMAC | 6 | 0 | 0 | 6 | 6 | 6 | 0 | 0 | 6 | 24 |
| CMMMM | 5 | 0 | 0 | 8 | 7 | 5 | 0 | 4 | 0 | 24 |
| MACM | 4 | 3 | 2 | 3 | 6 | 9 | 0 | 0 | 0 | 23 |
| ACAC | 4 | 0 | 0 | 12 | 0 | 5 | 0 | 3 | 0 | 20 |
| MACRM | 5 | 2 | 2 | 0 | 3 | 0 | 0 | 0 | 12 | 19 |
| MMMAC | 5 | 0 | 2 | 9 | 0 | 0 | 0 | 0 | 8 | 19 |
| CMMM | 4 | 0 | 2 | 10 | 0 | 7 | 0 | 0 | 0 | 19 |
| RMAC | 4 | 0 | 5 | 0 | 2 | 0 | 0 | 0 | 12 | 19 |
| ACMM | 4 | 0 | 0 | 9 | 4 | 5 | 0 | 0 | 0 | 18 |
| MMMRM | 5 | 0 | 12 | 0 | 0 | 0 | 0 | 5 | 0 | 17 |
| MMRMM | 5 | 0 | 8 | 0 | 5 | 0 | 0 | 4 | 0 | 17 |

Different tokens found in the VKB file appear with different frequency in the entire event space. Token ACRM appears with a highest frequency, 67 times over the entire history. Table 7.4 contains the 20 tokens with the highest frequency of appearance.

Table 7.4 also shows the distribution of tokens among the eight segments of history data. It is interesting to note that 54 of the 67 appearances of ACRM are in the last segment of the history data. This indicates that this may be the result of some feature added to VKB towards the end of the four year period. As indicated, some of the patterns appear more frequently at the beginning of the event space while others appears more frequently at the end of event space. Figure 7.3 graphs the distribution of tokens among different segments those appears with highest frequency. The label *token_freqa*, *token_freqb*, ……, *token_freqh* represents the number of occurrences in segments 1, 2, 3… 8 respectively.

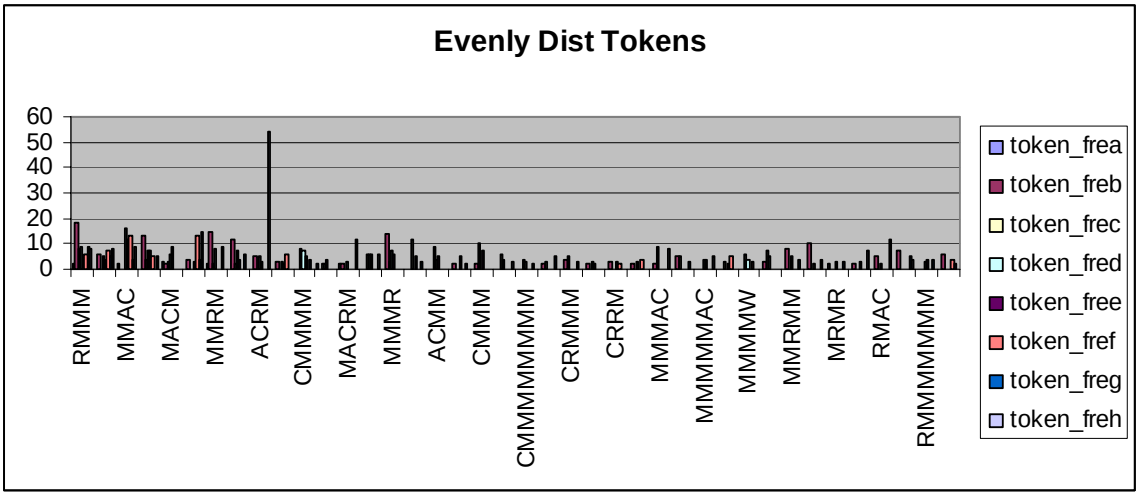



Figure 7.3: Distribution of tokens among different segments

## 7.3 RESULTS AND OBSERVATIONS FOR THE RULE-BASED APPROACH

The rule-based method produced a total of 174 tokens. The longest tokens were of length 10. Unlike the pattern-based approach, the rule-based approach allows tokens with length smaller than 4. Table 7.5 shows longest 25 tokens.

Table 7.5: Tokens of maximum length in rule based approach

| Token string | Token length |
|---|---|
| ACRMMMMMMMM | 10 |
| ACRMMCRCCRM | 10 |
| ARCRRGGMFR | 9 |
| ACRFMRTCM | 8 |
| MTTTTTTTT | 8 |
| ACRMMBMWR | 8 |
| CCCCCCCCC | 8 |
| ARCRGRMMM | 8 |
| FFFFlFFl | 7 |
| ACRMRCRM | 7 |
| WWMGGBWR | 7 |
| ACRMGTMM | 7 |
| CCCCCCCC | 7 |
| CCCCCCCC | 7 |
| ACRCRRMM | 7 |
| MMMMMMMM | 7 |
| ARGRCRMM | 7 |

Table 7.5 Continued

| Token string | Token length |
|---|---|
| ACMCRCRR | 7 |
| MMFRMMMF | 7 |
| ACRRMMRW | 7 |
| ACRCRRCR | 7 |
| MMRCRCR | 6 |
| GGWBBBG | 6 |
| CRRCRCR | 6 |
| ACGWWGG | 6 |

Tokens that had the highest frequency of appearance based on the rule-based approach are shown in Table 7.6. Notice that the frequencies are considerably smaller than those found by the pattern-matching approach.

Table 7.6: Token with maximum frequency of appearance from rule based method

| Token String | Frequency |
|---|---|
| ACRM | 23 |
| ACRC | 7 |
| ACRMM | 7 |
| MMMM | 7 |
| ACRG | 6 |
| ARCR | 6 |
| ACRRM | 5 |
| ACRWM | 3 |
| CCCCCCCC | 3 |
| CMCM | 3 |
| MMRM | 3 |
| ACGM | 2 |
| ACGR | 2 |
| ACMC | 2 |
| ACMR | 2 |

Due to the rule-based method's grouping of activity on a single object or collection, the method identified a variety of different tokens applied to same symbol over the course of the event history. Table 7.7 shows different tokens applied to the same symbol.

Table 7.7: Events applied on same symbol

| Symbol Id | Parent ID | Number of Tokens | Token String |
|---|---|---|---|
| 17 | 0 | 3 | ACGMX, MMRM, WMMM |
| 52 | 0 | 4 | MMMM, RMMMM, MMMMM, RRMMM |
| 157 | 52 | 3 | ACRG, MGBMRC, TTTT |
| 689 | 52 | 3 | ACGWWGG, MTTTTTTTT, MMRM |
| 788 | 52 | 5 | ACRC, ACRGW, CCCCCCCC, CCCCCCCC, CRRC |
| 4535 | 17 | 5 | RFlR, RRMMM, FlRRFR, BWRBG, RFMRM, FMRM |

## 7.4 RESULTS AND OBSERVATIONS FOR THE HIERARCHICAL CLUSTERING APPROACH

The hierarchical-agglomerative clustering approach relies heavily on the computation of similarity, in this case a linear function of the time and space distances between events. The following analysis provides a sense for these distance values and their distribution.

There were a total of 3291 potential nodes (pairs of consecutive events) generated. The maximum time between two subsequent events found was 18044945643 milliseconds, or 208 days. Clearly, there was a large period of time when meetings did not occur or the data file was not used. (Indeed, the faculty member running the group was on sabbatical for a year and so weekly meetings were much less frequent and of a different character.) The minimum time between events was 10 milliseconds, potentially indicating events added by a single user edit.

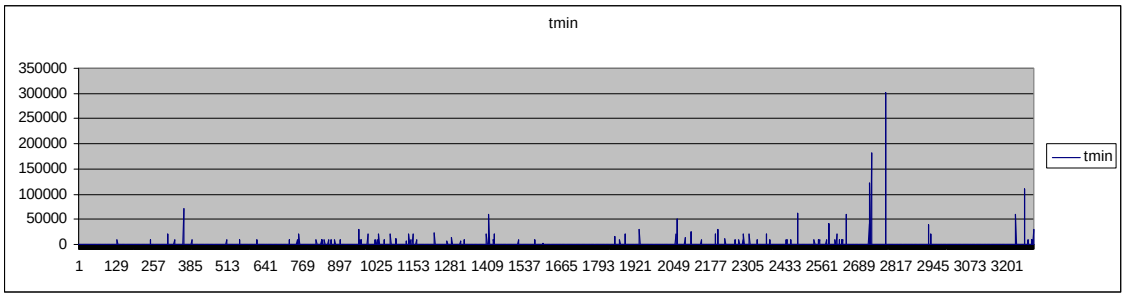



Figure 7.4: Time between events in the edit history

Figure 7.4 show the time between subsequent events in the history record. As can be seen, the vast majority of edits occur relatively few minutes apart. This is not surprising as a time gap of approximately 10000 minutes would be seen for weekly meetings. Figure 7.5 shows the distribution of the time between events.

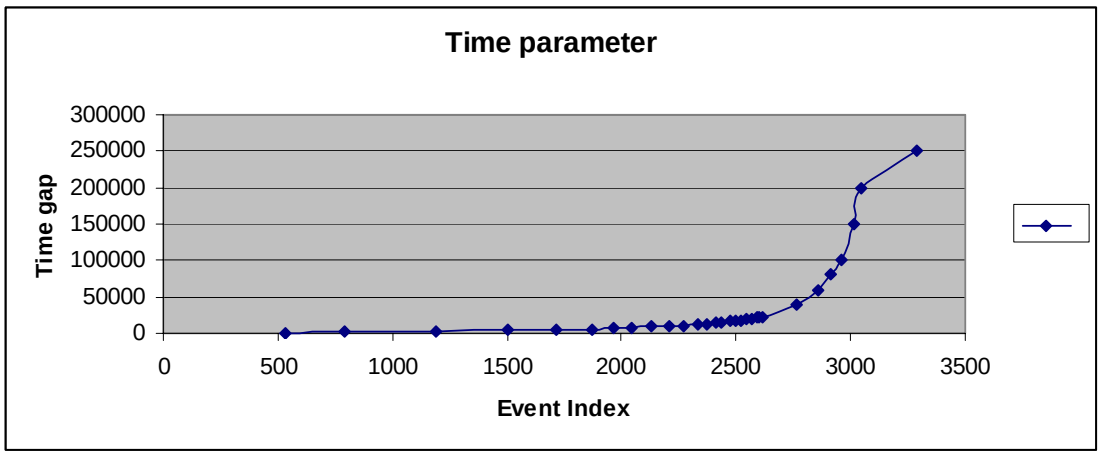



Figure 7.5: Distribution of time gap between subsequent edits

Of the 3291 pairs of consecutive events, 837 pairs (25%) were in different collections. For events within a single collection, the highest spatial distance was 1057.62. When

two operations were on the same object, the spatial distance is 0 – there were 1320 pairs (40%) where this was the case. Figure 7.6 shows the spatial distance between two consecutive events. Remember that 2000 was added to the computed distance when the events occurred in different collections.

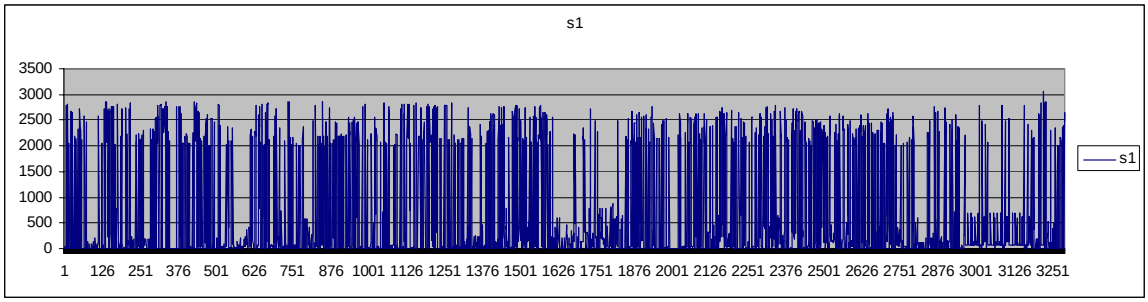



Figure 7.6: Spatial distance between events in the edit history

Figure 7.7 shows the distribution of spatial distances between events. It consists of two parabolic curves, one under 2000 for those event pairs that were within a single collection and one at or above 2000 for event pairs involving different collections.

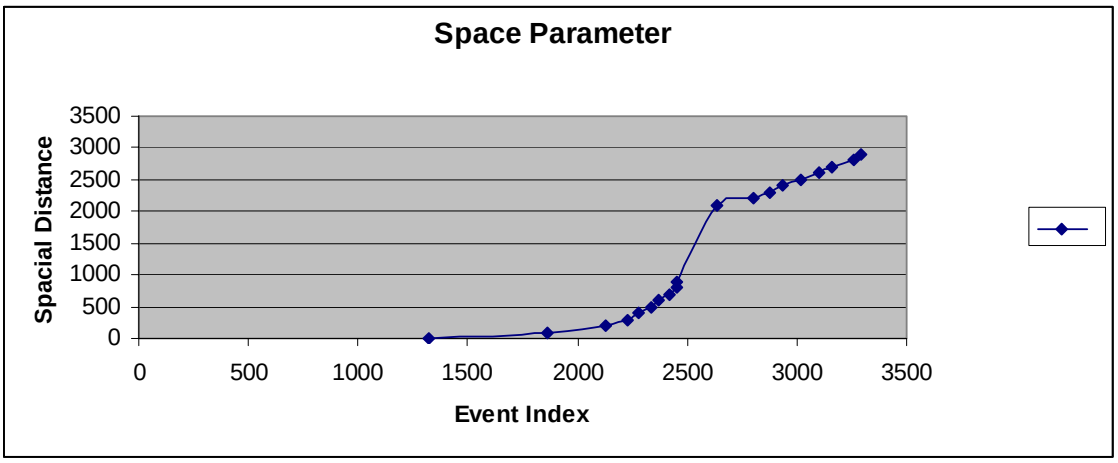



Figure 7.7: Distribution of spatial distances between subsequent events

The overall similarity function is calculated based on the linear function $d=s+t$ where both the coefficients of the linear function have been assumed to be 1. For clustering purposes, a threshold value of up to 30010 was considered. Out of 3291 edit events, 2628 events (79%) are covered within this threshold value, indicating their membership in a group. Thus, this approach generated lots of groups that included the majority of the edit events in the recorded history.

## 7.5 COMPARING RESULTS FROM THE THREE METHODS

The complete set of tokens generated from pattern-based method, rule-based method and hierarchical clustering method are shown in Appendix 1, Appendix 2 and Appendix 3. Some of the groups were generated from more than one approach. Table 7.8 shows the 20 patterns found by all three clustering methods.

Table 7.8: Common tokens obtained from all three methods

| Token # | Token |
|---|---|
| 1 | ACMM |
| 2 | ACMR |
| 3 | ACRG |
| 4 | ACRM |
| 5 | ACRMM |
| 6 | ACRRM |
| 7 | CMCM |
| 8 | CMMM |
| 9 | CRCR |
| 10 | CRMM |
| 11 | CRMR |
| 12 | CRRC |
| 13 | FMRM |
| 14 | MMMM |
| 15 | MMMMM |
| 16 | MMRM |
| 17 | MRCR |
| 18 | RCRR |
| 19 | TTTT |
| 20 | WMMM |

# 8. RESULTS AND DISCUSSION

The strategies developed for automatic clustering in this thesis work were applied on a real data set obtained from a research group's use of VKB. To better assess the three clustering approaches, the same data set is manually grouped by two subjects using VKB. From the outputs obtained from the two subjects a single representative output was generated. The output of this manual clustering of VKB events is shown in Table 8.1.

Table 8.1: List of groups obtained from manual grouping

| Time stamp | Description of objects on which events were applied | Token |
|---|---|---|
| Wed Nov. 03 15:44:31 CST 1999 | Evaluation | ACRM |
| Same as above | System feature | ACRCR |
| Same as above | History undo | ACMGB |
| Same as above | Visual Mapping | ACW |
| Same as above | Done | ARCRRGB |
| Same as above | WWW 2000 | ACGBRM |
| Same as above | DIS 2000 | ACRMRMRCR |
| Same as above | WWW 2000 | MRCRM |
| Same as above | NRHM Journal | ACRRM |
| Same as above | tempest | ACRMMMMMMMM |
| Wed Nov. 10 15:48:44 CST 1999 | Necessary Feature | ACRGB |
| Same as above | Types for objects | ACMRMM |
| Same as above | Bringing up the web | CRRCRCR |
| Wed Nov 17 16:02:07 CST 1999 | WWW 2000 | AGBCM |
| Same as above | Object 160 | WWMGBGBWWR |
| Same as above | Collection 1 | NX |
| Wed Dec 01 15:48:50 CST 1999 | Object 248 | ARMMGB |
| Wed Dec 08 15:31:40 CST 1999 | Object 248 | BBBBBBBBBBBB |
| Wed Jan 26 17:20:09 CST 2000 | Object 248 | MRCRTFlRM |
| Wed Feb 02 17:43:11 CST 2000 | XML Output | GBGBGBGMG |
| Same as above | HT 2001 | ACCR |

Table 8.1 Continued

| Time stamp | Description of objects on which events were applied | Token |
|---|---|---|
| Same as above | Visual Knowledge builder | ACRFlMRTCM |
| Wed Feb 09 17:45:45 CST 2000 | Type Management | AXCGB |
| Same as above | Apply to existing | ACR |
| Same as above | Propagating visual changes | GBCRCRR |
| Same as above | Visual mapping | ACRT |
| Wed Mar 22 17:29:14 CST 2000 | Smart Exploding | MRCR |
| Same as above | Linking Navigation | GBGBWBBGBGB |
| Wed Mar 29 17:33:26 CST 2000 | Linking to history | CMMM |
| Wed Apr 26 17:39:04 CST 2000 | ADL | MRM |
| Wed May 17 18:06:26 CST 2000 | Session level | ACRMMGBW |
| Wed Jul 12 17:44:15 CST 2000 | Misc. Objects | MMMMMMMMMMMMMMM |
| Wed Aug 02 18:15:22 CST 2000 | Suggestion | ACGBRMTTTTT |
| Wed Aug 09 18:23:39CST 2000 | System features | MMMM |
| Wed Aug 23 18:12:01 CST 2000 | Palette | ACGC |
| Thu Aug 31 18:23:27 CST 2000 | Link manager | ACCGBWRM |
| Same as above | Palette editing | GWBM |
| Tue Oct 17 17:37:50 CST 2000 | Different objects | MMMMMMMMM |
| Tue Nov 07 17:44:41 CST 2000 | Object 2130 | CMM |
| Tue Nov 16 17:36:38 CST 2000 | Object 2105 | CMR |
| Tue Nov 21 17:50:42 CST 2000 | Object 2261 | ACMCM |
| Tue Nov 28 17:48:59 CST 2000 | Collection 2340 | MCMM |
| Tue Jan 23 17:38:44 CST 2001 | DL 2001 | CMCM |
| Tue Mar 06 17:59:19 CST 2001 | Collection 2340 | RMMMM |
| Tue Mar 08 20:20:54 CST 2001 | Collection 2437 | MMR |
| Same as above | Object 2339 | MGBMRCMMM |
| Same as above | XML Output | MMRCRCR |
| Same as above | Collection 2403 | MBGM |
| Tue May 01 18:01:58 CDT 2001 | Object 788 | CCCCCCCC |
| Same as above | Object 2339 | TTTT |
| Tue Jul 17 15:53:48 CDT 2001 | Old Task that might | ACGBMR |
| Same as above | Encrypted Password | ACRCGBM |
| Tue Aug 28 15:41:28 CDT 2001 | User id in history | ACRWBMRM |
| Same as above | Overview Window | ACRBGM |
| Wed Sep 05 16:07:35 CDT 2001 | Grouping history | ACRRMMRW |
| Thu Feb 14 16:52:34 CST 2002 | Link data base | ACMRRMM |
| Wed Jun 19 16:36:22 CDT 2002 | XML data file format | ACMMRM |
| Thu Feb 13 16:54:54 CST 2003 | Magnetic poetry study | MCRMRGBW |

Table 8.1 Continued

| Time stamp | Description of objects on which events were applied | Token |
|---|---|---|
| Wed Sep 10 15:16:34 CDT 2003 | Attribute analysis in type.. | ACRCM |
| Same as above | Relation suggestion | CRCR |
| Same as above | Logarithmic distribution | MCRR |
| Same as above | Encrypted password | CRMM |
| Wed Oct 22 16:26:09 CDT 2003 | Do we base it | ACRRWGB |
| Same as above | Symbol Scenarios | ACRCRRCR |
| Same as above | VITE journal paper | FlRRFlR |
| Same as above | VITE journal paper | RFlMRM |

The edit sequences obtained from manual grouping were compared with those identified by the three automatic methods. There were a total of 70 manual groups. Of these, 8 groups were detected by all three strategies. Another 8 groups were detected by 2 strategies. 24 groups were detected by only one strategy and 30 groups were not detected by any strategy at all.

Out of the 70 manual groups generated, the pattern-based approach detected 14 groups, the rule-based approach detected 28 groups, and the HAC approach detected 22 groups. Table 8.2 lists all the manual groups and which group was detected by which strategies.

Table 8.2: Detection of manual groups by different strategies

| Token | Detected by Pattern Based? | Detected by Rule Based? | Detected by Hierarchical Clustering? | Total detection By different strategies |
|---|---|---|---|---|
| ACRRM | YES | YES | YES | 3 |
| MRCR | YES | YES | YES | 3 |
| CMMM | YES | YES | YES | 3 |
| MMMM | YES | YES | YES | 3 |
| CMCM | YES | YES | YES | 3 |

Table 8.2 Continued

| Token | Detected by Pattern Based? | Detected by Rule Based? | Detected by Hierarchical Clustering? | Total detection By different strategies |
|---|---|---|---|---|
| TTTT | YES | YES | YES | 3 |
| CRCR | YES | YES | YES | 3 |
| CRMM | YES | YES | YES | 3 |
| ACRCR | YES | YES | NO | 2 |
| ACGBRM | NO | YES | YES | 2 |
| MMMMMM | YES | NO | YES | 2 |
| GWBM | NO | YES | YES | 2 |
| MMMMMMMMM | YES | NO | YES | 2 |
| MCMM | YES | NO | YES | 2 |
| RMMMM | YES | YES | NO | 2 |
| ACRBGM | NO | YES | YES | 2 |
| ACW | NO | NO | YES | 1 |
| MRCRM | NO | YES | NO | 1 |
| ACRMMMMMMMM | NO | YES | NO | 1 |
| ACMRMM | YES | NO | NO | 1 |
| CRRCRCR | NO | YES | NO | 1 |
| AAAAAA | NO | NO | YES | 1 |
| ACR | NO | NO | YES | 1 |
| ACRT | NO | YES | NO | 1 |
| MRM | NO | NO | YES | 1 |
| MMMMMMMMMMMMMMM | NO | NO | YES | 1 |
| ACGC | NO | YES | NO | 1 |
| CMM | NO | NO | YES | 1 |
| CMR | NO | NO | YES | 1 |
| MMR | NO | NO | YES | 1 |
| MMRCRCR | NO | YES | NO | 1 |
| MBGM | NO | YES | NO | 1 |
| CCCCCCCC | NO | YES | NO | 1 |
| ACRRMMRW | NO | YES | NO | 1 |
| ACMRRMM | NO | YES | NO | 1 |
| ACMMRM | NO | YES | NO | 1 |
| ACRCM | NO | YES | NO | 1 |
| MCRR | NO | YES | NO | 1 |
| ACRCRRCR | NO | YES | NO | 1 |
| FlRRFlR | NO | YES | NO | 1 |
| ACMGB | NO | NO | NO | 0 |

Table 8.2 Continued

| Token | Detected by Pattern Based? | Detected by Rule Based? | Detected by Hierarchical Clustering? | Total detection By different strategies |
|---|---|---|---|---|
| ARCRRGB | NO | NO | NO | 0 |
| ACRMRMRCR | NO | NO | NO | 0 |
| ACRGB | NO | NO | NO | 0 |
| GBGBGBGB | NO | NO | NO | 0 |
| AGBCM | NO | NO | NO | 0 |
| GGGGGGBBBBBB | NO | NO | NO | 0 |
| GBGBGBGBGBGB | NO | NO | NO | 0 |
| WWMGBGBWWR | NO | NO | NO | 0 |
| NX | NO | NO | NO | 0 |
| ARMMGB | NO | NO | NO | 0 |
| BBBBBBBBBBBB | NO | NO | NO | 0 |
| MRCRTFlRM | NO | NO | NO | 0 |
| GBGBGBGMG | NO | NO | NO | 0 |
| ACCR | NO | NO | NO | 0 |
| ACRFlMRTCM | NO | NO | NO | 0 |
| AXCGB | NO | NO | NO | 0 |
| GBCRCRR | NO | NO | NO | 0 |
| GBGBWBBGBGB | NO | NO | NO | 0 |
| ACRMMGBW | NO | NO | NO | 0 |
| ACGBRMTTTTT | NO | NO | NO | 0 |
| ACCGBWRM | NO | NO | NO | 0 |
| ACMCM | NO | NO | NO | 0 |
| MGBMRCMMM | NO | NO | NO | 0 |
| ACGBMR | NO | NO | NO | 0 |
| ACRCGBM | NO | NO | NO | 0 |
| ACRWBMRM | NO | NO | NO | 0 |
| MCRMRGBW | NO | NO | NO | 0 |
| ACRRWGB | NO | NO | NO | 0 |
| RFlMRM | NO | NO | NO | 0 |

As a final comparison, Table 8.3 shows the number of groups generated by each of the three methods and how many of the manually-constructed groups were identified by each approach. The rule-based method not only detected the highest number of manual

groups but it also generated the fewest groups. This could indicate that it generates the fewest number of spurious groups, having the highest precision and recall of the three methods.

Table 8.3: Statistics of detected groups by automatic strategies

| Strategies | Number of groups |
|---|---|
| Event groups detected by Pattern-based method | 403 |
| Event groups detected by Rule-based method | 130 |
| Event groups detected by HAC based method | 613 |
| | |
| Manual groups detected by Pattern-based method | 14 |
| Manual groups detected by Rule-based method | 28 |
| Manual groups detected by HAC based method | 22 |

While this indicates the potential value of a few simple rules for clustering edit history, the advantages of the other approaches should also be acknowledged. The pattern-based approach does more than just grouping edit events, it uncovers repeated patterns of actions that could be used to generate macros to better support VKB users. This would require tuning the algorithm to be more selective and would benefit from the inclusion of wildcards and regular expression matching to the general approach. The HAC approach similarly needs to be tuned to be much more selective in grouping edit events. Of all the approaches, it is the one that could potentially recognize a single activity involving structures in multiple collections of a VKB workspace. Finally, the success of the rule-based method indicates that the addition of rules for additional levels of grouping based on the spatial parser should be investigated.

# 9. CONCLUSION

History mechanisms available in hypertext systems allow access to past user interactions with the system. This availability allows users to evaluate past work and to learn from past activity. It also allows systems to identify usage patterns and potentially predict behaviors with the system. Thus, recording history is useful to both the system and the user.

The Visual Knowledge Builder (VKB) is a *spatial hypertext* – a visual workspace for collecting, organizing, and sharing information. It stores a detailed history of all user edit interactions during previous authoring sessions. Various tools and techniques have been developed to group and annotate VKB history. Such grouping may be valuable for a number of applications: accessing desired segments of long edit histories, identifying portions of edit history with special or similar attributes, and predicting future interaction during VKB authoring sessions. But the problem with these tools is that the grouping operation or annotation is currently performed manually. For a large VKB history that has been growing over a long period of time, creating groups using such tools is difficult and time consuming. This thesis examines methods to analyze VKB history in order to create groups/clusters from history events automatically.

Three approaches to automatic grouping are compared in this thesis. The first approach is a pattern-matching approach that identifies repeated patterns of edit events in the history. The second approach is a rule-based approach that uses simple rules, such as “group all consecutive events on a single object”. The third approach uses hierarchical agglomerative clustering (HAC) where edits are grouped based on a function of edit time and edit location. Each approach has its own theoretical strengths and weaknesses. By implementing and evaluating the three approaches, this thesis examines their practical strengths and weaknesses as well.

The results showed that the pattern-matching approach generated many potential groupings but few matched those that were generated by people. Tuning of the algorithm to be much more selective, while also extending it to include wildcards and regular expressions, would likely be required before this approach would be usable.

The rule-based approach performed best in that it best matched human-defined groups and generated the fewest number of groups. Extensions to this approach include additional levels of grouping based on the spatial structures recognized by a spatial parser and the tuning of the rules to include time, position, and event type considerations.

The hierarchic agglomerative clustering approach was in between the other two approaches with regards to identifying human-defined groups. This algorithm relies on a similarity measure that has the potential to be much better defined based on the results of this thesis. The combination of such tuning with better rules for selecting clustering levels could rival or surpass the effectiveness of the rule-based approach. This is a case where the simplest approach and implementation performed the best. This could be because the rules best match what people do when they are forced to group edit events or it could be because the complexities of the other methods require significant tuning to generate reasonably good results.

# REFERENCES

AGGARWAL, C. AND YU, P. 1998. Data mining techniques for associations, clustering and classification. In *Proceedings of the Third Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining (*Beijing, China). 13-23.

AKKAPEDDI, R. 2003. *Grouping annotating and filtering history information in VKB*. MS Thesis, Dept. of Computer Science, Texas A&M University.

BUSH, V. 1945. As we may think. *The Atlantic Monthly*, August 1945, 101-108.

CAPELLE, M., MASSON, C. AND BOULICAUT, J. 2002. Mining frequent sequential patterns under a similarity constraint. In *Proceedings of the Third International Conference on Intelligent Data Engineering and Automated Learning (IDEAL), 2412*, (Manchester, UK). 1-6.

CHIU, P. AND WILCOX, L. 1998. A dynamic grouping technique for ink and audio notes. In *Proceedings of the ACM Symposium on User Interface Software and Technology ( UIST '98)*, (San Francisco, CA). ACM, New York, 195-202.

COCKBURN, A. AND JONES, S. 1996. Which way now? Analyzing and easing inadequacies in WWW navigation. *International Journal of Human-Computer Studies, 45, 105-129*.

CONKLIN, J. 1987. Hypertext: An introduction and survey. *IEEE Transactions on Computers, 20*, 9, 17- 41.

CONKLIN, J. AND BEGEMAN, M.L. 1988. gIBIS: A hypertext tool for exploratory policy discussion. *ACM Transactions on Information Systems, 6*, 4, 303-331.

DISESSA, A. AND ABELSON. H. 1986. Boxer: A re-constructible computational medium. *Communications of the ACM, 29*, 9, 859-868.

EDMONDS, E., MORAN, T.P. AND DO, E. 1988. Interactive systems for supporting emergence of concepts and ideas. *SIGCHI Bulletin, 30*, 1, 24-25.

EVERITT, B. 1981. *Cluster Analysis.* Heinemann Educational Books, New York, NY.

FEINER, R. 1988. Seeing the forest for the trees: Hierarchical display of hypertext structures. In *Proceedings of the ACM Conference on Office Information Systems* (Palo Alto). ACM, New York, 205-212.

GAROFALAKIS, M., RASTOGI, R. AND SHIM, K. 1999. SPIRIT: Sequential pattern mining with regular expression constraints. In *Proceedings of 25th International Conference on Very Large Data Bases* (Edinburgh, Scotland). Morgan Kaufmann Pub. Inc., San Francisco, CA, 223-234.

GATICA-PEREZ, D., SUN, M-T. AND LOUI, A. 2001. Consumer video structuring by probabilistic merging of video segments. In *Proceedings of the IEEE International Conference on Multimedia (ICME),* (Tokyo, Japan). IEEE Computer Society Press, Los Alamitos, 916-919.

GREENBERG, S. 1993a. Supporting command reuse: Mechanisms for reuse. *International Journal of Man-Machine Studies*, *39*, 391-425.

GREENBERG, S. 1993b. *The computer user as toolsmith: The use, reuse, and organization of computer-based tools.* Cambridge Series on Human-Computer Interaction. Cambridge University Press, New York.

GREENBERG, S. AND WITTEN, I. 1988. Directing the user interface: How people use command-based systems. In *Proceedings of the 3rd IFAC Conference on Man-Machine Systems (Oulu, Finland*). 14 -16.

GUHA, R. 2005. Meta Content Framework. Accessed on June 02, 2005 at: http://www.xspace.net/hotsauce/mcf.html.

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

HALASZ, F., MORAN, T. AND TRIGG R. 1987. Notecards in a nutshell. In *Proceedings of the CHI and GI '87 Conference on Human Factors in Computing Systems* (Toronto, Canada). ACM, New York, 45-52.

HAN, H., MANAVOGLU, E., GILES, C. AND ZHA, H. 2003. Rule-based word clustering for text classification. In *Proceedings of the 26th Annual International ACM Conference on Research and Development in Information Retrieval* (Toronto, Canada). ACM, New York, 445-446.

HANSON, S., KRAUT, R. AND FARBER, J. 1984. Interface design and multivariate analysis of Unix command use. *ACM Transactions on Office Information Systems, 2*, 1, 42-57.

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

JOYCE, M. 1991. Storyspace as a hypertext system for writers and readers of varying ability. In *Proceedings of Hypertext 91* (San Antonio, Texas). ACM, New York, 381-389.

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

KUM, H., PAULSEN, S.AND WANG, W. 2002. Comparative study of sequential pattern mining frameworks: Support framework vs. multiple alignment framework. In *Proceedings of the 2002 IEEE ICDM Workshop on The Foundation of Data Mining and Discovery* (Maebashi, Japan). 65-68.

LEE, A. 1992. *Investigations into history tools for user support.* Ph.D. Thesis, Computer Systems Research Institute, University of Toronto, Canada.

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

MARSHALL, C. AND SHIPMAN, F. 1997. Effects of hypertext technology on the practice of information triage. In *Proceedings of ACM Hypertext `97* (Southampton, UK). ACM, New York, 167-176.

MARSHALL, C., HALASZ, F., ROGERS, R. AND JANSSEN, W. 1991. Aquanet: A hypertext tool to hold your knowledge in place. In *Proceedings of Hypertext '91* (San

Antonio, Texas). ACM, New York, 261 – 275.

MARSHALL, C., SHIPMAN, F. AND COOMBS, J. 1994. VIKI: Spatial hypertext supporting emergent structure. *ECHT '94: European Conference on Hypertext Technology* (Edinburgh, Scotland). ACM, New York, 13-23.

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

RUI, Y. AND HUANG, T. 2000. A unified framework for video browsing and retrieval. In *Image and Video Processing Handbook (A. Bovik, Ed.)*. Academic Press, New York, 705–715.

SHIPMAN, F., CHANEY R. AND GORRY, G. 1988. Distributed hypertext for collaborative research: The virtual notebook system. In *Proceedings of the ACM Hypertext '89 (*Pittsburgh, Pennsylvania). ACM, New York, 129-135.

SHIPMAN, F., HSIEH, H., AIRHART, R., MALOOR, P. AND MOORE, M. 2001. The visual knowledge builder: A second generation spatial hypertext. In *Proceeding of the ACM Conference on Hypertext* (Aarhus, Denmark). ACM, New York, 113-122.

SIMPSON, R. 2001. Experiences with Web Squirrel: My Life on the Information Farm. In *Proceeding of the ACM Conference on Hypertext* (Aarhus, Denmark). ACM, New York, 127-128.

SMITH, J., WEISS, S. AND FERGUSON, G. 1987. A hypertext writing environment and its cognitive basis. In *Proceedings of the ACM Conference on Hypertext* (Chapel Hill, North Carolina). ACM, New York, 195 – 214.

STATSOFT. 2005. Data Mining Techniques. Available online at: http://www.statsoft.com/textbook/stdatmin.html.

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

TRIGG, R. AND WEISER, M. 1986. TEXTNET: A network based approach to text handling. *ACM Transactions on Information Systems, 4*, 1, 1-23.

WALKER, J. 1987. Document Examiner: Delivery interface for hypertext documents. In *Proceedings of a Conference held at the University of North Carolina* (Chapel Hill, North Carolina). 307-323.

WEXELBLAT, A. 1998. History-rich tools for social navigation. In *Proceedings of the Computer Human Interaction Conference (CHI '98),* (Los Angeles, California). ACM, New York, 359-360.

YEUNG, M., YEO, B. AND LIU, B. 1998. Segmentation of video by clustering and graph analysis. *Computer Vision and Image Understanding, 71*, 1, 94–109.

ZHAO, L., QI, W., WANG, Y., YANG, S. AND ZHANG, H. 2001. Video shot grouping using best-first model merging. In *Proceeding of the Storage and Retrieval for Media Databases*, 4315, (San Carlos, Mexico). 262–269.

ZHONG, D. AND ZHANG, H. 1996. Clustering methods for video browsing and annotation. In *Proceeding of the Storage and Retrieval for Still Images and Video Databases IV*, 2670, (San Jose, CA). 239–246.

# APPENDIX 1

## TOKENS GENERATED FROM PATTERN BASED ANALYSIS

Table A-1: Token generated from pattern based method

| Token # | Token generated |
|---|---|
| 1 | AAAAG |
| 2 | AAAG |
| 3 | AAAGAA |
| 4 | AAGAA |
| 5 | AAGGA |
| 6 | ACAC |
| 7 | ACACAC |
| 8 | ACACC |
| 9 | ACACRRMM |
| 10 | ACCAC |
| 11 | ACCC |
| 12 | ACCM |
| 13 | ACGM |
| 14 | ACMAC |
| 15 | ACMACM |
| 16 | ACMC |
| 17 | ACMM |
| 18 | ACMMM |
| 19 | ACMMMM |
| 20 | ACMMMMM |
| 21 | ACMMMMMM |
| 22 | ACMMMMMMM |
| 23 | ACMMMMMMMM |
| 24 | ACMR |
| 25 | ACMRM |
| 26 | ACMRMM |
| 27 | ACRA |
| 28 | ACRAC |
| 29 | ACRACR |
| 30 | ACRACRACRM |

Table A-1 Continued

| Token # | Token generated |
|---|---|
| 31 | ACRCR |
| 32 | ACRCRR |
| 33 | ACRG |
| 34 | ACRGACR |
| 35 | ACRM |
| 36 | ACRMAC |
| 37 | ACRMACR |
| 38 | ACRMACRM |
| 39 | ACRMM |
| 40 | ACRMMM |
| 41 | ACRMMMM |
| 42 | ACRR |
| 43 | ACRRM |
| 44 | ACRRMACR |
| 45 | ACRW |
| 46 | ACWA |
| 47 | ACWG |
| 48 | AGAA |
| 49 | AGAAAG |
| 50 | ARCRAC |
| 51 | AUUUUUUUUUUUU |
| 52 | BACW |
| 53 | BBBB |
| 54 | BBBWBB |
| 55 | BBGGG |
| 56 | BBWW |
| 57 | BBWWGGBB |
| 58 | BMBM |
| 59 | BMMMMM |
| 60 | BWWGG |
| 61 | BWWGGB |
| 62 | CACC |
| 63 | CACMM |
| 64 | CACR |
| 65 | CCACC |
| 66 | CCCCCCCCM |
| 67 | CCCCM |
| 68 | CCCCMC |

Table A-1 Continued

| Token # | Token generated |
|---|---|
| 69 | CCMCMCM |
| 70 | CCMCMCMCM |
| 71 | CCMM |
| 72 | CCRMM |
| 73 | CGACR |
| 74 | CMAC |
| 75 | CMCC |
| 76 | CMCM |
| 77 | CMCMC |
| 78 | CMCMCM |
| 79 | CMCMCMCM |
| 80 | CMMAC |
| 81 | CMMACM |
| 82 | CMMACMAC |
| 83 | CMMCM |
| 84 | CMMCMMAC |
| 85 | CMMM |
| 86 | CMMMM |
| 87 | CMMMMM |
| 88 | CMMMMMAC |
| 89 | CMMMMMM |
| 90 | CMMMMMMM |
| 91 | CMMMMMMMM |
| 92 | CMMR |
| 93 | CMWC |
| 94 | CRAC |
| 95 | CRACGBM |
| 96 | CRACR |
| 97 | CRACRACR |
| 98 | CRCR |
| 99 | CRMM |
| 100 | CRMMM |
| 101 | CRMMMAC |
| 102 | CRMMMM |
| 103 | CRMMMMMMMM |
| 104 | CRMR |
| 105 | CRRAC |
| 106 | CRRC |

Table A-1 Continued

| Token # | Token generated |
|---|---|
| 107 | CRRM |
| 108 | CRRMM |
| 109 | FMRM |
| 110 | FRMMM |
| 111 | FlACR |
| 112 | FlRM |
| 113 | GAAA |
| 114 | GAAAG |
| 115 | GAAAGGA |
| 116 | GBBWGG |
| 117 | GBMGG |
| 118 | GBWWGG |
| 119 | GBWWGGAAA |
| 120 | GCMM |
| 121 | GGAAA |
| 122 | GGAAAG |
| 123 | GGBB |
| 124 | GGGG |
| 125 | GGGGG |
| 126 | GGGM |
| 127 | GGMM |
| 128 | GMAC |
| 129 | GMMA |
| 130 | GMMAC |
| 131 | GMMACR |
| 132 | GMMM |
| 133 | GMMMACR |
| 134 | GMMMMMM |
| 135 | GRMAC |
| 136 | MACAC |
| 137 | MACACM |
| 138 | MACCMCMCM |
| 139 | MACG |
| 140 | MACM |
| 141 | MACMM |
| 142 | MACR |
| 143 | MACRG |
| 144 | MACRM |

Table A-1 Continued

| Token # | Token generated |
|---|---|
| 145 | MACRMM |
| 146 | MACRMMG |
| 147 | MACRMMM |
| 148 | MACRMMMM |
| 149 | MACRMMMMMM |
| 150 | MARC |
| 151 | MARCR |
| 152 | MBWM |
| 153 | MCAC |
| 154 | MCCCCCCCC |
| 155 | MCCCCCCCCM |
| 156 | MCCM |
| 157 | MCCMR |
| 158 | MCCR |
| 159 | MCMC |
| 160 | MCMCCM |
| 161 | MCMCM |
| 162 | MCMM |
| 163 | MCMMCM |
| 164 | MCMMMMM |
| 165 | MCRM |
| 166 | MCRMM |
| 167 | MGAC |
| 168 | MGMM |
| 169 | MGMMM |
| 170 | MMAC |
| 171 | MMACM |
| 172 | MMACMAC |
| 173 | MMACMMMMM |
| 174 | MMACMMMMMM |
| 175 | MMACMR |
| 176 | MMACR |
| 177 | MMACRC |
| 178 | MMACRM |
| 179 | MMACRMM |
| 180 | MMACRR |
| 181 | MMAR |
| 182 | MMARC |

Table A-1 Continued

| Token # | Token generated |
|---|---|
| 183 | MMBW |
| 184 | MMCM |
| 185 | MMCMC |
| 186 | MMCMM |
| 187 | MMCMMMMMM |
| 188 | MMCR |
| 189 | MMCRR |
| 190 | MMFR |
| 191 | MMGG |
| 192 | MMGW |
| 193 | MMMA |
| 194 | MMMAC |
| 195 | MMMACM |
| 196 | MMMACR |
| 197 | MMMACRM |
| 198 | MMMACRMMM |
| 199 | MMMACRMMMM |
| 200 | MMMACRMMMMMMM |
| 201 | MMMAR |
| 202 | MMMC |
| 203 | MMMCMM |
| 204 | MMMCR |
| 205 | MMMF |
| 206 | MMMG |
| 207 | MMMM |
| 208 | MMMMA |
| 209 | MMMMAC |
| 210 | MMMMACAC |
| 211 | MMMMACM |
| 212 | MMMMACMMMMMM |
| 213 | MMMMACRM |
| 214 | MMMMACRMACR |
| 215 | MMMMACRMMM |
| 216 | MMMMACRMMMM |
| 217 | MMMMC |
| 218 | MMMMCMM |
| 219 | MMMMCMMM |
| 220 | MMMMCR |

Table A-1 Continued

| Token # | Token generated |
|---|---|
| 221 | MMMMCRM |
| 222 | MMMMM |
| 223 | MMMMMA |
| 224 | MMMMMAC |
| 225 | MMMMMACAC |
| 226 | MMMMMACM |
| 227 | MMMMMACMC |
| 228 | MMMMMACR |
| 229 | MMMMMACRRG |
| 230 | MMMMMC |
| 231 | MMMMMCMM |
| 232 | MMMMMCMMMM |
| 233 | MMMMMFRMM |
| 234 | MMMMMM |
| 235 | MMMMMMA |
| 236 | MMMMMMAC |
| 237 | MMMMMMACR |
| 238 | MMMMMMC |
| 239 | MMMMMMM |
| 240 | MMMMMMMA |
| 241 | MMMMMMMAC |
| 242 | MMMMMMMACM |
| 243 | MMMMMMMACR |
| 244 | MMMMMMMACRMMM |
| 245 | MMMMMMMC |
| 246 | MMMMMMMMA |
| 247 | MMMMMMMMAC |
| 248 | MMMMMMMMM |
| 249 | MMMMMMMMMMF |
| 250 | MMMMMMMMMMMMMMMCMM |
| 251 | MMMMMMMMMMMMMMMMCRM |
| 252 | MMMMMMMMMMRMM |
| 253 | MMMMMMMMMR |
| 254 | MMMMMMMMMW |
| 255 | MMMMMMMMR |
| 256 | MMMMMMMMRM |
| 257 | MMMMMMMR |
| 258 | MMMMMMMRM |

Table A-1 Continued

| Token # | Token generated |
|---|---|
| 259 | MMMMMMMRMM |
| 260 | MMMMMMR |
| 261 | MMMMMMW |
| 262 | MMMMMR |
| 263 | MMMMMRM |
| 264 | MMMMMRMM |
| 265 | MMMMMW |
| 266 | MMMMMWWW |
| 267 | MMMMN |
| 268 | MMMMR |
| 269 | MMMMRM |
| 270 | MMMMRMM |
| 271 | MMMMRMMM |
| 272 | MMMMRMMMMMMM |
| 273 | MMMMRMMMMMMMM |
| 274 | MMMMW |
| 275 | MMMMWMM |
| 276 | MMMMWW |
| 277 | MMMN |
| 278 | MMMNMMM |
| 279 | MMMR |
| 280 | MMMRF |
| 281 | MMMRM |
| 282 | MMMRMM |
| 283 | MMMRMMM |
| 284 | MMMRMMMM |
| 285 | MMMRMR |
| 286 | MMMRRM |
| 287 | MMMW |
| 288 | MMMWM |
| 289 | MMMWMM |
| 290 | MMMWMMM |
| 291 | MMMWW |
| 292 | MMNM |
| 293 | MMNMM |
| 294 | MMRM |
| 295 | MMRMC |
| 296 | MMRMM |

Table A-1 Continued

| Token # | Token generated |
|---|---|
| 297 | MMRMMM |
| 298 | MMRMMMM |
| 299 | MMRMMMMMMM |
| 300 | MMRMMMMRM |
| 301 | MMRMMMMRMM |
| 302 | MMRMMRM |
| 303 | MMRMRM |
| 304 | MMRR |
| 305 | MMWM |
| 306 | MMWMM |
| 307 | MMWMMMMMMMMMMM |
| 308 | MMWMMWM |
| 309 | MMWW |
| 310 | MMWWB |
| 311 | MMWWW |
| 312 | MRAC |
| 313 | MRACM |
| 314 | MRCM |
| 315 | MRCR |
| 316 | MRCRMMMMMM |
| 317 | MRCRMMR |
| 318 | MRMAC |
| 319 | MRMACR |
| 320 | MRMC |
| 321 | MRMM |
| 322 | MRMMM |
| 323 | MRMMMM |
| 324 | MRMMMMM |
| 325 | MRMMMMMM |
| 326 | MRMMMMMMM |
| 327 | MRMMMMRM |
| 328 | MRMMRM |
| 329 | MRMR |
| 330 | MRMRM |
| 331 | MRRM |
| 332 | MRRRR |
| 333 | MWAC |
| 334 | MWMMMMM |

Table A-1 Continued

| Token # | Token generated |
|---|---|
| 335 | MWWMMM |
| 336 | RACCCC |
| 337 | RACGM |
| 338 | RACMMM |
| 339 | RACR |
| 340 | RACRR |
| 341 | RCRA |
| 342 | RCRR |
| 343 | RMAC |
| 344 | RMACR |
| 345 | RMACRM |
| 346 | RMFl |
| 347 | RMMM |
| 348 | RMMMACRM |
| 349 | RMMMC |
| 350 | RMMMM |
| 351 | RMMMMA |
| 352 | RMMMMM |
| 353 | RMMMMMM |
| 354 | RMMMMMMM |
| 355 | RMMMMMMMM |
| 356 | RMMMMMMMMM |
| 357 | RMMMMMMMMMM |
| 358 | RMMMMR |
| 359 | RMMMMRM |
| 360 | RMMMMRMM |
| 361 | RMMMMRMMM |
| 362 | RMMR |
| 363 | RMMRMM |
| 364 | RMMWW |
| 365 | RMRF |
| 366 | RMRM |
| 367 | RMRMMMM |
| 368 | RMRMR |
| 369 | RMRR |
| 370 | RRMA |
| 371 | RRMM |
| 372 | RRMMMM |

Table A-1 Continued

| Token # | Token generated |
|---|---|
| 373 | RRRR |
| 374 | RRRRM |
| 375 | RRRRR |
| 376 | TTTT |
| 377 | TTTTT |
| 378 | TTTTTTT |
| 379 | WACR |
| 380 | WACRMM |
| 381 | WBMM |
| 382 | WGGB |
| 383 | WGGBB |
| 384 | WGGBBWW |
| 385 | WMAC |
| 386 | WMMM |
| 387 | WMMMM |
| 388 | WMMMMM |
| 389 | WMMMMMM |
| 390 | WMMMMMMMMMM |
| 391 | WMMMRMAC |
| 392 | WMMW |
| 393 | WWBM |
| 394 | WWGG |
| 395 | WWGGB |
| 396 | WWGGBB |
| 397 | WWGGBBWWW |
| 398 | WWMM |
| 399 | WWMMMM |
| 400 | WWWG |
| 401 | WWWGG |
| 402 | WWWMM |
| 403 | pppp |

# APPENDIX 2

## GROUPS GENERATED FROM RULE BASED ANALYSIS

Table A-2: Group generated from rule based method

| Group # | Group generated |
|---|---|
| 1 | ACBWRM |
| 2 | ACCMMRR |
| 3 | ACGBRM |
| 4 | ACGC |
| 5 | ACGGRM |
| 6 | ACGM |
| 7 | ACGMR |
| 8 | ACGMX |
| 9 | ACGR |
| 10 | ACGRM |
| 11 | ACGWB |
| 12 | ACGWWGG |
| 13 | ACMC |
| 14 | ACMCC |
| 15 | ACMCRCRR |
| 16 | ACMG |
| 17 | ACMM |
| 18 | ACMMRM |
| 19 | ACMR |
| 20 | ACMRM |
| 21 | ACMRRMM |
| 22 | ACRBGM |
| 23 | ACRC |
| 24 | ACRCGM |
| 25 | ACRCM |
| 26 | ACRCR |
| 27 | ACRCRRCR |
| 28 | ACRCRRM |
| 29 | ACRCRRMM |
| 30 | ACRFMRTCM |

Table A-2 Continued

| Group # | Group generated |
|---|---|
| 31 | ACRG |
| 32 | ACRGG |
| 33 | ACRGW |
| 34 | ACRM |
| 35 | ACRMFlM |
| 36 | ACRMG |
| 37 | ACRMGTMM |
| 38 | ACRMM |
| 39 | ACRMMBMWR |
| 40 | ACRMMCRCCRM |
| 41 | ACRMMGW |
| 42 | ACRMMMMMMMM |
| 43 | ACRMRCRM |
| 44 | ACRMW |
| 45 | ACRRG |
| 46 | ACRRM |
| 47 | ACRRMG |
| 48 | ACRRMMRW |
| 49 | ACRRWBG |
| 50 | ACRT |
| 51 | ACRW |
| 52 | ACRWBMM |
| 53 | ACRWM |
| 54 | ACRWR |
| 55 | ACWBRM |
| 56 | ACWGM |
| 57 | ACWR |
| 58 | AGCM |
| 59 | AGMM |
| 60 | ARBC |
| 61 | ARCMC |
| 62 | ARCR |
| 63 | ARCRGRMMM |
| 64 | ARCRR |
| 65 | ARCRRGGMFR |
| 66 | ARGCRM |
| 67 | ARGRCRMM |
| 68 | ARMC |

Table A-2 Continued

| Group # | Group generated |
|---|---|
| 69 | ARMMG |
| 70 | AXCG |
| 71 | BWRBG |
| 72 | CCCC |
| 73 | CCCCCCCC |
| 74 | CCCCCCCCC |
| 75 | CCMC |
| 76 | CMCM |
| 77 | CMMM |
| 78 | CRCR |
| 79 | CRMM |
| 80 | CRMR |
| 81 | CRMRCMC |
| 82 | CRRC |
| 83 | CRRCRCR |
| 84 | FFFFlFFl |
| 85 | FMRM |
| 86 | FlRRFR |
| 87 | FlRRFlR |
| 88 | GBBGW |
| 89 | GBCR |
| 90 | GBGGBG |
| 91 | GBGWMB |
| 92 | GCRCRR |
| 93 | GGGGGG |
| 94 | GGWBBBG |
| 95 | GWBM |
| 96 | MBGM |
| 97 | MBWC |
| 98 | MBWM |
| 99 | MCRMRGW |
| 100 | MCRR |
| 101 | MFFl |
| 102 | MGBMRC |
| 103 | MGFlRC |
| 104 | MMFR |
| 105 | MMFRMMMF |
| 106 | MMMM |

Table A-2 Continued

| Group # | Group generated |
|---|---|
| 107 | MMMMM |
| 108 | MMMMMMMM |
| 109 | MMMRCR |
| 110 | MMRCRCR |
| 111 | MMRM |
| 112 | MMTT |
| 113 | MRCR |
| 114 | MRCRM |
| 115 | MTTTTTTTT |
| 116 | RCCCCC |
| 117 | RCRM |
| 118 | RCRR |
| 119 | RFMRM |
| 120 | RFlR |
| 121 | RMMMM |
| 122 | RMMRRM |
| 123 | RRCR |
| 124 | RRMMM |
| 125 | RRMR |
| 126 | TTTT |
| 127 | WBGBGM |
| 128 | WBGG |
| 129 | WMMM |
| 130 | WWMGGBWR |

# APPENDIX 3

## GROUPS GENERATED FROM HAC BASED ANALYSIS

Table A-3: Groups generated from HAC method

| Group # | Group Generated |
|---|---|
| 1 | AAA |
| 2 | AAAA |
| 3 | AAAAA |
| 4 | AAAAAA |
| 5 | AAAAAAA |
| 6 | AAAAAAAAA |
| 7 | AAAAAAAAAA |
| 8 | AAAAAAAAAAA |
| 9 | AAAAAAAAAAAAAAAAA |
| 10 | AAAAAAAAAAAAAAAAAAAA |
| 11 | AAAAAAAAAAAAAAAAAAAAAAAAAAAGBBWGGBBWGGBWWGGBWWGGAAAGG<br>AAAGAAUUUUUUUUUUUU |
| 12 | AAAAAAAAAAAAAAAAAAAAAAAAAAAGBBWGGBBWGGBWWGGBWWGGAAAGG<br>AAAGAAUUUUUUUUUUUUUCRMRAAAAAAAAAAAAAAAAAAAAAAAAAA |
| 13 | AAAAAAAAAAAAAAAAAAAAAAAAAAAGWGGBBWWGGBBWWWWGBWWGGAAA<br>AGAAAGGAUUUUUUUUUUUUUUAAAAAAAAAAAAAAAAAAAAAAAAAAWW |
| 14 | AAAAAAAAAAAAAAAAAAAAAAAAAAGBBWGGBBWGGBWWGGBWWGGA |
| 15 | AAAAAAAAAAAAAAAAAAAAAAAAAAGWGGBBWWGGBBWWWWGBWWGGA |
| 16 | AAAAAAAAAAAAAAAAAAAAAAAAAAWWWGGBBWWWGGGBBBWWGGBBA |
| 17 | AAAAAAAAAAACM |
| 18 | AAAAAAAAAAACMGMMM |
| 19 | AAAAAAAAWWWGGBBWWWGGGBBBWWGGBBA |
| 20 | AAAAAGBBWGGBBW |
| 21 | AAAAAGWGGBBWWG |
| 22 | AAAAAGWGGBBWWGGBBWWWWGBWWGGA |
| 23 | AAAAAWWWGGBBWW |
| 24 | AAAAAWWWGGBBWWWGGGBBBWWGGBBA |
| 25 | AAAGAAAGGA |
| 26 | AAGGAAAGAA |
| 27 | ACA |

Table A-3 Continued

| Group # | Group generated |
|---|---|
| 28 | ACAC |
| 29 | ACACMACACMGGMMMM |
| 30 | ACACMRMMRMMRMC |
| 31 | ACACRRMM |
| 32 | ACACRRMMRW |
| 33 | ACBWRM |
| 34 | ACBWRMMMG |
| 35 | ACC |
| 36 | ACCGMGWW |
| 37 | ACCM |
| 38 | ACCMCMC |
| 39 | ACGBRM |
| 40 | ACGMRA |
| 41 | ACGMRACRCGMMM |
| 42 | ACGRA |
| 43 | ACM |
| 44 | ACMACMMMMMMMMMMMMMMMMMMMMMMMMM |
| 45 | ACMACRRMGA |
| 46 | ACMACRRMGACACACCMMACMACGACMACMMACM |
| 47 | ACMCRCR |
| 48 | ACMM |
| 49 | ACMMMM |
| 50 | ACMMMMA |
| 51 | ACMMMMACRM |
| 52 | ACMMMRACMCMCRM |
| 53 | ACMR |
| 54 | ACMRACM |
| 55 | ACMRRM |
| 56 | ACR |
| 57 | ACRA |
| 58 | ACRAAMMMMMMMMMMMMMMMMCRMRGWACRMMRCCACRRMRMMMMRMFl |
| 59 | ACRACR |
| 60 | ACRACRMM |
| 61 | ACRBGM |
| 62 | ACRCMMMMMMMMMMMMM |
| 63 | ACRCRRMMMMMM |
| 64 | ACRCRRMMMMMMMMMMMM |
| 65 | ACRG |

Table A-3 Continued

| Group # | Group generated |
|---|---|
| 66 | ACRGA |
| 67 | ACRGACRACRACRMMMACRMGACMRMMMACRMMMM |
| 68 | ACRM |
| 69 | ACRMACR |
| 70 | ACRMACRM |
| 71 | ACRMACRMCRRCRCRACR |
| 72 | ACRMFlMMM |
| 73 | ACRMGTMM |
| 74 | ACRMM |
| 75 | ACRMMBMWR |
| 76 | ACRMMGWACRWACRMMMMWM |
| 77 | ACRMMM |
| 78 | ACRMMMM |
| 79 | ACRMMMMMMMRM |
| 80 | ACRMMMMMMMRMACM |
| 81 | ACRMRMACRMACGRACAC |
| 82 | ACRMW |
| 83 | ACRMWACM |
| 84 | ACRR |
| 85 | ACRRG |
| 86 | ACRRM |
| 87 | ACRRMGMM |
| 88 | ACRRMMMM |
| 89 | ACRRMWWMMMWW |
| 90 | ACRTACMMMM |
| 91 | ACRWBMMMMM |
| 92 | ACRWM |
| 93 | ACRWMACRM |
| 94 | ACW |
| 95 | ACWA |
| 96 | ACWBRM |
| 97 | AGAAGGAGAAAG |
| 98 | AGCM |
| 99 | AGMMM |
| 100 | AGMMMMM |
| 101 | AMMMMMMMMMMMMMMMM |
| 102 | ARB |
| 103 | ARG |

Table A-3 Continued

| Group # | Group generated |
|---|---|
| 104 | ARGR |
| 105 | ARM |
| 106 | ARMM |
| 107 | AXCGACRACRAR |
| 108 | AXCGACRACRARCGGG |
| 109 | BBB |
| 110 | BBBWBBBBBWBB |
| 111 | BBBWBBBBBWBBTT |
| 112 | BBG |
| 113 | BGGG |
| 114 | BGM |
| 115 | BGW |
| 116 | BMB |
| 117 | BMGMBW |
| 118 | BMGMBWMWW |
| 119 | BWM |
| 120 | CAC |
| 121 | CACC |
| 122 | CACCC |
| 123 | CACCM |
| 124 | CACCMMACMACGA |
| 125 | CACM |
| 126 | CACMMRMRM |
| 127 | CACMRMMR |
| 128 | CACRM |
| 129 | CACRRMRM |
| 130 | CARB |
| 131 | CARBCMMRMACMMARCRR |
| 132 | CBWRM |
| 133 | CCAC |
| 134 | CCACCC |
| 135 | CCC |
| 136 | CCCACCAC |
| 137 | CCCC |
| 138 | CCCCC |
| 139 | CCCCCC |
| 140 | CCCCCCC |
| 141 | CCCCCCCCCC |

Table A-3 Continued

| Group # | Group generated |
|---|---|
| 142 | CCCCCCCCMCCCCCCCC |
| 143 | CCCCCCCCMCCCCCCCCMW |
| 144 | CCCCCMCCCCCCCC |
| 145 | CCCCMM |
| 146 | CCM |
| 147 | CCMC |
| 148 | CCMM |
| 149 | CCMRCRMM |
| 150 | CCRRMMRMMMMMMMMMMMMMMMMCRMM |
| 151 | CGA |
| 152 | CGACRC |
| 153 | CGB |
| 154 | CGBMGGBMG |
| 155 | CGBMGGBMGGA |
| 156 | CGBMMR |
| 157 | CGBR |
| 158 | CGBRM |
| 159 | CGC |
| 160 | CGCC |
| 161 | CGCCCCCCCCCCCCCCCCCCCCCC |
| 162 | CGCCCCCCCCCCCCCCCCCCCCCCCC |
| 163 | CGCCCCCCCCCCCCCCCCCCCCCCCCMBGGGACMRMMM |
| 164 | CGGB |
| 165 | CGGBMMMMM |
| 166 | CGGBMMMMMARGR |
| 167 | CGGG |
| 168 | CGGRMA |
| 169 | CGM |
| 170 | CGMG |
| 171 | CGMM |
| 172 | CGMRA |
| 173 | CGMXA |
| 174 | CGR |
| 175 | CGRMA |
| 176 | CGRMACRMRCRMMR |
| 177 | CGW |
| 178 | CGWBA |
| 179 | CGWWGG |

Table A-3 Continued

| Group # | Group generated |
|---|---|
| 180 | CMA |
| 181 | CMACMMACM |
| 182 | CMC |
| 183 | CMCARM |
| 184 | CMCCCCMM |
| 185 | CMCM |
| 186 | CMCMCMCMCMWCMCM |
| 187 | CMCMCMCMWCMCM |
| 188 | CMCMMCMM |
| 189 | CMCMR |
| 190 | CMG |
| 191 | CMGA |
| 192 | CMGACGWBA |
| 193 | CMGACGWBACWGMACWACWGACBACWA |
| 194 | CMGGG |
| 195 | CMGGM |
| 196 | CMGGMMMM |
| 197 | CMM |
| 198 | CMMAR |
| 199 | CMMC |
| 200 | CMMCM |
| 201 | CMMFFl |
| 202 | CMMM |
| 203 | CMMMM |
| 204 | CMMMMMMA |
| 205 | CMMMMMMMMMMMMMMMMMMMMM |
| 206 | CMMMMMMR |
| 207 | CMMMMMMRRM |
| 208 | CMMMR |
| 209 | CMMMRACMCMCR |
| 210 | CMMRM |
| 211 | CMMRMA |
| 212 | CMMRMRM |
| 213 | CMMRR |
| 214 | CMR |
| 215 | CMRM |
| 216 | CMRMM |
| 217 | CMRMMMMMMMMRMAR |

Table A-3 Continued

| Group # | Group generated |
|---|---|
| 218 | CMRRM |
| 219 | CMWCMCM |
| 220 | CMXN |
| 221 | CNCMM |
| 222 | CNMXMMXMMMM |
| 223 | CRA |
| 224 | CRAC |
| 225 | CRACCCCACCACCRMMRRMM |
| 226 | CRACGBMGGBMGGACRACACACACCCMGGG |
| 227 | CRACGBMGGBMGGACRACACACACCCMGGGMMM |
| 228 | CRACR |
| 229 | CRACRACCC |
| 230 | CRACRMM |
| 231 | CRACRMMMA |
| 232 | CRACRRCAC |
| 233 | CRAR |
| 234 | CRB |
| 235 | CRBG |
| 236 | CRC |
| 237 | CRCCRMMMMMM |
| 238 | CRCGMMM |
| 239 | CRCMMMMMMMMMMMMM |
| 240 | CRCR |
| 241 | CRCRACWWBBMMMMMMMM |
| 242 | CRCRM |
| 243 | CRCRRAMMMMNMM |
| 244 | CRCRRCR |
| 245 | CRFMRT |
| 246 | CRFMRTCMMFFl |
| 247 | CRG |
| 248 | CRGG |
| 249 | CRGRM |
| 250 | CRGW |
| 251 | CRM |
| 252 | CRMA |
| 253 | CRMACR |
| 254 | CRMACRMMM |
| 255 | CRMACRMMMMMMMMMMMMM |

Table A-3 Continued

| Group # | Group generated |
|---|---|
| 256 | CRMACRMWACRWR |
| 257 | CRMDMMR |
| 258 | CRMFl |
| 259 | CRMG |
| 260 | CRMGACMRMMMACRMMMM |
| 261 | CRMGTMM |
| 262 | CRMM |
| 263 | CRMMA |
| 264 | CRMMBMW |
| 265 | CRMMGW |
| 266 | CRMMGWA |
| 267 | CRMMGWACRWACRMMMMWM |
| 268 | CRMMM |
| 269 | CRMMMM |
| 270 | CRMMMMMM |
| 271 | CRMMMRMMMMR |
| 272 | CRMMR |
| 273 | CRMMRA |
| 274 | CRMMRMMRMM |
| 275 | CRMMRMMRMMMMMRMMMRMRCRMMMMMMMMMMMMMMMMMMMM MMMMMMMMMMCMMMMMMMRMRMR |
| 276 | CRMR |
| 277 | CRMRGWACRMMRCCACRRMRMMMMRM |
| 278 | CRMRMA |
| 279 | CRMRMM |
| 280 | CRMRMMMM |
| 281 | CRMW |
| 282 | CRMWM |
| 283 | CRR |
| 284 | CRRC |
| 285 | CRRCA |
| 286 | CRRCACRCRRMMM |
| 287 | CRRCR |
| 288 | CRRG |
| 289 | CRRM |
| 290 | CRRMACRM |
| 291 | CRRMG |
| 292 | CRRMGA |

Table A-3 Continued

| Group # | Group generated |
|---|---|
| 293 | CRRMM |
| 294 | CRRMMM |
| 295 | CRRMMMM |
| 296 | CRRMMRW |
| 297 | CRRMR |
| 298 | CRRWBG |
| 299 | CRT |
| 300 | CRTA |
| 301 | CRTACMMMM |
| 302 | CRW |
| 303 | CRWB |
| 304 | CRWBMMMMM |
| 305 | CRWM |
| 306 | CRWMA |
| 307 | CRWR |
| 308 | CWA |
| 309 | CWBRM |
| 310 | CWGM |
| 311 | CWGMACWACW |
| 312 | CWRM |
| 313 | CWRMMM |
| 314 | FFFFlFFl |
| 315 | FFl |
| 316 | FFlFFl |
| 317 | FMR |
| 318 | FMRM |
| 319 | FMRT |
| 320 | FRA |
| 321 | FRM |
| 322 | FRMMM |
| 323 | FlR |
| 324 | FlRR |
| 325 | FlRRFRMBWR |
| 326 | FlRRFRMBWRB |
| 327 | FlRRFRMBWRBGM |
| 328 | FlRRFl |
| 329 | GACB |
| 330 | GBBGGG |

Table A-3 Continued

| Group # | Group generated |
|---|---|
| 331 | GBBGW |
| 332 | GBBW |
| 333 | GBBWGGBBW |
| 334 | GBBWWWW |
| 335 | GBBWWWWGB |
| 336 | GBG |
| 337 | GBGGBG |
| 338 | GBGW |
| 339 | GBM |
| 340 | GBMG |
| 341 | GBW |
| 342 | GBWR |
| 343 | GFB |
| 344 | GGB |
| 345 | GGBB |
| 346 | GGBBA |
| 347 | GGBBW |
| 348 | GGBBWW |
| 349 | GGBWW |
| 350 | GGBWWGGB |
| 351 | GGG |
| 352 | GGGBBBWW |
| 353 | GGGG |
| 354 | GGM |
| 355 | GGMFRACGMXACGGRMA |
| 356 | GGMFRACGMXACGGRMACGRMACRMRCRMMR |
| 357 | GGW |
| 358 | GGWB |
| 359 | GGWBBBG |
| 360 | GMG |
| 361 | GMMM |
| 362 | GWB |
| 363 | GWBM |
| 364 | GWGGBBWWG |
| 365 | MACACMM |
| 366 | MACGACRCMACRGW |
| 367 | MACM |
| 368 | MACMMM |

Table A-3 Continued

| Group # | Group generated |
|---|---|
| 369 | MACRGG |
| 370 | MACRM |
| 371 | MACRMMA |
| 372 | MACRMMACGWWGG |
| 373 | MAM |
| 374 | MAR |
| 375 | MARCRACGBMMR |
| 376 | MARMMGR |
| 377 | MAXCACRGACRRM |
| 378 | MBG |
| 379 | MBM |
| 380 | MBMBMMW |
| 381 | MBMW |
| 382 | MBW |
| 383 | MCACM |
| 384 | MCCC |
| 385 | MCM |
| 386 | MCMM |
| 387 | MCMMM |
| 388 | MCRMMCRR |
| 389 | MFC |
| 390 | MFRM |
| 391 | MFRMMM |
| 392 | MFRMMMF |
| 393 | MFlR |
| 394 | MGB |
| 395 | MGBMRCMMMMMMR |
| 396 | MGFlR |
| 397 | MGG |
| 398 | MGGBWR |
| 399 | MGGG |
| 400 | MGGM |
| 401 | MGMM |
| 402 | MGR |
| 403 | MMA |
| 404 | MMACA |
| 405 | MMACGC |
| 406 | MMACMRCCCCC |

Table A-3 Continued

| Group # | Group generated |
|---|---|
| 407 | MMACMRCCCCCC |
| 408 | MMACMRM |
| 409 | MMACRMMMMMM |
| 410 | MMACRMWMM |
| 411 | MMARM |
| 412 | MMCM |
| 413 | MMCMM |
| 414 | MMCMMMMM |
| 415 | MMCMMMMMMRRM |
| 416 | MMFRM |
| 417 | MMGWBM |
| 418 | MMGWBMM |
| 419 | MMM |
| 420 | MMMACRM |
| 421 | MMMACRMACRM |
| 422 | MMMACRMMMMMMMMM |
| 423 | MMMCMMWM |
| 424 | MMMCR |
| 425 | MMMG |
| 426 | MMMM |
| 427 | MMMMA |
| 428 | MMMMACWRMMMM |
| 429 | MMMMM |
| 430 | MMMMMA |
| 431 | MMMMMACM |
| 432 | MMMMMACMMMMMMGBGWMBMMMM |
| 433 | MMMMMGBGW |
| 434 | MMMMMGBGWMBMMMM |
| 435 | MMMMMM |
| 436 | MMMMMMACACCRRMMRMMMMMMMMMMMMMMMMCRMM |
| 437 | MMMMMMM |
| 438 | MMMMMMMACMCACRM |
| 439 | MMMMMMMCCCCMMM |
| 440 | MMMMMMMM |
| 441 | MMMMMMMMM |
| 442 | MMMMMMMMMACACRRMM |
| 443 | MMMMMMMMMFRM |
| 444 | MMMMMMMMMM |

Table A-3 Continued

| Group # | Group generated |
|---|---|
| 445 | MMMMMMMMMMM |
| 446 | MMMMMMMMMMMM |
| 447 | MMMMMMMMMMMMM |
| 448 | MMMMMMMMMMMMMM |
| 449 | MMMMMMMMMMMMMMMMM |
| 450 | MMMMMMMMMMMMMMMMMM |
| 451 | MMMMMMMMMMMMMMMMMMM |
| 452 | MMMMMMMMMMMMMMMMMMMM |
| 453 | MMMMMMMMRMAR |
| 454 | MMMMMMMRMCRMMCRRACRMMMM |
| 455 | MMMMMMMRMMMMMMM |
| 456 | MMMMMW |
| 457 | MMMMN |
| 458 | MMMMNMM |
| 459 | MMMMRC |
| 460 | MMMMRM |
| 461 | MMMMRMRMR |
| 462 | MMMMW |
| 463 | MMMN |
| 464 | MMMR |
| 465 | MMMRFRMMMC |
| 466 | MMMRFlRM |
| 467 | MMMRM |
| 468 | MMMRMCRMMCRRA |
| 469 | MMMRMM |
| 470 | MMMRMMMRMRCRMMMMMMMMMMMMMMMMMMMMMMMMMMM |
| 471 | MMMRMMWWBM |
| 472 | MMMW |
| 473 | MMMWM |
| 474 | MMMWMMM |
| 475 | MMMWMMMMMMMM |
| 476 | MMN |
| 477 | MMR |
| 478 | MMRFRMMM |
| 479 | MMRFlRM |
| 480 | MMRM |
| 481 | MMRMMCMMMMMMRRMBGM |
| 482 | MMRMMM |

Table A-3 Continued

| Group # | Group generated |
|---|---|
| 483 | MMRMMMMMMMM |
| 484 | MMRMMMMRM |
| 485 | MMRMMMRMMMMRM |
| 486 | MMRRMRMR |
| 487 | MMRRMRMRCACACMRMMRMMRMCMMMMM |
| 488 | MMW |
| 489 | MMWBGGMMM |
| 490 | MMWBGGMMMACRMACACMMRMRMACR |
| 491 | MMWM |
| 492 | MMWW |
| 493 | MMWWW |
| 494 | MMWWWWWW |
| 495 | MMWWWWWWM |
| 496 | MMX |
| 497 | MMXMGFlRCMM |
| 498 | MMXMGFlRCMMACM |
| 499 | MMXMGFlRCMMACMMMMMMM |
| 500 | MMXMMMNM |
| 501 | MRC |
| 502 | MRCR |
| 503 | MRCRTFRM |
| 504 | MRCRTFRMC |
| 505 | MRM |
| 506 | MRMACRWMMMRMACMCCRMRMMMM |
| 507 | MRMMM |
| 508 | MRMMMMR |
| 509 | MRMMMMRM |
| 510 | MRMMMMRMMM |
| 511 | MRMMMMW |
| 512 | MRMMMRM |
| 513 | MRMMR |
| 514 | MRMMRA |
| 515 | MRMMRACAR |
| 516 | MRMMRMR |
| 517 | MRMRMR |
| 518 | MRRC |
| 519 | MRRCRRC |
| 520 | MRT |

Table A-3 Continued

| Group # | Group generated |
|---|---|
| 521 | MRW |
| 522 | MTT |
| 523 | MWA |
| 524 | MWACMMMMM |
| 525 | MWACRMACRMMMMMMMMMMMMMMMMACRM |
| 526 | MWB |
| 527 | MWM |
| 528 | MWMMMM |
| 529 | MWR |
| 530 | MWW |
| 531 | MWWB |
| 532 | MWWBB |
| 533 | MWWBBMMMMM |
| 534 | MWWM |
| 535 | MXDNN |
| 536 | NCCMXN |
| 537 | NMX |
| 538 | NMXMMX |
| 539 | NMXMMXMMMM |
| 540 | NXAAAAAAGBM |
| 541 | NXAAAAAAGBMWW |
| 542 | NXCMMMMMMACGACRMRMACRMACGRACACN |
| 543 | NXMXNX |
| 544 | RACC |
| 545 | RACMM |
| 546 | RACMMM |
| 547 | RCA |
| 548 | RCRM |
| 549 | RCRR |
| 550 | RFMRM |
| 551 | RFl |
| 552 | RGW |
| 553 | RMACMC |
| 554 | RMG |
| 555 | RMM |
| 556 | RMMM |
| 557 | RMMMMR |
| 558 | RMMR |

Table A-3 Continued

| Group # | Group generated |
|---|---|
| 559 | RMMRMR |
| 560 | RMMRMRMR |
| 561 | RMR |
| 562 | RRCR |
| 563 | RRCRCRA |
| 564 | RRM |
| 565 | RRMRR |
| 566 | RRRR |
| 567 | RRRRR |
| 568 | TTT |
| 569 | TTTT |
| 570 | UUU |
| 571 | UUUU |
| 572 | UUUUU |
| 573 | UUUUUU |
| 574 | UUUUUUU |
| 575 | UUUUUUUUUUU |
| 576 | UUUUUUUUUUUU |
| 577 | WBG |
| 578 | WBGB |
| 579 | WBGBGM |
| 580 | WBGG |
| 581 | WBGGMMM |
| 582 | WBMGR |
| 583 | WBR |
| 584 | WBRMM |
| 585 | WGGGBBBWW |
| 586 | WMMM |
| 587 | WMRRCRRCCCCCCCCC |
| 588 | WWB |
| 589 | WWBB |
| 590 | WWBBMMMMMMMM |
| 591 | WWBM |
| 592 | WWGG |
| 593 | WWGGA |
| 594 | WWW |
| 595 | WWWGGBBWW |
| 596 | WWWMMMMMGBMRCMMMMMMR |

Table A-3 Continued

| Group # | Group generated |
|---|---|
| 597 | WWWW |
| 598 | XACGMMMNM |
| 599 | XAR |
| 600 | XCA |
| 601 | XCACRGACRRM |
| 602 | XCG |
| 603 | XCGA |
| 604 | XCMCM |
| 605 | XMM |
| 606 | XMMNMXGWBN |
| 607 | XWN |
| 608 | XXA |
| 609 | XXACRG |
| 610 | pGGG |
| 611 | ppPP |
| 612 | ppp |
| 613 | ppppp |

# VITA

| | |
|---|---|
| Name: | Bikash Mandal |
| Permanent Address: | 390 South Paikpara, Mirpur, Dhaka, Bangladesh |
| Education: | M.S. Computer Science, Texas A&M University-College Station, Texas, USA- 2005<br>B. Sc. Engg. Computer Science & Engineering, Bangladesh University of Engineering & Technology, Dhaka, Bangladesh, 1994 |